

ZebraBase III

User Manual

Zebrateam Team

zebrateam.org

Table of contents

1. Zebrabase 3	4
2. GUI	5
2.1 Basic navigation and messaging system	5
2.2 Lists	8
2.3 Fish	14
2.4 Facility	35
2.5 Calendar and requesting	46
2.6 Statistics	50
2.7 Useful tips, shortcuts, etc.	56
2.8 FAQ	60
3. Admin	67
3.1 Initial facility configuration	67
3.2 Animal reporting module	80
3.3 Custom fields	99
3.4 Notification system	101
3.5 Printing	105
3.6 Printing troubleshooting	114
3.7 Multi-species module	115
3.8 Batch actions	118
3.9 Attachments	120
3.10 How to use the animal reporting module	121
3.11 Pricing and billing	131
3.12 Crons	134
3.13 Water systems	135
4. Api	137
4.1 Basics	137
4.2 Get	139

4.3	Create & Update	145
4.4	Delete	148
4.5	Substock actions	149
4.6	Accessing and downloading backups	154
5.	Newsletters	155
5.1	3.1.9: Changes in action inputs	155
5.2	3.1.10: Reducing number of projects assigned to substock	158
5.3	3.3.0: Zebrabase News (May/2022)	159
5.4	3.4.0: Zebrabase News (October/2022)	161

1. Zebrabase 3

Dear User,

Welcome to Zebrabase, an intuitive database developed for research facilities that house laboratory animals. Zebrabase combines comprehensive animal stock tracking and management functionalities with advanced visualization and reporting options. Zebrabase supports access via desktop browsers (Win/Mac/Linux) or mobile devices.

This manual will lead you through every step in the process of setting up and using Zebrabase in your animal facility. All developed features are described in the manual as well as the general workflow. The manual is continuously updated. Please note: some functionalities may not be available in your version of Zebrabase, either because your pay plan does not include them or because they are only enabled upon request.

In the introductory email, you have been provided with the web address of your database instance and a single admin account that will enable you to use all the features available in Zebrabase. If you wish, you can first visit our demo instance loaded with sample data and test the system before you start the configuration of your instance.

If you encounter any problems, please, use the help desk to report them. The help desk can also be used to suggest improvements or new features for future versions of Zebrabase.

Zebrabase is a non-profit project. We charge a fee to partially cover the expenses of running the service. To support starting labs and small research facilities, we provide a Basic plan that includes up to 150 tanks. For larger facilities, with more active substocks and users, we offer the Standard or the Premium Plan. These include various sets of features and an unlimited number of tanks. For more information on pricing and features, please, visit the [Zebrabase pricing](#).

Your Zebrabase Team

2. GUI

2.1 Basic navigation and messaging system

There is a [video tutorial](#) available for this section.

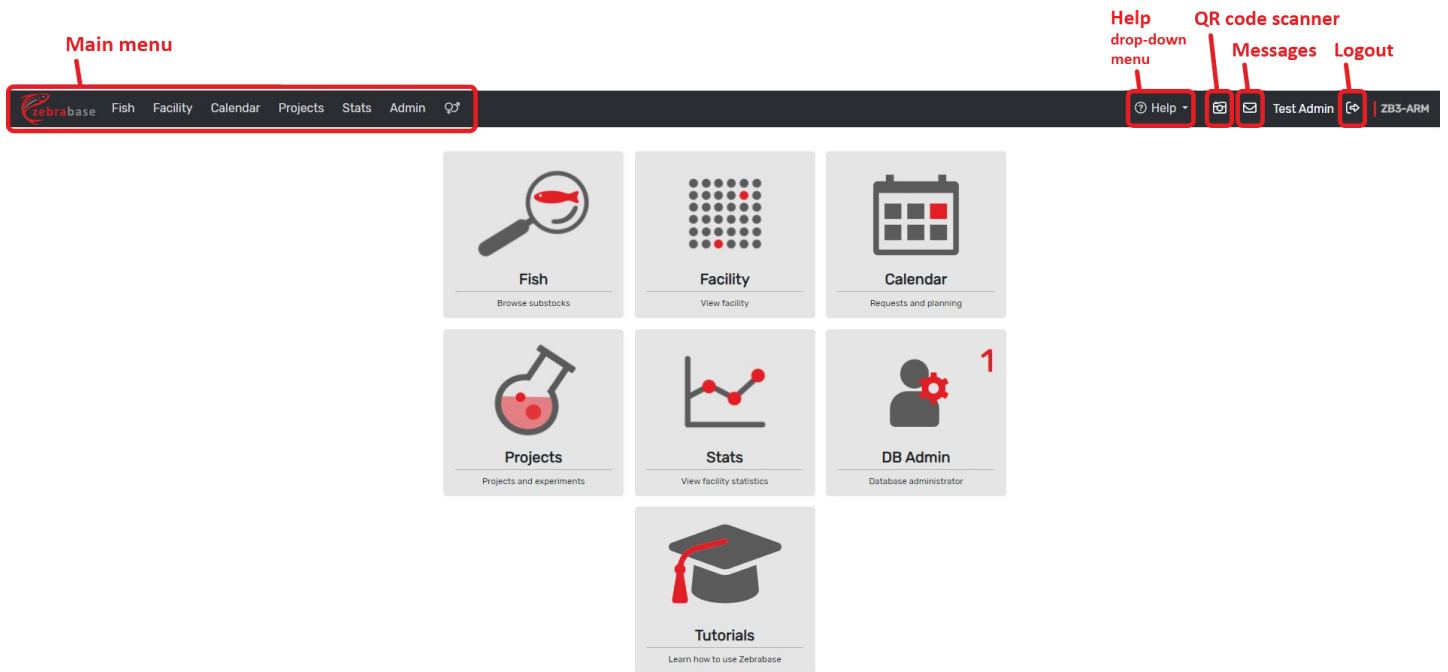
Zebrabase can be accessed from any web browser, however, it is optimized for Google Chrome. For each Zebrabase licence (purchased by a zebrafish facility, for example), there is an independent database instance running. To access your instance, use the unique URL prefix of your instance and type in <prefix>.zebrabase.org into your browser.

2.1.1 Basic navigation

Home page

The Home page is the first page the user sees after logging in. It consists of tiles that navigate the user to the main sections of Zebrabase. Please note that some sections from the image below may be unavailable due to the pay plan settings and user permissions (home page for the Premium plan with admin permissions is shown).

On the top, there is the main navigation panel which can be accessed from any page.



Submenu

Each section (for example Fish) has its submenu that directs the user to the specific pages of the section.



2.1.2 Messaging system

The inbuilt messaging system allows users to send and receive messages, or to receive system notifications. To access it, click the envelope icon in the upper right corner.

To write a new message, click the blue paper plane icon. To reply to a message, go to its detail and click **Reply** or **Reply to all**.

Search messages
filter/quick search

Send new message

Subject	From	To	Text	Sent
ZebraBase request notification	root	facility	New request was created by alesbalik and assigned to you. Setup for crossing fishline WT (AB) https...	2021-05-17 09:56 [UTC]
ZebraBase request notification	root	facility	New request was created by jovana and assigned to you. Setup for crossing substocks p53 ^{M214K}	2021-05-10 12:34 [UTC]
ZebraBase request notification	root	facility	New request was created by jovana and assigned to you. Cross fishline WT (AB) https://zebrabase.cz/c/...	2021-05-10 12:29 [UTC]

NOTE: For system notifications, there is also the option to receive emails or Slack messages, see chapter [Notification system](#) for more info.

2.2 Lists

2.2.1 Filters

Filtering is possible according to nearly any variable in the list. To open the filter settings, click the filter icon in the upper left corner. Choose the parameters and then click **Apply filter**, this will set your filter for the current session (the filter will remain after leaving the page, but will reset after logging out).

The second option is the **Quick search**, which searches for a string in the name or the ID.

The screenshot shows the Zebrafish database interface. At the top, there are navigation tabs: Genotypes, Fishlines, Stocks, and Substocks (selected). Below the tabs, there are several icons: a filter icon (labeled 'Filter'), a refresh icon (labeled 'Reset filter'), a search icon (labeled 'Quick search'), and a save icon (labeled 'Save filter'). The search bar contains the text 'status & (dead, split, merged)'. The table below shows a list of substocks with columns: Name, BG, F-gen, Num, Birth, Status, Count, Position, Owners, Responsible, Workgroups, Project, Funding, Use, Statusbar, and ID. The table is filtered to show 666 items out of 14700 total items. The active filters are listed as 'status & (dead, split, merged)'. The interface also includes a 'Filter' icon, a 'Reset filter' icon, a 'Quick search' icon, and a 'Save filter' icon. The 'Filtered items / All items' count is shown as 'Σ 666 / 14700'. The 'Export list' and 'Organize columns' icons are also visible.

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Project	Funding	Use	Statusbar	ID
WT (AB)	AB	1	2023-11-23	unproductive	not set	1	fish	facility	CD Lab	Core Fishlines					7524-16703
cd41:GFP ^{+/+}	AB	1	2023-11-21	unproductive	not set	1	fish	iva	CD Lab						7519-16694
CASPER	AB	9	2023-11-15	unproductive	25	801-D04	fish		CD Lab						7504-16686
CASPER	AB	8	2023-11-15	unproductive	25	003-E03	fish		CD Lab						7506-16685

Active filters (both session and URL) are listed just above the list. The whole panel is hidden when there are no active filters. Individual filters can be cleared from this list, another option is to **Clear all filters** in the filter settings.

The screenshot shows the Zebrafish database interface. At the top, there are navigation tabs: Genotypes, Fishlines, Stocks, and Substocks. Below the tabs, there are search and filter icons. The 'Active filters' section shows a filter for 'status ≠ [dead, split, merged]'. Below this, there are input fields for 'Name contains:', 'Date of birth:', and 'Description contains:'. There are also buttons for 'Apply filter', 'Reset to default', and 'Clear filter'. Below the filters, there is a table with columns: Name, BG, F-gen, Num, Birth, Status, Count, Position, Owners, Responsible, Workgroups, Project, Funding, Use, Statusbar, and ID. The table contains three rows of data: WT (AB), cd410FP +/-, and CASPER. Red boxes and arrows highlight the 'Active filters' section, the 'Reset to default' and 'Clear filter' buttons, and the corresponding actions in the table.

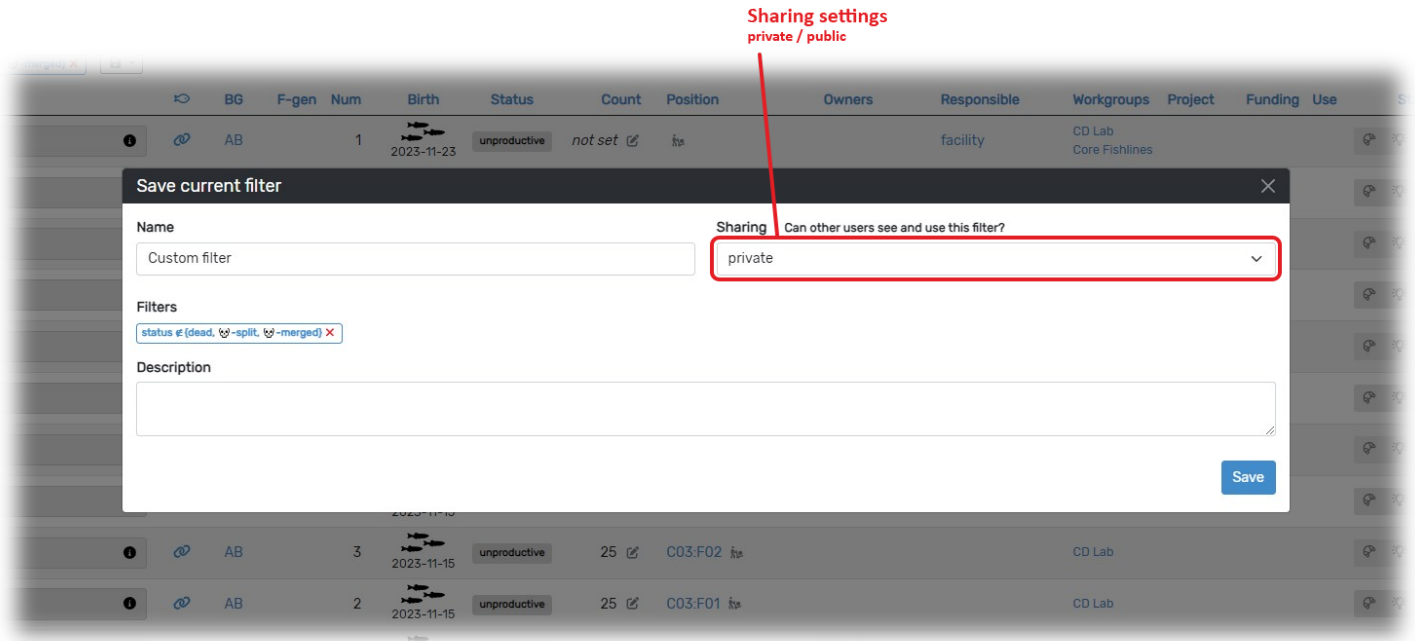
Default filters

When a filter is active, the user can click the **Save** button and save it as **User default filter**. Admins can also set an **Instance default filter**. Default filters are set up after logging in and are stored in the session. User settings are the primary source for the default filters, while the facility settings form the secondary source.

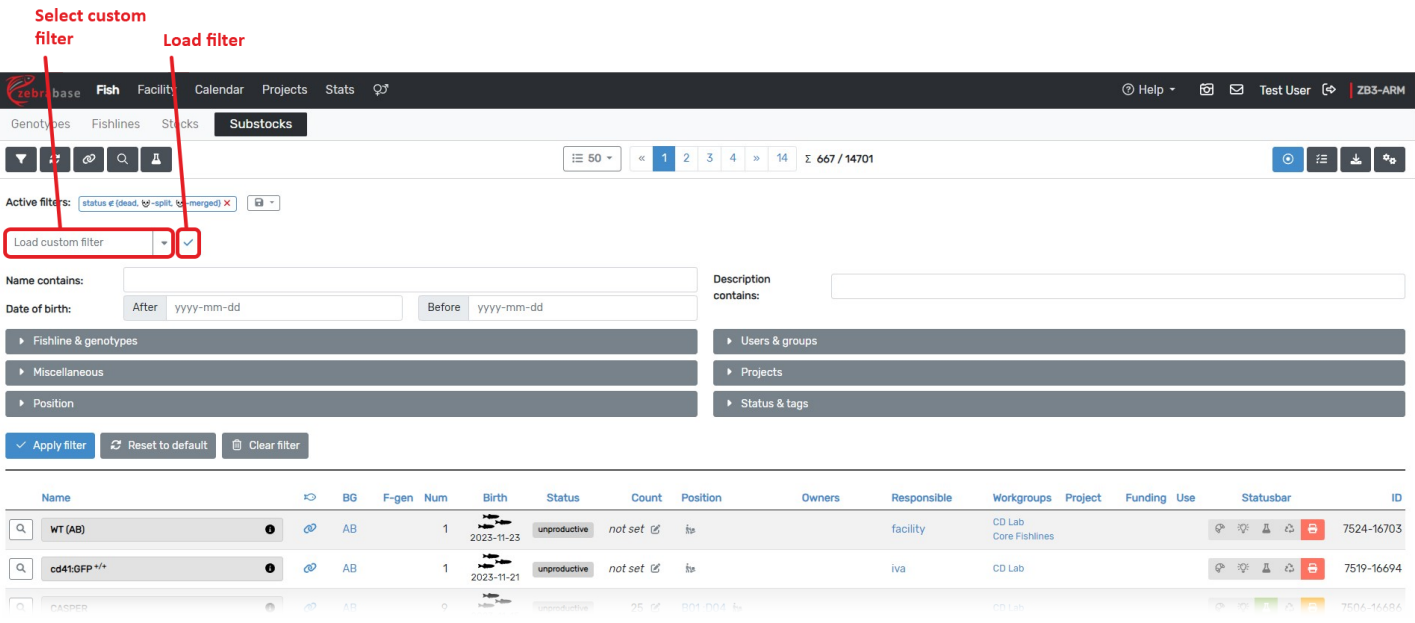
Both types of default filters use the same format of key: `defaults.list.<page>.filter` (e.g. `defaults.list.substock.filter`), the value is a JSONised dictionary `attribute[_operator]: value`.

Custom filters

Another option when saving is the **Custom filter**. Each user can save the current filter settings as a custom filter and share them with others by setting the option **public** to the filter when saving.



After saving, a new field **Load custom filter** will appear in the filter settings. The user can select the custom filter by typing in its name or by selecting it from the drop-down menu. The filter is loaded by clicking the checkmark icon **Load filter**.



Custom filters can be managed in Admin/Codebooks/Others/Custom filter.

Filter URL

Filtering can also be done using URL parameters. The user can get the URL of a currently used filter by clicking **Get filter URL**. URL filters have higher priority over the session filters and will be lost after leaving the page.

Filter format

The URL filter format is Django-based with one exception: negative filters end with `!`, so for example `status_in!` means "status not in ...". Values for the `_in` operator are comma-separated.

Example:

JSON filter settings

```
{"status_in!=dead, -split, -merged", "stock_date_of_birth_gte": "2010-01-01"}
```

corresponding URL filter

```
/list/substock?status_in!=dead, -split, -merged&stock_date_of_birth_gte=2010-01-01.
```

Special URL parameters

There are two special URL parameters related to filters. Parameter `_url` omits all session filters so that only URL filters are used (this is used in generated links in Zebrabase). Parameter `_all` displays all objects including any previously deleted objects.

Examples of use:

```
zebrabase.org/list/substock?_all
```

```
zebrabase.org/list/substock?_all&_url
```

```
zebrabase.org/list/substock?genotypes_isnull=True&_url
```

2.2.2 Columns

Columns to display in a list and their order can be organized in **Column settings**. The form is invoked by clicking the cog icon on the right side of the list toolbar.



If the column setting originates from the instance default the icon will change to:



If the setting is a personal default the icon will change to:



The form is divided into two panels, on the left side there are the active columns that will be displayed in your list, and on the right side there are the inactive columns. To display a column in your list, simply drag and drop it into any position in the left panel.

Load column settings
Temporary / Personal default / Instance default / Zebrabase default

Display column on mobile device / PC

Save current settings
Temporary / Personal default / Instance default

Column settings can be stored as **Personal default** by any user or as **Instance default** by the admin. The default user settings can be reverted from a confirmation message above the list or in the column settings.

The columns to be displayed are by default shown on both **mobile and desk devices**. Clicking the blue phone or PC icon will de-select the column from being displayed on that device.

2.2.3 Export

Filtered lists can be exported as .XLSX files by clicking the download icon in the upper right corner. The items in your list can be managed by the filter settings, however, column settings will not transfer into your exported file.

If you encounter problems with exporting, please check the length of your filtered list. Very long lists can have problems with export, and the option to export is not offered for excessively long lists.

2.3 Fish

There is a [video tutorial](#) available for this section.

The section Fish provides access to the lists of Genotypes, Fishlines, Stocks, and Substocks.

2.3.1 Genotypes

Genotypes are the elementary genetic modifications (e.g. mutations, transgenes), that combine to form a **fishline**.

Name	Mod. category	Driver	Triggered gene	Affected gene	Allele / Mutation	Fishlines
p53 ^{M214K}	Mut[N]			p53	M214K	filtered list
dll4-GAL4	Tg	dll4	GAL4			filtered list
mpx:GFP	Tg	mpx	GFP			filtered list

New genotypes can be added by clicking the plus icon in the upper right corner of the list. This opens a new table, where you select the type of the genotype: **transgenic**, **mutant (natural)**, or **mutant (engineered)**, and specify further details about the modification. Add new parameters by clicking on the plus icon, or select a previously defined parameter from the list. In transgenic genotypes, the **Driver** and the **Triggered gene** are obligatory fields, and in mutant genotypes, only the **Affected gene** is obligatory.

New genotype Save

Modification category *

Transgenic

Driver *

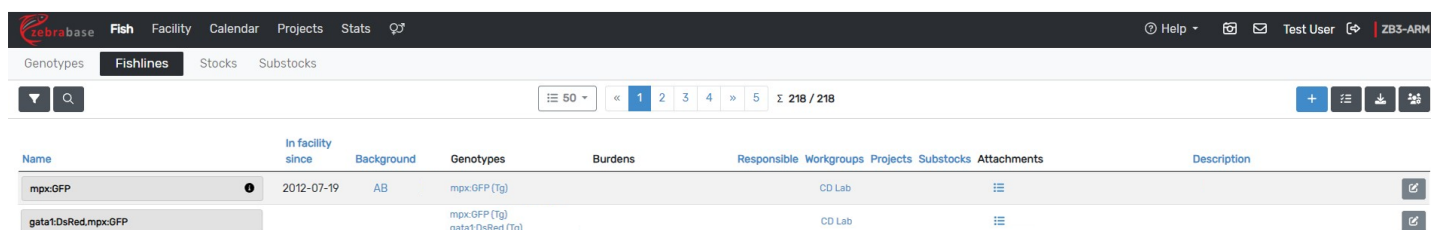
Triggered gene *

Allele / Mutation

Save Save and add another Back to list

2.3.2 Fishlines

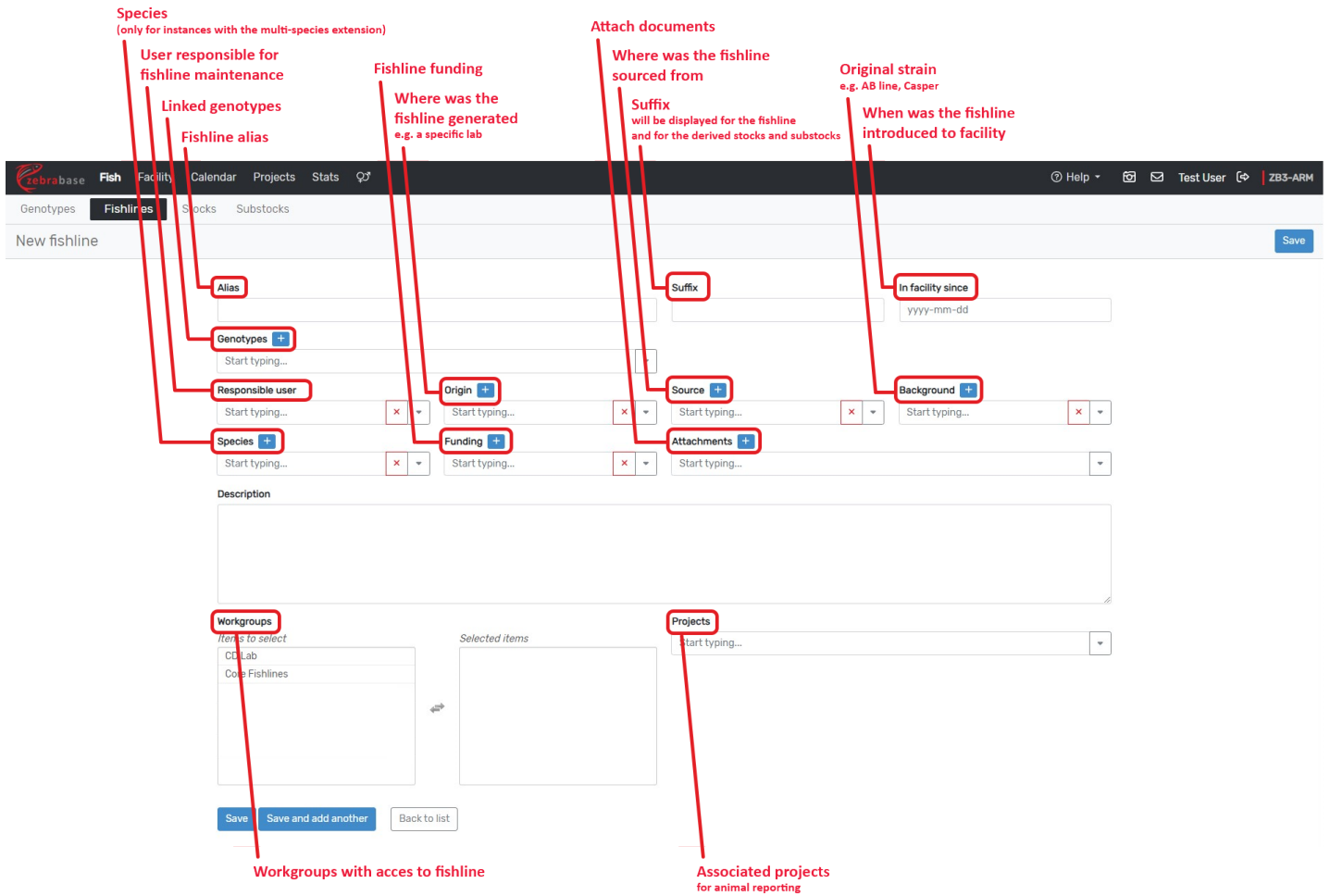
Fishlines are combinations of genotypes that occur in your fish. You can either design them de novo (for example when obtaining new fish from a different facility), or create them during crossing: a new fishline with the corresponding genotype combination is automatically generated when out-crossing two different fishlines, or when assigning a new genotype during crossing (for example modifications done by microinjections).



The screenshot shows the ZebraBase interface for Fishlines. The top navigation bar includes 'Fish', 'Facility', 'Calendar', 'Projects', and 'Stats'. The main content area has tabs for 'Genotypes', 'Fishlines', 'Stocks', and 'Substocks'. A search bar and pagination controls (showing 1-5 pages, total 218/218) are visible. Below is a table with the following columns: Name, In facility since, Background, Genotypes, Burdens, Responsible, Workgroups, Projects, Substocks, Attachments, and Description.

Name	In facility since	Background	Genotypes	Burdens	Responsible	Workgroups	Projects	Substocks	Attachments	Description
mpx:GFP	2012-07-19	AB	mpx:GFP (Tg)			CD Lab				
gata1:DsRed.mpx:GFP			mpx:GFP (Tg) gata1:DsRed (Tg)			CD Lab				

New fishlines can be added by clicking the plus icon. **It is mandatory to fill in either the fishline alias or the genotypes**, other fields are optional. Additional documents can be added as attachments. The field **Workgroups** specifies the users who will be able to view the fishline and the derived substocks. All derived stocks and substocks will contain the **Fishline suffix** in their name.



2.3.3 Stocks

Stocks are the complete set of fish coming from one crossing. They further divide into substocks, which are groups of related fish bred in one tank. New stocks are generated automatically after a crossing (i.e. they are always connected to at least one existing substock).

Name	Birth	Generation	Responsible	Workgroups	Substocks	Loc. stock ID	Stock ID
CASPER 2023-11-27/2	2023-11-27			CD Lab	1 (list)	7528	7528
CASPER 2023-11-27/1	2023-11-27			CD Lab	1 (list)	7527	7527
WT (AB) 2023-11-27/1	2023-11-27		facility	CD Lab Core Fishlines	1 (list)	7526	7526
WT AB (DR) 2023-11-27/1	2023-11-27		iva	CD Lab Core Fishlines	1 (list)	7525	7525

Stock detail

The stock detail contains information on the fishline, date of birth, and the derived substocks. On the right side, there is the **Observation log** and **Action log**.

It is possible to add a generation number to a stock. The number can be manually set in **Update stock** (in stock or substock detail). With each in-cross, the number is automatically increased by one.

The **Stock suffix** is another information that can be maintained from the **Update stock** form. All derived substocks will contain the stock suffix in their name.

2.3.4 Substocks

Substocks are groups of related fish coming from one crossing, and bred together in one tank. They are the key units of the database, and most actions are directly connected to them.

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Project	Funding	Use	Statusbar	ID
WT (AB)	AB	1	2023-11-23	unproductive	not set	fs	facility	CD Lab Core Fishlines							7524-16703
cd41GFP +/+	AB	1	2023-11-21	unproductive	not set	fs	iva	CD Lab							7519-16694
CASPER	AB	9	2023-11-15	unproductive	25	B01.D04 fs		CD Lab							7506-16686
CASPER	AB	8	2023-11-15	unproductive	25	C03.E03 fs		CD Lab							7506-16685

Filtering the list of substocks is possible according to nearly any variable. The filter icon opens a form with filtering rules sorted into categories. There is also the option to load a custom filter that has been saved previously. Any filter setting can be saved as an instance default, user default, or a custom filter. Custom filters can be named and shared with other users (if saved as public). **As a default setting, dead or terminated substocks are not included in the search results.**

Another option is the Quick search, which searches for a string in the name or the ID.





The screenshot shows the ZebraBase Substocks interface. At the top, there are navigation tabs: Genotypes, Fishlines, Stocks, and Substocks. Below the tabs, there are icons for Filter (a downward arrow) and Quick search (a magnifying glass), both highlighted with red boxes and labels. The main content area includes an 'Active filters' section with a dropdown menu and a 'Load custom filter' button. Below this, there are input fields for 'Name contains:', 'Date of birth:' (with 'After' and 'Before' options), and 'Description contains:'. There are also several expandable sections for filtering: Fishline & genotypes, Miscellaneous, Position, Users & groups, Projects, and Status & tags. At the bottom, there is a table with columns: Name, BG, F-gen, Num, Birth, Status, Count, Position, Owners, Responsible, Workgroups, Project, Funding, Use, Statusbar, and ID. The table contains two rows of data:

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Project	Funding	Use	Statusbar	ID
WT (AB)	AB		1	2023-11-23	unproductive	not set	fs		facility	CD Lab Core Fishlines					7524-16703
cd41:GFP ^{+/+}	AB		1	2023-11-21	unproductive	not set	fs		iva	CD Lab					7519-16694









































In Standard and Premium plans, the columns that are displayed in the lists can be customized by clicking the cog icon in the upper right corner of the list. For substocks, there are over 60 different columns at the moment that can be selected. It is advisable to inspect which columns are displayed in the default setting and possibly customize the selection.

Use drag & drop to select columns to display

Current settings source: **Zebrabase**

Load settings:    

Displayed columns

(...detail)	 
Name (list_name)	 
(...fishline)	 
BG (list_background)	 
F-gen (generation)	 
Num (number)	 
Birth (maturity_date_of_birth)	 
Status (status_tags)	 
Count (editable_count)	 
Count (count)	 
Position (editable_position)	 
Position (position)	 
Owners (owners)	 
Responsible (responsible)	 
Workgroups (workgroups)	 
Project (project)	 
Funding (funding)	 
Use (substock_use)	 
Statusbar (statusbar)	 
ID (sid_ssid)	 

Not displayed columns

(action_menu)
SSID (id)
ID (sid_num)
Name (name)
Stock (stock)
Fishline gen. name (fishline_generated_name)
Fishline Attachments (fishline_attachments)
Age [days] (age_days)
Age [weeks] (age_weeks)
Age [months] (age_months)
Age (age_human_readable)
Birth (date_of_birth)
Death (date_of_death)
Description (description)
Status (status)
Status (maturity_status)
Tags (tags)
Exp. Project (experimental_project)
LSID (local_stock_id)
Suffix (suffix)
Burdened (burdened)
To print (to_print)
Printed (printed)
Printed (last_print)
Experiment (experiment)
Most severe procedure (most_severe_procedure)
Highest severity (highest_severity)
Species (species)
Diet (diet)
Light (light)
Conditions (conditions)
Siblings (link_to_siblings)
Observ. (link_to_observations)
(observations_modal)
Σ (average_egg_quantity)
☆ (average_egg_quality)
Crossings (latest_crossings)

See more info on filters and lists [here](#).

Founder substock

New substocks can be added by clicking the blue **Founder substock** icon. This function allows the creation of a new substock without a previous crossing and **should only be used if generating the new substock by crossing is not possible** (for example when obtaining fish from a different facility). It is recommended to always use the dedicated crossing function when creating new substocks from substocks already present in your facility. Only this will ensure that the pedigree is created and the information on parents and siblings preserved.

The form consists of two parts, the first part is dedicated to the respective **Stock** and the second part to the **Substock**. Leaving the **Fish in each substock** field empty will generate substocks with the fish count value "not set".

The screenshot shows the 'New substock' form in the ZebraBase application. The form is organized into several sections:

- Stock Section:** Contains a 'Fishline' dropdown menu, a 'Date of birth' text input with a 'YYYY-MM-DD' placeholder, and two empty 'Gen.' and 'Suffix' text inputs.
- Substock Section:** Contains a 'Number of substocks' text input with the value '1', a 'Fish in each substock' text input, a 'Position' dropdown menu with a search field 'Find a position by name', and a 'Suffix' text input.
- Description:** A large text area for entering details about the substock.
- Ownership and Projecting:** Three dropdown menus for 'Substock owners', 'Workgroups', and 'Substock project'.
- Actions:** Three buttons at the bottom: 'Save', 'Save and add another', and 'Back to list'.

Substock detail

Clicking on a substock displays its **Substock detail**. All available information about the substock is displayed here, including the **Parent tree**, the **Label preview** and **Action log** that contains all actions performed with the substock or its parent structures (stock, fishline). All **Siblings** of this substock can be viewed as well.

The field **Crossings** contains information on the last three crossings performed with this substock (date, egg quality, egg quantity).

Substock #15975 - ubi:mCherry+ 2023-06-13/1 (5)

Buttons: Delete, Update, Update stock, Back to list, Notify owners, Print QR

Species: 15975
 Status: productive
 Tags: ubi:mCherry 2023-06-13/1
 Num: 5 / 7
 Fishline: ubi:mCherry
 Suffix: TU
 Background: B02 :C05
 F-gen: B02 :C05
 Use: TU
 Funding: B02 :C05
 Count: 39
 Birth: 2023-06-13
 Death: not yet
 Most severe procedure: none
 Parents: ubi:mCherry+ 2022-06-08/1 (3), ubi:mCherry+ 2022-06-08/1 (3)
 Description: fishline: GM0 APPROVED: 2012
 LSID: 7281
 Responsible: facility
 Diet: default
 Light: default
 Experiment: none
 Burdened: x
 Printed: 2023-06-19
 Project: none
 Exp. Project: none
 Genotypes: -3.Subi:mCherry+
 Crossings: 2023-11-08: 2 stars
 Owners: CD Lab
 Workgroups: CD Lab

Label: ubi:mCherry (+)
 QR code: DOB: 2023-06-13 [FNone], ID: 7281-15975 [5 / 7], POS: B02 :C05, OW: , PRJ:

Observation log: There is nothing here, yet.

Action log:
 transfer 2023-06-19
 genotyping [1] 2023-06-19
 count 2023-07-17
 transfer 2023-10-19
 count 2023-10-19
 merge [1] 2023-10-19
 crossing 2 stars [2] 2023-11-08
 productivity 2023-11-08
 count 2023-11-16

Siblings:

IDs / Num	Position	Count
7281-15970	---	0
7281-15972	---	0
7281-15973	---	0
7281-15974	B02 :C06	20
7281-15976	---	0
7281-15977	---	0

Parent tree:
 1 ubi:mCherry+ 2023-06-13/1 (5)
 2 ubi:mCherry+ 2023-06-13/1 (1)
 3 ubi:mCherry+ 2022-06-08/1 (3)

2.3.5 Substock actions

Actions are used to associate new information with the substock (e.g. number of fish, position, genotyping information). Records of all performed actions are stored in the **Action log** in substock detail, and the full list of all actions from all substocks is also available in Admin/Logs/Action log, where actions can also be searched, filtered, and exported.

Users can access substock actions from multiple sites:

- In **Substock detail** - there is also the option to **Update** substock or **Delete** it
- In **Substocks** - you can select **Action menu** to be displayed in the list, to be able to access it directly from here
- In **Facility view** - select a substock in the **View** mode and the action bar will open

Scanning the **QR code** on a tank in your facility will lead you to the **Substock detail** of the substock in the tank.

The actions accessible from the action bar are:



Count - change fish count, report dead fish



Terminate - terminate the whole tank



Crossing - report crossing (create child substock)



Split - split off more substocks from an existing one



Merge - merge akin substocks into one



Genotyping - record genotyping results



Productivity - change the productivity tag



Transfer - change substock position



Update tags - change custom tags



Project experiment - add/remove substock from an experiment (only available when ARM module is active)



Print QR - print substock label

Update

The function **Update** allows the users to change the basic information about the substock: **Substock Use, Funding, Description, Substock suffix, Sex, Printing status, Project, Owners and Workgroups**. A **Note** on the action can also be included.

Count & Terminate

This action enables efficient tracking of fish counts in the substock. Use the **Count** action to specify the number of fish in a tank, to report animals that were found dead or those that have been used for an experiment. The action offers a selection of various **Count types** that serve to distinguish between these situations. **Terminate** is used when all fish in the substock are terminated at once, and it supports the use of the same count types. If you declare all remaining fish in the substock as dead by using **Count**, the result will be the same as using the **Terminate** action - the substock will obtain the "deceased" status and its position will be unassigned.

1) Select a substock and click **Count** in the action bar

2) Set the **Count type**:

- **Start count** - the initial count of fish, other count types can only be selected once the start count has been set
- **Died** - fish that were found dead in the tank
- **Euthanized** - euthanized fish
- **Used for Experiment** - fish euthanized as part of an experiment - this count type is only available when ARM is disabled
- **Correct** - use it to correct a mistake in fish count

IMPORTANT: If the substock has not been counted yet, you need to assign the **Start count** first, meaning you count how many fish there are in the tank and record it. After that, you will be allowed to perform any subtractive operations like **Died**, **Euthanized**, **Used for experiments**, or correct the count via **Correct**.

3) Write down the count (the number of fish this action concerns).

4) You can change the date when the action was performed, or add a note that will be visible in the action log.

5) Save the record.

See the section [Animal reporting module](#) to learn how to use Count to report animals used for experimental procedures with ARM enabled.

Crossing

This function serves for in-crossing and out-crossing substocks in the database. When out-crossing substocks of different fishlines, a new fishline with the corresponding genotype combination is automatically created.

1) **Select the substock** you want to cross, either in the **Fish** list or in the **Facility view**. If you wish to perform out-cross, find one of the parents first. Pick **Crossing** in the action menu. **An empty crossing form can also be accessed by clicking the crossing icon in the top bar or by pressing Shift+X+X.**

Crossing [X]

Performed * [•] date of birth of progeny substocks Note

2024-03-04

Parent substocks + Parent

Exclude dead fish Include default filter ▽ prefilters Sex

1 No matching data. [x] [v] unknown [v] [trash]

Exclude dead fish Include default filter ▽ prefilters Sex

2 No matching data. [x] [v] unknown [v] [trash]

Change productivity of parents to productive Egg quantity: 0 0 0 0 x Egg quality: ☆☆☆☆ x

Progeny

Number of new substocks *	Number of fish in each substock	Total fish count	Stock suffix	Generation [•]
<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Owners of the new substock Project for the new substock

[x] [v]

Add non-parental genotypes [•] (optional)

Add genotypes: [v]

Check form Save action Save and shelf

2) In case you are performing an out-cross, clear one of the parent fields and start typing the name or the ID of the second substock. The autocomplete function will show you the list of available candidates. In a large facility with a lot of records, prefiltering by room, rack or fishline might help to reduce the number of results. In that case, toggle the **Prefilters** button to reveal the filters.

3) **Fill in the date** when the crossing was performed. This will become the **Date of birth** of the child substock.

4) You can set the **number of new child substocks and number of fish in each substock** right away, or you can leave the field empty which will leave the value as "not set". It is also possible to specify the **Sex** for each of the parent substocks by using the drop-down menu.

5) **Adjust the genotype**, if needed. When **outcrossing**, if there is no fishline with a genotype set matching the combination of parent fishlines, the user has to create a new one during the crossing. If there are some suitable fishlines, the user can pick one of them or force-create a new one. When **adding a non-parental genotype (for example by microinjection)**, specify it in the dedicated field.

6) **Save action**. By choosing **Save and shelf**, the newly generated substocks will be placed on the shelf in facility view, so they can be easily dragged to their new position.

In the Premium plan, the **quality and quantity of eggs** can be rated to keep track of your crossings. The predefined egg quality levels are: bad, mostly bad, average, and good. The egg quantity levels are no eggs, low, normal, and high. The rating can be amended retrospectively by the admin (in Admin/ Action log / Action type / Crossing / Select a specific crossing / Update). In the list of substocks, you can display either the quality and quantity of eggs in the last three crossings separately, or a general indicator that shows the average calculated from the last three crossings.

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Use	Statusbar	IO	α°	Crossings
WT (AB) 7352	AB		21	2023-07-18	productive	20	F07:E04		facility	Bartunek Core Fishlines					2024-02-29 2024-01-05 2023-12-20
WT (AB) 7352	AB		6	2023-07-18	productive	20	F07:E05		facility	Bartunek Core Fishlines					2023-12-12
WT (AB) 7352	AB		5	2023-07-18	productive	20	F07:E03		facility	Bartunek Core Fishlines					2024-02-29 2024-02-16 2024-02-15
WT (AB) 7352	AB		2	2023-07-18	productive	20	CD6:B09		facility	Bartunek Core Fishlines					2023-12-12
WT (AB) 7351	AB		2	2023-07-18	productive	19	CD6:B10		facility	Bartunek Core Fishlines					2023-12-12

Split

This action allows splitting a substock into multiple smaller ones. You can assign additional information to the newly generated substocks, for example zygosity, positions, or owners.

- 1) **Select the substock** either in the **Fish** or **Facility** menu. Select **Split** in the action menu.
- 2) **Fill in how many substocks you want to separate** from the source substock and how many fish per substock there should be. You can also enter owners, projects, and workgroups, if they are the same for all newly generated substocks.

Split

Substock: 7284-15908 CASPER 2023-06-14/1 (1) F08:D05 15

Performed: 2024-03-05 Note:

Substocks to split off: 1 Fish per substock: 1

14 fish will be left in the original substock.

Owners of the new substocks: Start typing... Project for the new substocks: Start typing... Workgroups for the new substocks: Start typing... CD Lab

Detailed split Check form Save action Save and shelf

3) Pick **Detailed split**, if you want to assign information unique for each substock, for example position, fish count, or owners. An extended form listing all newly generated substocks appears.

The screenshot displays the 'Split' form interface. At the top, the substock ID '7284-15908' and name 'CASPER 2023-06-14/1 (1)' are shown. The 'Performed' date is '2024-03-05'. Below the form, there are navigation options: 'Simple split' and '+ Add substock form'. A red-shaded section contains a checked option 'Perform genotyping as well', an 'ARM Project' dropdown set to '101/2021 - Manipulace', and a 'Genotyping method (optional)' dropdown set to '-- none --'. A summary bar shows '1' substock with 'count 1'. The 'Position' field is set to 'no change'. The 'Productivity' is 'productive'. Under 'Genotyping', there are two rows for 'mitfa^w2?' and 'roy^s9?' both set to 'unspecified (?)'. At the bottom right are buttons for 'Check form', 'Save action', and 'Save and shelf'.

4) Confirm by **saving the action**. Pick **Save and shelf** if you wish to put the substocks on the shelf in facility view.

If you are splitting fish because of genotyping or sorting, you can assign zygosity to any genotype that is assigned to that particular fishline in the detailed split form. You can also terminate any of the new substocks and record the number of fish you have euthanized.

The position can be either written down in the input field, or you can click the magnification glass icon to initiate the positional widget.

IMPORTANT: If the number of fish in the new substocks reaches the number in the original substock, the source substock will be terminated and will obtain the status “split out”. This warning will be displayed:

The original substock will be destroyed - you are taking all the fish from it.

If the sum exceeds the available fish count, there is another warning with the option to auto-adjust the original substock count before splitting. The original substock will be assigned as "split out".

Sum of fish in new substocks (15) exceeds total fish count in original substock (7). Please modify the counts.

It's OK, the count isn't precise.

Merge

Merging serves to combine two substocks coming from the same stock into one. Only sibling substocks that are **alive and not in an experiment** can be merged. In the transfer mode, it is possible to perform merge by drag and drop.

Transfer

Transfer can be used to change the position of your substock, as well as assign and unassign it. You can also drag and drop the substock when in the facility view to change its position.

1) **Select the substock** you want to transfer either in the **Substocks** list or the **Facility view**. This can be a substock without a position or one already positioned within the facility. Click **Transfer**.

2) **Choose the desired position** in the position widget.

4) Save the record by clicking **Save action**.

If you choose a position that is already assigned to a substock, the position will be shared by both substocks. If you wish to perform another operation, like e.g. swapping of the positions or removing the original tank from its position, use the **Transfer mode** in facility view.

Productivity

This function allows you to change the **productivity tag** of your substocks. Fish can be reported as **productive** or **unproductive**. By default, new progeny substocks are **unproductive** until they are reported **productive**. Productive substocks can also be reported during crossing, by ticking the **Change productivity of parents to "productive"** toggle.

1) **Select the substock either in Fish or in Facility view.** Select **Productivity** in the action menu.

2) **Fill in the date.**

3) **Change the tag.** The options are **Unproductive juvenile** and **Productive**.

4) **Save the record.**

Genotyping

Genotyping assigns zygosity (+/+, +/- etc.) for each of the genotypes present in a substock. It also allows to specify the genotyping method and report the procedure when the ARM module is active.

1) **Select the substock either in the Fish or Facility menu.** Select **Genotyping** in the action menu.

2) **Select zygosity** for each of the genotypes from the drop-down menu. When a new substock is generated, its default zygosity is **unspecified (?)**. This can be changed to any of the following options:

sorted - negative (-)	At least one allele is negative
sorted - positive (+)	At least one allele is positive
homozygote - transgene (+/+)	Positive homozygote, i.e. two gain-of-function alleles
homozygote - mutant (-/-)	Negative homozygote, i.e. two loss-of-function alleles
wild type (WT)	WT genotype
heterozygote (+/-)	One allele is positive and one negative

3) You can select the genotyping method that was used. New methods can be defined in Admin/Codebooks/Genotyping Method.

Name	Severity	Description	Active
fin clip (PCR)	mild	Fin clip to obtain a sample for PCR genotyping. Fish undergo anesthesia.	✓
sorting post 5dpf (microscope)	mild	Fluorescence sorting with anesthesia.	✓
sorting before 5 dpf (microscope)	below threshold	Fluorescence sorting. Age threshold for reporting not reached.	✓

4) If ARM is in use, it is obligatory to **specify the project** under which the procedure should be reported.

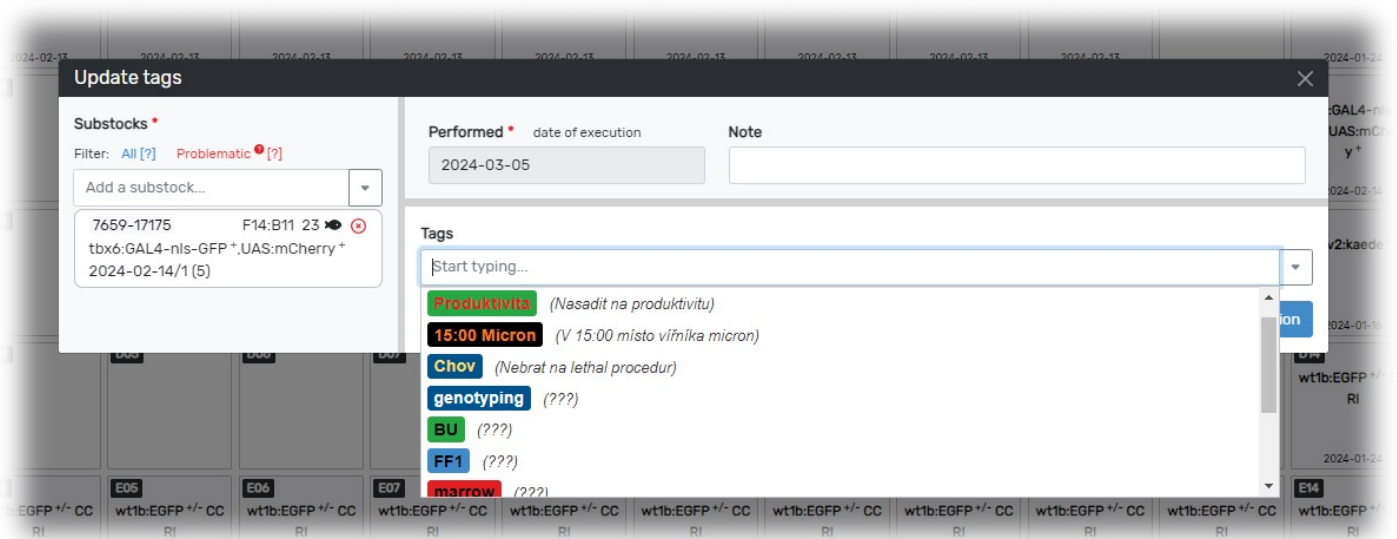
5) Save the record.

ARM project and genotyping method can be retrospectively amended by the admin in Admin/Logs/Action log/Action type: Genotyping.(Not possible for automated genotyping inputs.)

Tags

It is possible to add a tag to your substock, to associate some additional, quickly accessible information with it (such as low productivity, illness, single fish in tank...). Tags are visible either in substock detail next to the name of the substock, or in facility view.

New tags can be designed in Admin/Codebooks/Tag. You can define the name and the color of the tag, and write a short description. The name should be short (ideally one word). Tags are most useful when only one or two of them are associated with a substock.



Conditions

Diet and light regimes can be updated here. Define your custom values in Admin/Codebooks/Diet and Admin/Codebooks/Light.

Experiment

This is a part of the ARM module. You can add the substock to an experiment here, or create a new experiment for the substock.

2.3.6 Date of execution boundaries

Each action must have an associated date stating when it was performed. In the forms, this date is called Performed (date of execution). To preserve the logic of the database, the date has certain boundaries. The automatic limit is the date of birth of a substock - no action can happen before a substock has been born. There are additional boundaries specific to some action types:

- **count + terminate, split, merge** - limited by the date of the last split, merge, or count
- **transfer** - the date of the last transfer
- **genotyping** - the date of the last genotyping
- **productivity** - the date of the last productivity change

2.3.7 Batch actions

Some actions can be performed on multiple substocks at once, i.e. in a batch. You can initiate the batch mode either in the list of substocks, or in the facility view, by clicking **Batch actions**, or **Batch select**. After selecting the substocks, use one of the available actions: update substock, record count action, crossing, genotyping... Selected substocks can also be added to a marked list.

In Substocks, batch mode is switched off and the selection is wiped out after navigating to a different page of the list.

In order to perform batch actions, it is necessary to have the **Batcher** permission active. By default, it is only given to admins, but it can be assigned to any permission group in Admin/Permission groups.

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Project	Funding Use	Statusbar	ID
<input checked="" type="checkbox"/> WT (AB)	AB		1	2023-11-23	unproductive	not set	fx		facility	CD Lab Core Fishlines				7524-16703
<input checked="" type="checkbox"/> cd41:GFP +/-	AB		1	2023-11-21	unproductive	not set	fx		iva	CD Lab				7519-16694
<input checked="" type="checkbox"/> CASPER	AB		9	2023-11-15	unproductive	25	B01:D04 fx			CD Lab				7506-16686
<input type="checkbox"/> CASPER	AB		8	2023-11-15	unproductive	25	C03:E03 fx			CD Lab				7506-16685
<input type="checkbox"/> CASPER	AB		7	2023-11-15	unproductive	25	C03:E02 fx			CD Lab				7506-16684

2.3.8 The marked list

The marked list is a list of selected substocks that persists within the session (it is not accessible after logging out nor from a different computer). Only one marked list can be held in a session. A marked list can be created in the batch mode, either by selecting individual substocks and clicking **Add selected to marked list**, or by clicking **Add filtered to marked list** to add all filtered results.

The link to a marked list appears in the Fish submenu after it is created. The list can be used to perform batch actions on a larger number of substocks, or to export the data into an XLS file. **Batch actions called on a marked list are applied to all substocks in the list.**

The screenshot shows the ZebraBase web interface with a 'Marked list of substocks [4]' view. The interface includes a navigation bar with 'Fish', 'Facility', 'Calendar', 'Projects', and 'Stats'. Below the navigation bar, there are tabs for 'Genotypes', 'Fishlines', 'Stocks', and 'Substocks'. The main content area displays a table of substocks with columns for Name, BG, F-gen, Num, Birth, Status, Count, Position, Owners, Responsible, Workgroups, Project, Funding, Use, Statusbar, and ID. The table contains four rows of substocks, all with a status of 'unproductive'.

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Project	Funding	Use	Statusbar	ID
WT (AB)	AB		1	2023-11-23	unproductive	not set	fish	facility		CD Lab Core Fishlines					7524-16703
cd41:GFP ^{+/+}	AB		1	2023-11-21	unproductive	not set	fish		iva	CD Lab					7519-16694
WT (AB)	AB		12	2023-10-19	unproductive	20	C07:E03 fish	facility		CD Lab Core Fishlines					7469-16603
WT (AB)	AB		10	2023-10-19	unproductive	20	C07:E04 fish	facility		CD Lab Core Fishlines					7469-16601

2.3.9 Action log

The action log is a field in the substock, stock and fishline detail, that holds information on all performed actions, within the structure or its parents. For example, any actions performed on structures higher than substock (stock or fishline) will be projected into the respective substock action log as well. You can filter individual action types in the log, or view the detail of an action by clicking the icon on the right of the name. The **info** button opens a short version of the detail.

Action log of the whole facility can be viewed in Admin/Logs/Action Log. The list can be filtered by action type, exported, and individual actions can be updated (only some details can be changed, to sustain the logic of the database).

Observation log

The observation log serves to keep track of various information concerning the husbandry of your fish (for example about the health, treatment with medication, medical checks etc.). It is available for substocks, stocks, and fishlines. The field is designed to hold any written information - custom observation types can be designed in Admin/Codebooks/Observation Type.

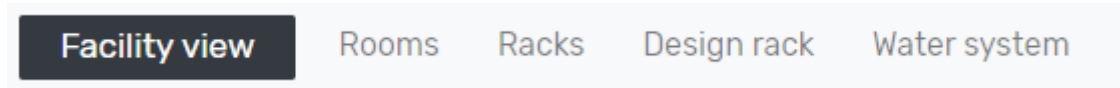
Observations are logged (include the date and the user who created it). In the admin action log, individual logs can be viewed and corrected.

2.3.10 Known bugs

- Available actions in substock menu don't reflect the user's permissions or substock status.

2.4 Facility

There are four (five) submenu options:



- **Facility view** - allows to explore rooms and racks and to perform some operations with substocks
- **Rooms** - standard list of rooms
- **Racks** - standard list of racks
- **Design rack** (visible only for users with rack_modify permission!)- this feature allows users to design a custom rack
- **Water system** - allows to browse water systems of the facility

Facility View

The **Facility View** is a display of substocks in tanks, organized as in the actual facility. It is accessible through the main menu, or by clicking on the position information in the substock detail - this will navigate the user to the Facility View and highlight the inspected substock with a red frame. The settings can be personalized by clicking on the cog wheel in the upper-right corner, and include hiding certain rooms or racks or setting the default rack to view after accessing the page.

The screenshot shows the Facility View interface with the following annotations:

- Occupancy indicator**: Occupied / Blocked
- Color-coding status**: Status dropdown menu
- Mode selection**: View, Transfer, Batch Select, Block buttons
- Rooms**: B, C, L, Q tabs
- Racks**: 01, 02 tabs

Room	Rack	Substock	Status	Occupancy
A01				
A02				
A03				
A04				
A05				
A06		WT (AB)		
A07		WT (AB)		
A08		gata1:DsRed ⁺		
A09				
A10				
B01				
B02				
B03		p53 ^{M214K} +/-		
B04		WT (AB) 6794		
B05		WT AB (DR)		
B06		WT (AB) 6540		
B07				
B08		gata1:kalt4 ⁺ UAS:kae de ⁻		
B09		gata1:kalt4 ⁺ UAS:RFP ⁺		
B10				
C01				
C02				
C03		CASPER		
C04		p53 ^{M214K} +/-		
C05		p53 ^{M214K} +/-		
C06		p53 ^{M214K} +/-		
C07		cd45:DsRed ⁺		
C08		cd45:DsRed ⁺		
C09				
C10				
D01		gata1:kalt4 ⁺ UAS:kae de ⁻		
D02		gata1:kalt4 ⁺ UAS:kae de ⁻		
D03		gata1:kalt4 ⁺ UAS:kae de ⁻		
D04		CASPER		
D05		CASPER		
D06				
D07				
D08				
D09				
D10				
E01				
E02				
E03				
E04				
E05				
E06				
E07				
E08				
E09				
E10				

Color-coding of tanks

There are multiple color-coding options to help with navigating the facility. The blue **Status** icon in the upper right corner of the Facility View opens a drop-down menu that serves to select the according scheme:

- Status of the fish (unproductive, productive, retired, dead)
- Maturity (fry, juvenile, adult, retired)
- Diet
- Species
- Light regime
- Count (fish density)
- Experiment
- Substock use (experiment, breeding, backup...)
- Printing status (which tanks have their label printed already, or need a reprint)
- Egg count (average from the last 3 crossings)
- Egg quality (average from the last 3 crossings)
- Price

A legend for the color-coding is displayed on the right panel and can be hidden if desired. Some values used for color-coding can be defined by admin users, for example, the Count (fish density) values can be modified in Admin/Configuration/substock.count and also for the specific species in the Species extension in Admin/Codebooks/Species.

Mode selection

The four (five) buttons in the upper right corner change the functionality of the Facility View.



- **View** - this is the default mode for Facility View, clicking on tanks makes it possible to explore the contents and update the properties of the tank
- **Transfer**- substocks can be moved between tanks using drag-and-drop
- **Batch Select** - allows the selection of multiple substocks, to perform operations with the whole batch in one go
- **Block** - allows to block (i.e. book) empty tanks for a specified user
- **Pricing** (visible only for users with db_admin permission!)- a defined price can be assigned to selected positions and any tank placed in the priced position will be included in the billing statistics. **NOTE:** The price in a specific position does not change by assigning a new substock to it and needs to be redefined manually.

View (Shift+V+V)

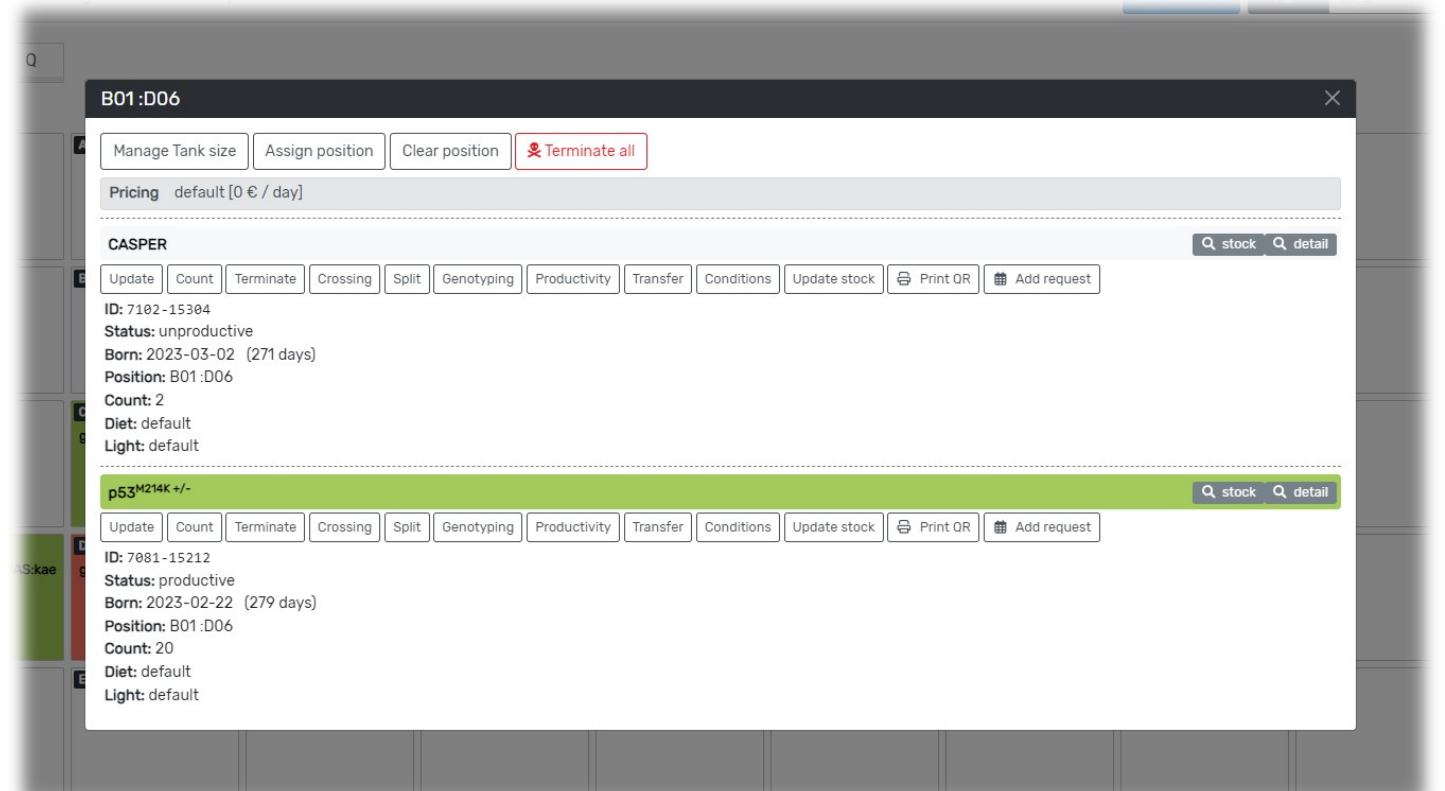
In this mode, users can select and assign an empty position or select an occupied tank and update its properties. The detail tooltips on tanks (accessible by hovering the cursor over a tank) contain substock name, status, fish count, owners, and projects.

Action window opens by clicking on a tank:

Empty tank:



Occupied tank: (two substocks sharing one position)



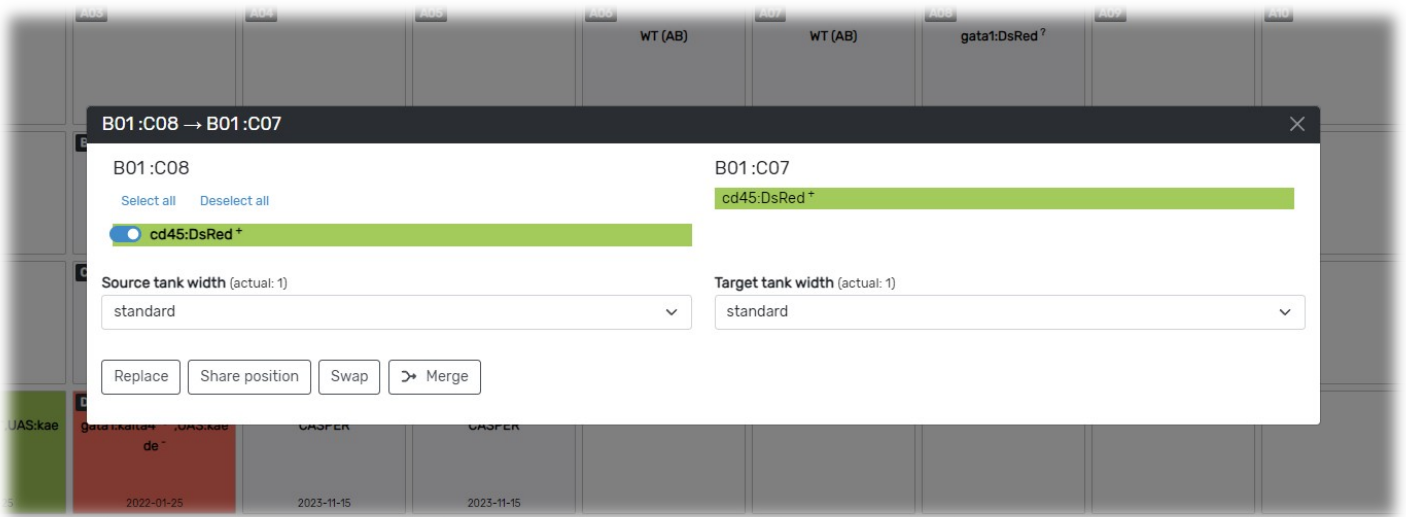
Transfer (Shift+T+T)

This function allows transferring the substocks between fish tanks using "drag and drop". A tank can be moved to any position which is not blocked by another user.

After drop

After dropping the tank to a selected position, the user can specify:

- Which substocks should be transferred (checkboxes on the left side)
- Width of both target and source tanks
- If the target position is empty, the only transfer option is Move
- If the target position is already occupied, the user can choose between:
 - Replace (unassign all substocks from the target position; these substocks lose their position information)
 - Share position (add selected substocks from source position to target position)
 - Swap (positions of target and source tanks will be switched)
 - Merge (merge into one substock; this option is available only if the source and the target contain sibling substocks)



In case users want to redistribute substocks from a shared tank into new positions, it is advisable to use the **Shelf** feature:

Shelf

The **Shelf** is a drag-and-drop feature that can be used to transfer substocks between different racks and rooms. It opens automatically once the user enters the Transfer mode. Dropping a tank on the shelf makes it possible to move the tank into a new position in a different rack. There is also a drop-down menu Action, allowing the user to perform actions on all substocks placed on the shelf (Show on list, Crossing, Clear positions, Print QR, Clear shelf). The fish icon displays the number of substocks on the shelf. In Facility View, substocks currently placed on the shelf will be marked by a black flag at the right edge of the box. A substock placed on the shelf remains in its original position until the transfer to another position is completed.

The screenshot displays the 'Facility view' interface with a grid of tanks (A01-E10). Annotations include:

- Flags**: Red boxes on tanks A08 and A09, with a note: "Flags indicate which substocks are put on shelf".
- The Shelf menu**: A dropdown menu on the right side of the interface.
- Number of substocks put on shelf**: A red arrow points to the '4' in the shelf menu header.
- Action menu**: A red arrow points to the 'Action' button in the shelf menu.

Batch Select

Batch Select allows to select multiple tanks and execute batch actions. Tanks can be selected by clicking on individual positions, or by ticking the box on the left side of each row, which selects all tanks in the specific row. Batch actions can be selected in a drop-down menu in the right panel and include updating the substock information, printing the QR codes, and performing crossing. Function Count can be used for setting the initial Start count and further updating the count of fish in each tank of the whole batch in one go. **NOTE:** The action Count in Batch Select mode cannot be performed on a substock that is currently assigned to an experiment.

The batch-specific functions are:

- List (creates a custom list of substocks)
- Export (downloads a spreadsheet containing a custom list of substocks)

Selected tanks
are highlighted with a red outline

Batch actions menu

Block (Shift+B+B)

This function is used to block a tank for use by a certain user. It is possible to select only empty or blocked tanks and not the occupied ones. **Block** will block all unblocked tanks from the selection, **Unblock** will unblock all blocked tanks from the selection. A note which will stay visible on the blocked tanks can be added.

The screenshot displays the Zebrafish Facility management interface. At the top, there are navigation tabs for Fish, Facility, Calendar, Projects, and Stats. Below this, a 'Facility view' section shows various system components like Rooms, Racks, Design rack, and Water system. A status bar indicates 'Blocking: 1 succeeded, 0 failed.' Below the status bar, there are buttons for 'Status', 'View', 'Transfer', 'Batch Select', 'Block', and 'Unblock'. The main area is a grid of tanks, each with a unique ID (e.g., A01, B01, C01, D01, E01) and a status. Three tanks (A02, A03, A04) are highlighted with a red outline and labeled as 'Blocked tanks'. A note is added to tank A04, labeled 'Note can be added during blocking'. Two tanks (A06, A07) are also highlighted with a red outline and labeled as 'Selected tanks are highlighted with a red outline'. On the right side, a 'Blocking menu' is open, showing options to 'Block' or 'Unblock' selected tanks, with a list of blocked tanks (B01 :A06-free, B01 :A07-free) and a legend for tank status (unproductive, productive, retired, dead) and tags (10:00 Micon, bad eggs, Chow, FFS, genotyping, low eggs, normal, unproduct).

Managing tank size

A single tank can be occupied by more substocks (a shared tank), or it can be split into a matrix of smaller subtanks. There is also the option to set the tank width to 2x, 3x, or 4x of the standard width. This can be accessed by clicking on the **Manage Tank Size** button in the tank detail.

The screenshot displays the Zebrafish Facility Management System interface. The main area shows a 5x10 grid of racks, each with a unique identifier (e.g., A01, B01, C01, D01, E01) and associated data such as genotype, date, and status. A legend on the right side provides a key for status (unproductive, productive, retired, dead) and tags (16:00 Micron, bad eggs, Chew, CTR, GFP, genotyping, low eggs, unproduc.). Three specific racks are highlighted with red boxes and arrows:

- 2x wide tank:** Rack E03, labeled 'gata1:DsRed ?' with date 2023-10-11.
- Shared tank:** Rack D06, labeled 'CASPER p53^{M214K} +/-' with date 2022-11-03. A red box around the '2' in the substock count indicates the number of substocks in a shared tank.
- Split tank:** Racks C08 and C09, both labeled 'WT (AB)' with date 2022-10-10.

Design rack

(visible only for users with rack_modify permission!)

Step 1 - Design rack

- The default design is 5 rows x 10 columns
- Racks can be edited either manually, or by directly changing the code in the **Generated code** field below the rack (it is not necessary to edit the code)

Step 2 - Fill in the name and set room of the new racks

- The user can design multiple racks of the same design at once by clicking the button **Add rack form**, then filling the forms and hitting **Save**

Delete tank

Step 1: Design rack structure

Delete row

Add tank

Move row up/down

Step 2: Specify racks

Name *

Room *

Description

Save

Water system

Here, you can define water systems used in the facility and specify the value ranges of the basic parameters for each system (pH, conductivity, nitrites, nitrates, and ammonia). Users with db_admin permission are able to define new water systems in Facility / Water system. Individual racks can be ascribed to a water system in the Racks submenu.

Name	Racks	Temperature	pH	Conductivity	NH ₃ [mg/l]	NO ₂ [mg/l]	NO ₃ [mg/l]	Last Log	Active
Fish facility I	☰	27.9 26.0 - 29.0	7.3 6.0 - 8.0	691.0 300.0 - 1500.0	0.0 0.0 - 0.1	0.05 0.0 - 0.3	24.7 0.0 - 25.0	2023-11-02 13:00 [UTC]	✓
Back up II	☰	7.0 26.0 - 29.0	27.5 6.0 - 8.0	698.0 300.0 - 1500.0	0.0 0.0 - 0.1	0.04 0.0 - 0.3	27.6 0.0 - 25.0	2023-11-02 13:00 [UTC]	✓
Karanténa 1	☰	27.8 26.0 - 29.0	6.93 6.0 - 8.0	704.0 300.0 - 1500.0	0.0 0.0 - 0.1	0.03 0.0 - 0.3	0.0 0.0 - 25.0	2022-11-17 18:00 [UTC]	✓
Fish facility II	☰	27.9 26.0 - 29.0	7.05 6.0 - 8.0	700.0 300.0 - 1500.0	0.0 0.0 - 0.1	0.0204 0.0 - 0.3	15.0 0.0 - 25.0	2023-11-23 14:28 [UTC]	✓
Back up I	☰	27.1 26.0 - 29.0	7.0 6.0 - 8.0	707.0 300.0 - 1500.0	0.0 0.0 - 0.1	0.0 0.0 - 0.3	8.4 0.0 - 25.0	2023-11-02 13:00 [UTC]	✓

Once you have a new system stored and associated with corresponding racks, you can store values for each water quality parameter and track them in time - go to the detail of a selected water system and click the **Log water status** button.

Add new water quality log

List of previous logs

Back to list Log water status

Name	Back up II
Description	
Temperature	7.0
Temperature range	26.0 - 29.0
pH	27.5
pH range	6.0 - 8.0
Conductivity	698.0
Conductivity range	300.0 - 1500.0
NH ₃ [mg/l]	0.0
NH ₃ [mg/l] range	0.0 - 0.1
NO ₂ [mg/l]	0.04
NO ₂ [mg/l] range	0.0 - 0.3
NO ₃ [mg/l]	27.6
NO ₃ [mg/l] range	0.0 - 25.0
Last Log	2023-11-02 13:00 [UTC]
Racks	801 802

Last data see full list

Timestamp	Temp.	pH		Cond.
	NH ₃	NO ₂	NO ₃	
2022-05-12 13:00 [UTC]	28.0	7.0	699.0	
	0.22	0.079	33.3	
2022-05-19 14:00 [UTC]	27.8	7.0	706.0	
	0.0	0.02	20.0	
2022-05-26 13:00 [UTC]	27.9	7.0	698.0	
	0.04	0.026	36.8	
2022-06-02 13:00 [UTC]	27.9	7.0	693.0	
	0.03	0.053	25.4	
2022-06-09 13:00 [UTC]	27.9	7.0	712.0	
	0.13	0.076	31.3	
2022-06-16 13:00 [UTC]	28.0	7.0	703.0	
	0.06	0.033	9.7	
2022-06-30 13:17 [UTC]	28.0	7.0	691.0	
	0.05	0.023	39.1	
	28.0	7.0	698.0	

NOTE: Uploads of water logs exported from the water control units will be available in one of the next releases.

2.4.1 Known bugs

- Actions available in the action menu accessible in view mode do not reflect the user's permissions nor substock status (all actions are available although they cannot be saved in case the users do not have permissions for it).

2.5 Calendar and requesting

The calendar serves as an interface for planning experiments and other procedures. The user can request actions from the facility staff and other users, or plan an action or a task for oneself. To create a request, click **Add request** in the Action menu or click the **Plus button** in the calendar. By default, all users can add requests, but this right can be restricted for each request type in Admin/Permission groups.

It is possible to choose from the following request types:

- **Setup fish for crossing** - Transfer fish to the breeding tanks on a specific date.
- **Crossing** - Cross the fish on a specific date (this generates **two requests**: a request to setup the fish for crossing one day ahead, and a request for crossing on the date specified in the request).
- **Fish** - Produce fish of a desired age by a certain date. Requests for crossing and setup for crossing will be generated automatically accordingly to the requested fish age.
- **Transfer** - Transfer a substock to a defined position in the facility.
- **Terminate** - Terminate a substock in the facility on a certain date.
- **Import** - Import a fish stock/fishline from a source outside the facility.
- **Export** - Export a fishline/stock to a different facility.
- **Genotyping** - Perform genotyping on the selected substock.
- **Mobilize** - Special treatment of a fish substock for a defined period, e.g. special feeding regime or mating. The treatment should be described in detail in the Description field.
- **Custom** - This request type allows the specification of any custom request.

2.5.1 Browsing

Calendar view

The calendar can be displayed in the form of a **weekly or monthly view**. A filterable list of all requests is available in the **Requests** submenu.

The screenshot displays the Zebrabase web interface for the 'Calendar' section. The top navigation bar includes 'zebrabase', 'Fish', 'Facility', 'Calendar', 'Projects', 'Stats', and 'Admin'. The user is logged in as 'Test User' with the role 'ZB3-ARM'. The main content area shows a calendar for March 2024. A red box highlights the calendar icon in the top left, with the annotation 'Switch to weekly/monthly view'. Another red box highlights a plus sign button in the top right, with the annotation 'Add new request'. A third red box highlights the 'show all requests' button in the top left, with the annotation 'show all requests'. The calendar grid shows events for various dates, including 'WT (AB)' on March 28th and 'Pax/GFP*' on March 4th.

Filtering

The default view shows only requests relevant to the specific user, i.e. the user's assignments, submissions, or general entries not assigned to any specific user (i.e. available for any user to accept). To change this, click **Show all requests** in the top left corner.

You can search for requests based on the assignee, submitter, request status, room, or request type. It is also possible to search for keywords in the request descriptions.

The screenshot shows the Zebrabase interface with the 'Calendar' view selected. The top navigation bar includes 'Filter' and 'Show all / show relevant' buttons. Below the navigation bar, there are filter controls for 'View', 'Description contains', 'Assignee', 'Submitter', 'Omit auto-generated requests', and 'Status'. The 'Status' section includes checkboxes for 'assigned [24]', 'declined [350]', 'performed [4084]', 'accepted [300]', and 'unassigned [161]'. The 'Request type' section includes radio buttons for 'fish [226]', 'transfer [1]', 'custom [264]', 'crossing [1987]', 'setup [2261]', 'genotyping [69]', 'terminate [81]', and 'mobilize [30]'. The main area displays a calendar for March 2024, with events shown as colored bars on specific dates.

2.5.2 Workflow

When a new request is entered, it will be displayed in the calendar. A notification will be sent to in-app or other channels (see more in [Notifications](#)). In case the assignees are specified, the notification will be sent only to them, otherwise, the notification will reach all users.

Statuses

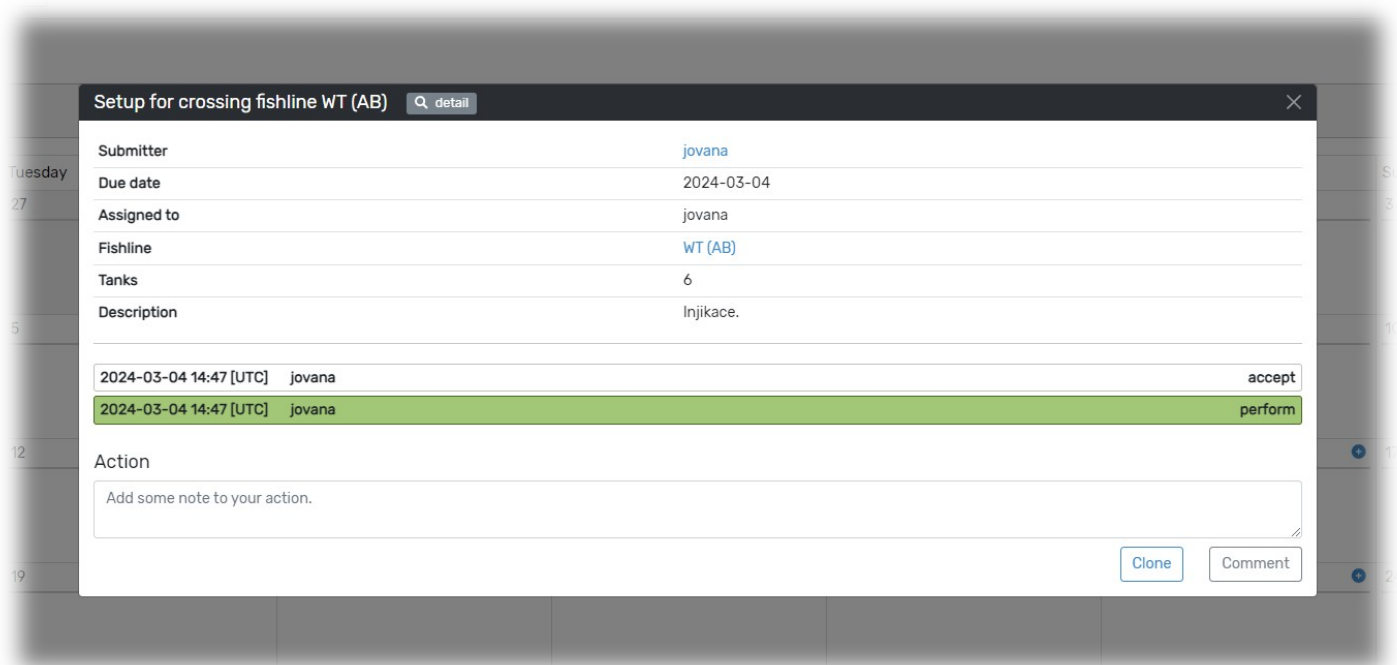
By default, all users are allowed to change the status of any request (i.e. accept, perform, decline, or unassign themselves...). The statuses are color-coded in the following manner:

- **grey** - new, unassigned request
- **light blue** - request assigned to a specific user(s)
- **blue** - accepted request
- **green** - performed request
- **red** - declined request

A **yellow bar** on the left side of the request indicates a request associated with the user (it has been either entered by this user or assigned to him/her).

Further information about the request (i.e. the requested task and the assignee) can be viewed either in the request detail or by hovering over it with a cursor.

The changes in the status are visualized in the request detail, including user comments. Comments can be added to a request independently of any action, or can be connected to a specific action. For the action **Decline**, it is mandatory to enter an explanation before saving the action.



2.6 Statistics

Statistics displays an overview of your facility's fish stocks. Using the export button in the top right corner you can export the data to a .xls file based on the selected filters.

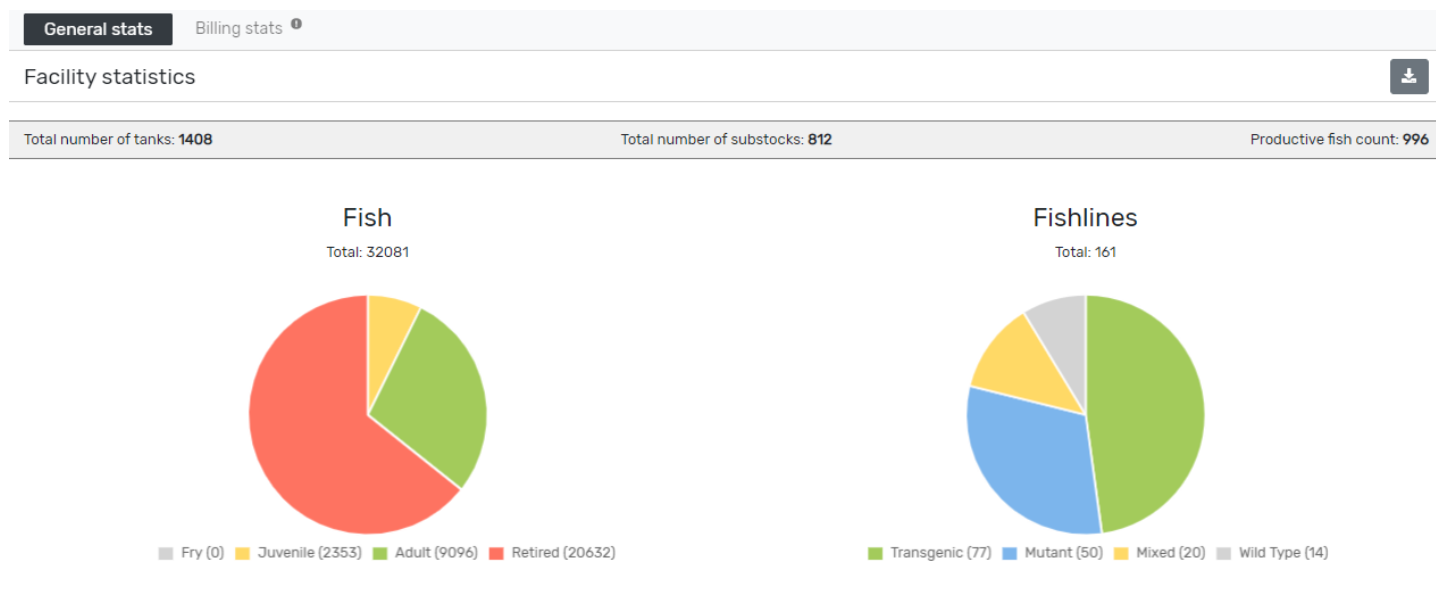
2.6.1 General stats

General overview

- **Total number of tanks** The number of tank positions available in your facility
- **Total number of substocks** The total number of all live substocks
- **Productive fish count** The total number of productive fish in the facility

Pie charts

- **Fishlines** based on the genotype (transgenic, mutant, mixed, wild-type)
- **Fish** based on age (fry, juvenile, adult, retired)

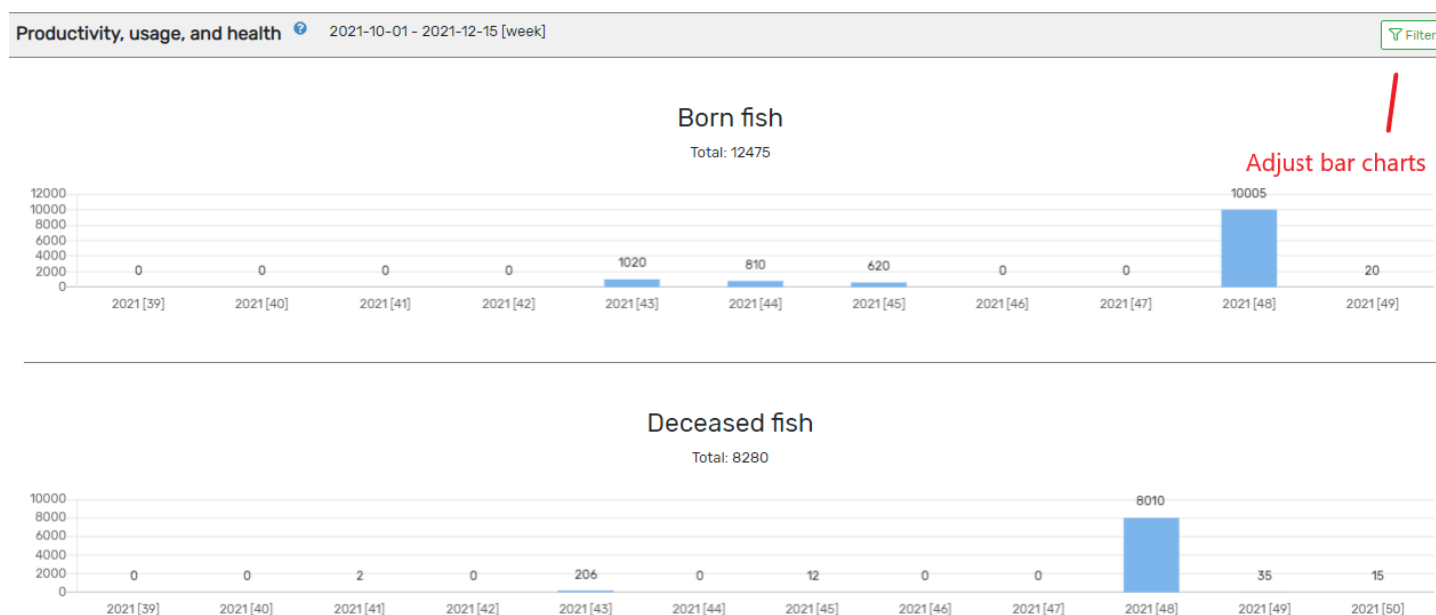


Bar charts

By default, bar charts are showing monthly statistics for the last 12 months.

- **Born fish** the number of animals based on the initial counts of substocks with the corresponding date of birth
- **Deceased fish** the total number of fish reported as "found dead"
- **Euthanized fish** the total number of fish reported as "euthanized"
- **Fish used for experiments** the total number of fish used in an experiment, can be further refined by defining the desired experiment in the filter.

To change the bar period or filter for any further specifications, click on the **Filter** button next to the **Productivity, usage, and health** heading. Additional details can be defined here.












Custom bar charts

added in v. 3.1.7

Other count types can be included in the statistics by setting the count types parameter `custom stats` to `True` (go to Admin > Codebooks > Count Type). Every count type with `custom stats = True` will then have its own bar chart in stats (named Custom stats: <count type name>).

List of count types

≡ 50 ▾ « 1 » Σ 9 / 9 Deleted items Add count type ⚙️

Name	Type	Can be set by user	Is initial count type	Needs experiment	Needs method	Custom stats	Active	
start count	set	✓	✓	✗	✗	✗	✓	
died	sub	✓	✗	✗	✗	✗	✓	
euthanized	sub	✓	✗	✗	✗	✗	✓	
used for experiment	sub	✓	✗	✓	✗	✗	✓	
split out	sub	✗	✗	✗	✗	✗	✓	
merge	set	✗	✗	✗	✗	✗	✓	
correct	set	✓	✗	✗	✗	✗	✓	
split start count	set	✗	✓	✗	✗	✗	✓	
special count type	sub	✓	✗	✗	✗	✓	✓	

Filtering

[General info about filters](#)
[How filters work - useful if you need something special](#)

Filters can be combined except for the WT/non-WT switch and genotypes.

Bar chart data can be filtered based on any string included in the substock name by adjusting:

- **Name contains**

For the bar charts, it is further possible to select:

- **Fishline** - one or multiple fishlines
- **Responsible user** - one or multiple users who are associated with the corresponding fishline
- **Fishline background** - one or multiple fishline background(s)
- **Modification category** - one or multiple categories of the associated genotypes
- **Marker gene** - one or multiple genes associated with the corresponding transgenic genotypes (e.g. GFP)
- **Date of birth** - to adjust the period for DOB
- **Experiment** - one or multiple experiments
- **Rooms** - rooms based on current substock positions
- **Racks** - racks based on current substock positions
- **Workgroups** - one or multiple workgroups, based on the current workgroups associated with substocks
- **Owner** - one or multiple owners, based on owners associated with substocks (current state)
- **Project** - one or multiple projects, based on projects associated with substocks (current state)
- **Genotypes** - one or multiple genotypes
- **WT/non-WT switch** - to show all data only for WT or only for the modified substocks
- **Status** - one or multiple statuses that should be displayed

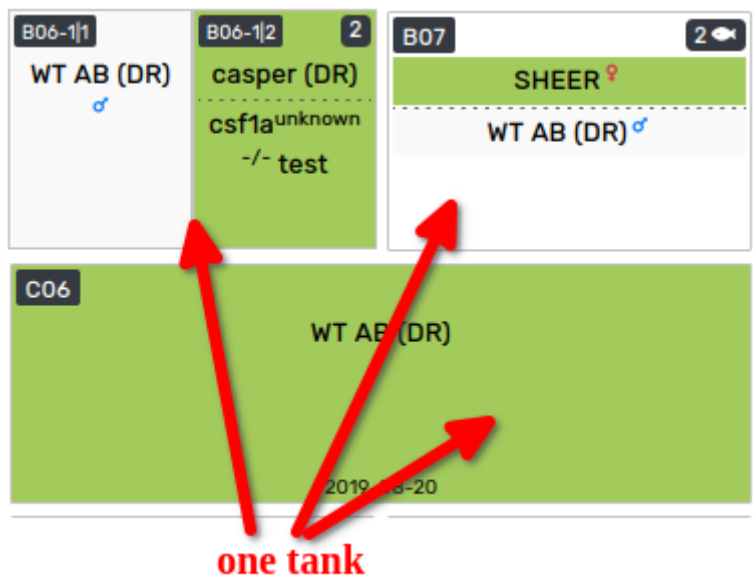
2.6.2 Billing stats

Generating billing statistics is slow so please be patient.

The billing statistics report the number of living fish and used tanks over a given period. The default period is the last whole calendar month; this can be changed in the **Filters**.

Tank counts (!)

- filtering by room provides only the counts of fish placed in the selected room
- wider tanks are considered single tank
- split or shared tanks are considered single tank
- if a tank is occupied by two substocks owned by different workgroups and billing stats are calculated for the workgroups separately, this tank will be counted twice (once for each of the workgroups)



Tank pricing

added in v. 3.2.0

The price per day can be set for each tank (for admins only).

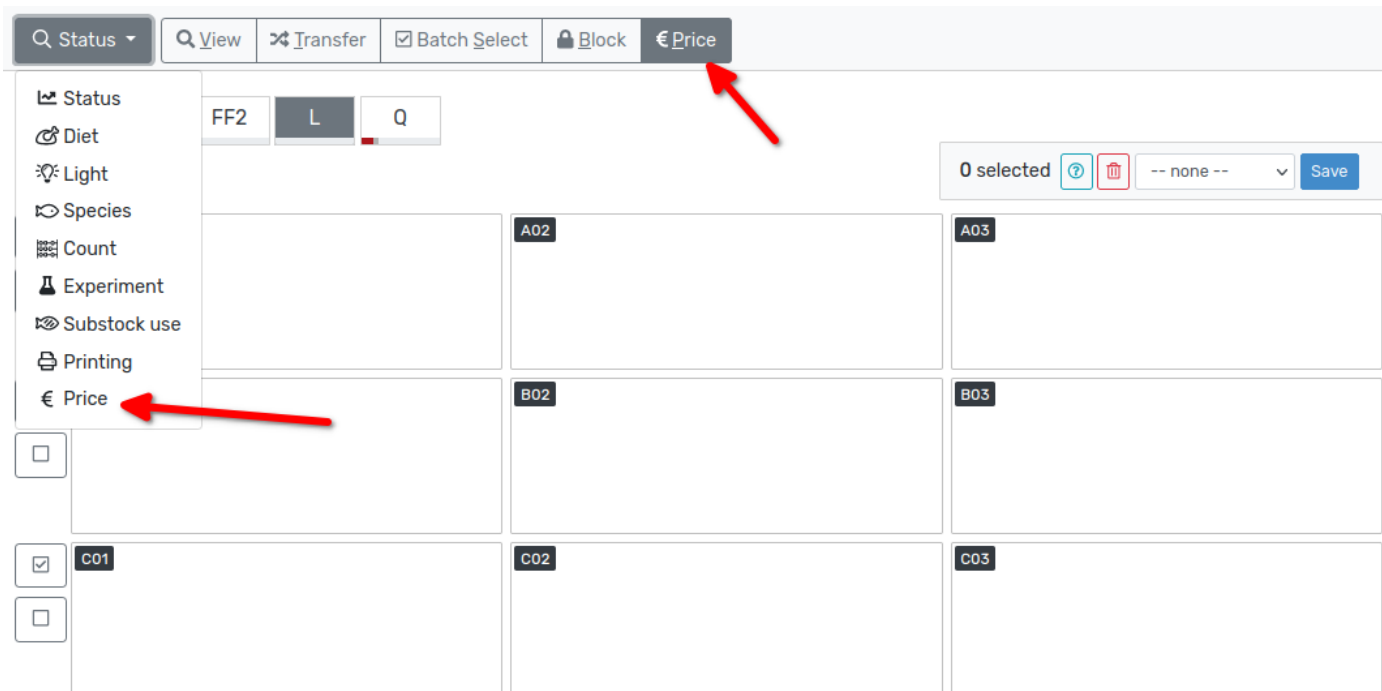
1. **prepare predefined prices - Admin / Codebooks / Others / Position Price:**

prepare some predefined prices that can be attached to the tanks

2. **set default price - Admin / Configuration / `defaults.position.price`**

3. **assign prices to tanks - Facility view:**

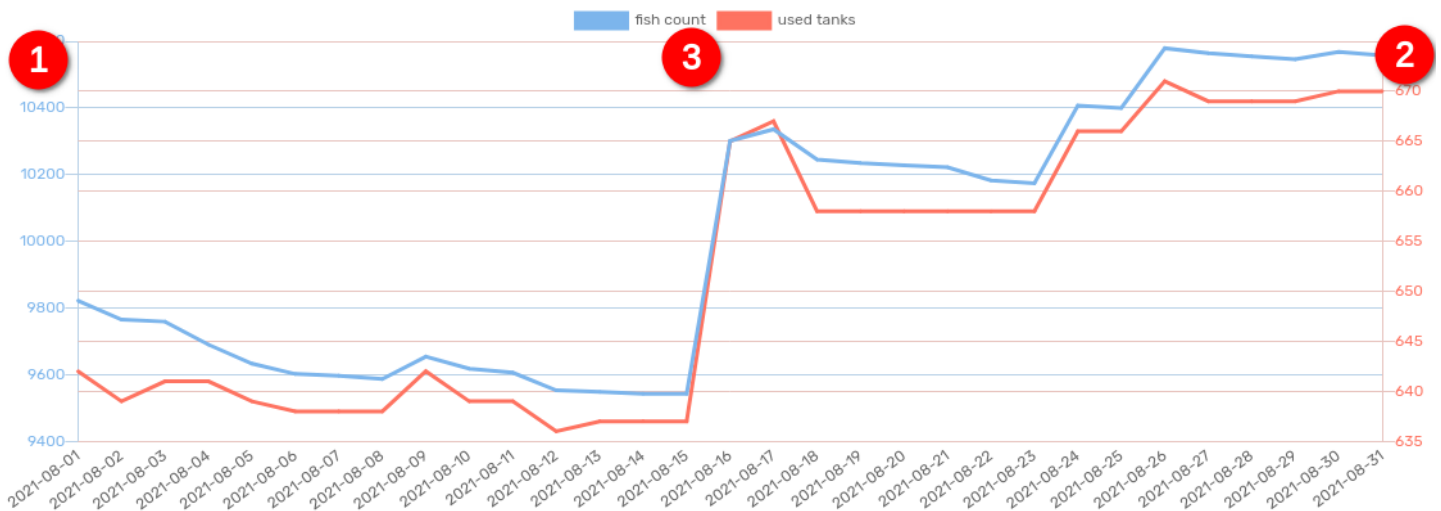
use scope "Price" to visualize and mode "Price" to edit (you can set a price for both empty and occupied tanks)



Predefined price cannot be assigned retrospectively!

When the default tank price is changed, the change will affect the pricing in general disregarding when the actual change was performed. However, if the predefined tank price is reassigned, the tank price will change starting from the date the change was performed (changing the value of a predefined price works retrospectively).

Billing statistics graph



1. fish count scale
2. used tanks scale
3. legend - clicking on a dataset toggles its visibility

Plans for future versions

- billing based on requests

2.7 Useful tips, shortcuts, etc.

2.7.1 Use cases

Use your mobile device to perform substock actions

Zebrabase is designed to be fish-facility-friendly. The website is composed to work on any kind of device, so many actions can be performed directly on-site with a **mobile phone or a tablet**, without having to go to your PC. You can get the most out of it if you use the **QR labels** provided by Zebrabase - the QR code leads to the detail of the specific substock, where you can perform any action you need right away!



Substock #16877 - WT (AB) 2023-12-05/3 7545 (8)

🗑️
✍️
📄 stock
☰
✉️
🖨️

Σ
☠️
♀️
↩️
➡️
⏳

P
👤
⚙️
📅
🐟
🧪

Species	
SSID	16877
Status	unproductive 🐟
Tags	
Stock	WT (AB) 2023-12-05/3
Num	8 / 9
Fishline	WT (AB)
Suffix	7545
Background	AB
Position	C05:D06
F-gen	
Use	
Funding	
Count	10
Birth	2023-12-05
Death	not yet
Most severe procedure	
Parents	🔴 WT (AB) 2022-06-02/2 6471

Count
✕

Substocks *

Filter: All [1] Problematic [0]

Add a substock...

7545-16877 WT (AB) 2023-12-05/3 7545 (8)
C05:D06 10 🐟
🗓️ 143

Performed * 📅 date of execution

2024-04-26

Note

Count type *

died after 90dpf

Count *

2 📄 🔴 all fish *

Check form
Save action

Terminate
✕

Substocks *

Filter: All [1] Problematic [0]

Add a substock...

7545-16877 WT (AB) 2023-12-05/3 7545 (8)
C05:D06 10 🐟
🗓️ 143

Performed * 📅 date of execution

2024-04-26

Note

Count type *

ethanized (healthy)

Method *

Start typing... ✕

Reason *

Start typing... ✕

Check form
Save action

Use the substock/stock/fishline suffixes

Zebrabase keeps track of three types of name suffixes: substock, stock, and fishline. The suffix will be used in names of all subordinate substocks, i.e. stock suffix for all fish in the stock and fishline suffix for all fish in the fishline. You can use all three at the same time, and all will be displayed in the substock name.

Substock #16936 - CASPER 2023-12-13/1 fishline-suffix stock-suffix substock-suffix (2)

Delete
Update
Update stock
Back to list

Notify owners
Print QR

Σ

☠

♀♂

↔

➔

⌛

P

👤

⚙️

📅

👁

🧪

Species		
SSID	16936	
Status	unproductive	
Tags		
Stock	CASPER (fishline-suffix) 2023-12-13/1 stock-suffix	
Num	2 / 4	
Fishline	CASPER (fishline-suffix)	
Suffix	substock-suffix	
Background	AB	
Position	C02:C08	
F-gen		
Use		

Label 🔗

**CASPER fishline – suffix stock – s
uffix substock – suffix**

DOB: 2023 – 12 – 13

ID: 7581 – 16936

POS: C02:C08

Observation log +

There is nothing here, yet.

2.7.2 Keyboard shortcuts

Global

- **Esc+Esc** closes all alert messages

Facility

- **Shift+V+V** - view mode
- **Shift+T+T** - transfer mode
- **Shift+S+S** - batch select mode
- **Shift+B+B** - blocking mode

Forms

- **Ctrl+S** - submit form - doesn't work when the cursor is in a text field and also may work differently depending on the browser

2.7.3 URL filters

Replace the <prefix> field with the URL prefix of your instance.

- Filter for specific stock ID in Substocks:

```
https://<prefix>.zebrabase.org/list/substock?stock_id=655&_url=True
```

- Filter for specific substock ID in Substocks:

```
https://<prefix>.zebrabase.org/list/substock?id=500&_url=True
```

- Filter for specific substock ID in the billing statistics:

```
https://<prefix>.zebrabase.org/stats/billing?id=13069&_url
```

- Filter for specific stock ID in Count actions in Action log:

```
https://<prefix>.zebrabase.org/list/input_count?substock__stock_id=6552&_url
```

 (for checking how the fish numbers in the stock were changing throughout the time)

2.7.4 Excel spreadsheet

You can use this [Excel spreadsheet](#) for accessing and downloading certain lists (substocks, fishlines, ...) from Zebrabase.

2.8 FAQ

2.8.1 Racks

I was wondering if there is any way to **delete a rack**?

Go to Facility and select "Racks" in the top menu. Select a rack from the list and go to its detail by clicking its name, then click "Delete". This action is available only to users with the corresponding permission (admin and facility).

I needed to **edit a rack** that was customized manually but can't manage to do it. Is it possible to edit customized racks once already created?

Please create a new rack, move the substocks there, and delete the old one (this workflow may be helpful: batch mode → select a row or whole rack → put on shelf → transfer mode → move from the shelf to the new rack).

Is it possible to **move an entire rack** from one room to another?

Go to Facility/Racks, select your rack and click Update.

2.8.2 Actions

I need your expertise in how is best to proceed when we're **injecting** a construct not into WT lines but into a transgenic line. What is the best way to create such a fishline?

You have the option to add a genotype while crossing - in the Crossing form, click Add non-parental genotypes and select a genotype from the list (i.e. the genotype needs to be defined first in Genotypes). This will work for both WT and transgenic parents.

Is there a way of registering **crossings without generating a new substock**? Most of the crossings we do are to generate embryos for experiments, so they will not give rise to a new stock. Therefore, we find it useful to have records for the dates of crosses, but not to have associated stocks.

We recommend setting the “Number of fish in each substock” to 0 in the Crossing form - this will automatically create a dead substock which will not appear on lists nor in statistics.

I crossed one line to originate another substock and by mistake used the wrong parent lines. How can I **reverse this crossing** and change the parent lines that originated the substock?

You can delete the new substock and the crossing action and try it again. The link to the action detail is in Action log in the substock detail, or you can find it in Admin / Logs / Action log. The substock can be deleted by clicking the red Delete button in the substock detail.

2.8.3 Pricing & billing

Could you **explain your pricing**? I see a “flat fee” and a “per tank fee.” Do I combine these? Our facility has about 1000 tanks, with the possibility of adding more. Could you estimate our cost per year for the different plans, (Premium, Standard, Basic)?

The price is a combination of a flat fee and price per tank. The number of tanks is usually estimated as the actual size of your facility, however, can be lower if you don't use its whole capacity. For 1000 tanks, it would be 999 EUR per year for Standard or 1299 EUR for Premium. You can use the price calculator available [on our webpage](#).

How do I determine the **number of tanks** we should pay for?

The general rule of thumb is the actual size of your facility, however, we actually bill on the basis of live substocks (fish groups occupying one tank) in the database. This is because some facilities don't use their whole capacity, but still want to have all tanks defined in Zebrabase. So, you can only count the “active tanks” in your facility and pay for those.

The number of paid substocks should not be breached at any point of the year. Of course, it is always possible to add more substocks for the price in the [pricing table](#).

2.8.4 Users & passwords

I am trying to add a user, but he **cannot log in** to change his password to start.

You as an admin only fill in the username, email, and the first and the last name of the user. The user receives a randomly generated password by e-mail immediately after his account is created. If the email address is incorrect, the original password email is lost, but you (admin) can set the correct address by updating the user, and then the user can reset his/her password from the login page.

I managed to create users and assign them to workgroups, but I could not generate user accounts with unique **permissions and privileges**. Is this possible? So far, we are using only the admin login credentials.

You can assign a user to a permission group either when creating the user account, or in Admin/Users/go to user detail/Update. There are predefined permission groups (e.g. facility, admin, etc.), or you can create specialized groups depending on your facility's needs. This can be done in Admin/Permission groups.

I would like to **create a new user group**. People from our department will access my database to get fish counts and tank counts. Could I create a group that would have permission to access all data, but can't edit anything?

You don't need any special permissions to read the data, so it is possible to leave these users without any group. Another option is to create a new "guest" permission group with no permissions. New permission groups can be created in Admin/Permission groups.

I have been trying to add new users to my database. Somehow they seem to **not receive the email** with their login credentials. Can I somehow resend the email? Or what would be the best way of adding them?

They can recover their password from the login page by clicking Forgot password? this will reset the password and send the new credentials to the user's email. If the new recovery email is also lost, the problem is most probably in your antispams or firewalls. The admin can change the user email address in Zebrabase and retry to recover the password this way.

Is it possible to **change the user name** of an existing account? It is currently XXX but she would like to have the user name YYY, for identification purposes once the labels are printed. Is it possible to do this for the admin? And for the other users, will we need to create new accounts to update the names in the same way?

Admins cannot change the username of an already existing user. Nevertheless, in exceptional situations (just one or two users), we can change it for you. Please, think in advance about what usernames you want to use. And of course, you can always do it the way you suggested - create a new user account and deactivate the old one.

One user requested a new password, which she got, but now she would like to **change the password** to something easier to remember.

Every user can change his/her own password: click your name in the top-right corner of the page to get to the user profile page, then click Change password. No permissions are needed to do it. Check if you are entering the generated password correctly (no leading/ending spaces etc.). Is the new password strong enough? The requirements are:

1. at least 8 characters long
2. not numeric-only
3. not too common (basic sequences or dictionary words are disqualified)

2.8.5 Data import

I can't find a way to **import my data** into your database. I have a lot of information in Excel files about the facility that I would like to import. Even though I have to make changes to my format it is much easier to import than to transfer one by one.

Data imports can be performed only with our assistance. Upon request, we will provide you with a template Excel file. After filling out the template, return it to us and we will proceed with the import. You will be able to import the stock/substock-related data. If your import contains position information, substock owners, workgroups, etc., it is necessary to have these structures already predefined in Zebrabase before the import.

2.8.6 Backups

Is it possible to receive an exported version of our database, so we can separately store a **backup of our data**? If there is a way for me to complete the export, could you point me to how to get started?

You can export all your substock data to an .xlsx file (there is an export/download icon in the upper right corner of the substock page). If you wish for the complete database backup (dump), we can also provide that but it is not accessible from the GUI. I.e. downloading data backups is possible upon request.

2.8.7 Printing

Finally, I have a Zebra GK420t printer available. In the database, however, I see the message **printer unreachable**.

Please look at https://docs.zebrabase.org/admin/09-printing_troubleshooting/ and try to locate the problem. It looks like the Printwes2 Client software is not running on the computer where you run Zebrabase. Alternatively, the printer is not switched on or is not connected to the same computer. If you are unable to solve the problem, send us snapshots of what the Printwes2 Client is showing (connection status, list of available printers).

Just recently the database started **printing 2 labels** rather than 1. It prints a duplicate label regardless of the device used to print from (iPad/computer)

You probably have Printwes2 Client running on two different computers and both computers have access to the same printer. You can prevent printing the label twice by switching off one of the Printwes2 Clients.

I'm trying to connect my new printer with Zebrabase and I went through your troubleshooting list: I can't print your example even from the Printwes2 Client GUI web page "Print". **I may not be using the right language** for the template. The user manual indicates that this printer supports the Datamax®3 language, Zebra®6 language, and Eltron®6 language. I also noticed the print button in Zebrabase (in Print QR) remains gray.

If you are using a Mac OS to run the Printwes2 Client and it does not work, you should try the procedure for installation of label printers for that system from the Zebra printers manufacturer: Installing a Zebra Printer on a Mac. Your printer should support ZPL language emulation, but the quality of emulation offered by non-Zebra manufacturers is uncertain and can differ from model to model, and the fastest method how to find out if it works for your labels is just to try. The first attempt should be made with software delivered by the manufacturer of your printer.

The **Printwes2 client will not open** on my Mac so I cannot print labels from Zebrabase. I have tried uninstalling and reinstalling Printwes2 three times, and the application still will not open/run. My Intermec PC43t printer works and will print labels from other applications. Do you know what I should do to correct this problem, or do you have the contact information for the people who developed Printwes?

The version of the Printwes2 client you are using has the problem that if the program is not shut down correctly, it leaves in the folder `~/.printwes2` the file `printwes_systray.pid`. The presence of this file indicates that the program is already running (even if it does not) so an attempt to start the Printwes2 client again will fail. Please download a new version of Printwes. Alternatively, the workaround is to delete this file manually, i.e., delete the file `~/.printwes2/printwes_systray.pid`. The `~` means the home folder of the user who installed the Printwes2 client, usually `/Users/USERNAME`, where the `USERNAME` is replaced with the actual user name.

I'm trying to work out how to apply the **split code**. In my label code below, the `substock.project.name` is often too long to fit on one line, but there is space within the label to split it across two lines. How would I apply the split code to do this please? Also, is it possible to change the font size of just one line or one item?

You can find the information about printing templates in the manual:

<https://docs.zebrabase.org/admin/08-printing/#printing-templates>

You can also get information about individual ZPL commands (e.g., `^CF`, `^FO`, ...) on the page [Labelary Online ZPL Viewer](#). Here, you can paste your Rendered source code (from the template update page) and when you click on a ZPL command in the text it will show you help for that command in the lower part of the web page. You can split any text using the `smart_split` command which is already used in the template for the substock name, so for your project name you can use, e.g.: `{% set prj_name_parts = substock.project.name | unicode | smart_split(30) %}` and then print `prj_name_parts[0]`, `prj_name_parts[1]`, ... (the `unicode` filter splits the eventual accents from the letters). You can change 30 to any other number of characters to be in the part of the project name. But I would recommend using the `substock.project.code` unless you have a special reason to have the full name of the project on the label. To change the font just for the following `^FD` command you can use `^A` command, e.g., to print a really small text: `^A0,10,10`.

I have got the printer set up and working, **using the printwes client on a networked computer**. However, it only works when I, individually log onto that specific computer to launch the client. If I log off, or log onto another computer on our network the printer is unreachable. Is it the case that the Printwes client software must be running in order to print from Zebrabase on any computer? This is problematic as we hotdesk, so I'm not always logged onto that computer. Any advice you can provide?

This is how the Printwes client works now. It must be running in a logged-in user account. Nevertheless, the screen for the user account can be locked if security is a problem. Or, a special user account can be created just for the Printwes client.

If you are using the remote mode configuration of Printwes, it makes sense to install the Printwes client only on computers that are connected to different printers. The printers should have different names assigned to recognize them. Another option is to switch Printwes to the local mode, then you can have Printwes clients installed on any number of computers (all with connection to the printer). But you would be able to print only from the computers with the Printwes client installed.

We are interested in purchasing a label printer. Based on the docs, it appears that the Zebra ZD420 is recommended. Have you found **labels that work well** for this printer and fish facility tanks? If so, could you please send me the info on the labels?

Any labels of the size 1 x 1.5 inches (25 mm x 38 mm) or larger can be used for printing the Zebrabase QR labels. If you want to use larger labels, you should keep in mind that the width of the label must be smaller or equal to the maximum supported label width of your label printer the roll core diameter and its outer diameter must correspond to the values stated in the specification of your label printer.

I want to install the **Printwes 2.0.8 client on a Windows server** without having a user to be logged on all the time. I have not found documentation about this setup, could you please help me out?

Printwes Client currently can not be run as a Windows service. Running it from a regular user account proved to be easy and for the Printwes local mode, it is also quite reasonable. See also [Printwes2 Client user account](#) FAQ above.

3. Admin

3.1 Initial facility configuration

This chapter explains the steps required for the initial setup of rooms and racks in the facility layout, creating new fishlines and substocks, adding users and defining user rights. It is advisable to go through the setup process step by step before you start using Zebrabase but you can also come back later to re-define any part of the configuration.

Please, be aware that the facility configuration is only accessible to administrators. For the initial setup, use the admin credentials you have received via e-mail.

Sections:

1. Fishlines
2. Founder substocks
3. Rooms
4. Racks
5. Species parameters
6. Diet, Light
7. Tags
8. Users
9. Workgroups
10. Projects
11. Printing

3.1.1 1. Fishlines

Fish / Fishlines / Add fishline

In Zebrabase, each fish substock is assigned to a specific fishline that is defined by genotypes and/or an alias. It is mandatory to fill at least one of these two fields. If an alias is available, it will be used preferentially for creating stock and substock names.

Entering wild-type fishlines

1) Enter fishline alias

This is the only compulsory field for WT lines. The recommended format of an alias for a wild-type line is “WT (AB)”, where the background is in brackets.

2) Fill in other optional parameters and fields

- Genotypes
- Origin - who created the fishline
- Source - who provided the fishline
- In facility since - date
- File Attachment - protocol, image, publication etc.
- Suffix - this is an additional note or descriptor for your fishline; for instance, if you have two fishlines with identical genotypes, you can distinguish them by attaching a suffix to one or both of them
- Responsible user - this person will appear as a responsible person for any fish belonging to the particular fishline
- ... and more, see image:

The image shows a screenshot of the Zebrafishbase web application's 'New fishline' form. The form is titled 'New fishline' and has a 'Save' button in the top right corner. The form is divided into several sections, each with a red box around its title and a red line pointing to the corresponding field(s) in the form.

- Species** (only for instances with the multi-species extension): Points to the 'Species' dropdown menu.
- User responsible for fishline maintenance**: Points to the 'Responsible user' dropdown menu.
- Linked genotypes**: Points to the 'Genotypes' dropdown menu.
- Fishline alias**: Points to the 'Alias' text input field.
- Fishline funding**: Points to the 'Funding' dropdown menu.
- Where was the fishline generated** (e.g. a specific lab): Points to the 'Origin' dropdown menu.
- Attach documents**: Points to the 'Attachments' dropdown menu.
- Where was the fishline sourced from**: Points to the 'Source' dropdown menu.
- Suffix** (will be displayed for the fishline and for the derived stocks and substocks): Points to the 'Suffix' text input field.
- Original strain** (e.g. AB line, Casper): Points to the 'Background' dropdown menu.
- When was the fishline introduced to facility**: Points to the 'In facility since' date input field.
- Workgroups with access to fishline**: Points to the 'Workgroups' selection area, which includes a list of items to select (CC Lab, Core Fishlines) and a 'Selected items' box.
- Associated projects for animal reporting**: Points to the 'Projects' dropdown menu.

At the bottom of the form, there are three buttons: 'Save', 'Save and add another', and 'Back to list'.

Entering genetically-modified fishlines

1) Select genotypes

Genetically modified or mutant lines are defined by their genotypes. Each genotype describes one mutation/transgene, and their combination describes the fishline. If the genotype is already stored in the database, start writing into the genotype field and select from the suggested options. If you need to assign more genotypes, repeat the procedure.

If the fishline you are creating carries genotypes that have not been previously defined in Zebrafishbase, you will need to define them first. To do so, click the plus sign next to the Genotype field and:

1. Select the modification category (transgenic or mutant) from the drop-down menu.
2. For transgenic lines, enter driver, triggered gene, and optionally the allele/mutation, e.g. *mitfa*:*BRAFV600E* or *gata1*:*DsRed*.
3. For mutant lines, define the affected gene and optionally, allele/mutation.
4. Save the genotype.
5. To add another genotype, click the plus button again and repeat the process.

You can also define genotypes independently in Fish / Genotypes / Add genotype.

The screenshot shows the 'New genotype' form in the Zebrafishbase application. The form is titled 'New genotype' and has a 'Save' button in the top right corner. The 'Modification category' dropdown menu is set to 'Transgenic'. Below this is a text input field containing 'Transgenic'. There are two input fields for 'Driver' and 'Triggered gene', both with plus signs and 'Start typing...' text. Below these is an 'Allele / Mutation' input field. At the bottom, there are three buttons: 'Save', 'Save and add another', and 'Back to list'.

Optionally, you can also create a new modification category in Admin / Codebooks / Mod Category. This can be an “Enhancer trap”, for example. You will need to specify if the modification category “behaves” like a transgenic (driver and triggered gene) or mutant (affected gene) genotype.

Editing genotypes

If you wish to edit some of the genotypes that have already been assigned to one or more fishlines, you can do so in Fish / Genotypes / Edit. If you edit a genotype, it will be updated for all the fishlines that have it assigned.

3.1.2 2. Founder substocks

Fish / Substocks / Founder substock

Substock is the basic functional unit in Zebrabase. It represents a group of fish that originate from a single crossing and are kept together in a tank. A substock, together with its sibling substocks from the same crossing, form a stock. All the fish from one stock share the same parents and date of birth (DOB), but they are rather a virtual unit in Zebrabase, i.e. not many actions are connected directly to stocks. The real functional subunits, i.e. the fish of a defined origin in a tank are called substocks.

Usually, substocks are created automatically by the Crossing action. Founder substock should be created only in case it is necessary, e.g. when receiving fish from another facility, or when setting up Zebrabase.

In the Founder substock form, it is mandatory to select the fishline and the date of birth. The rest of the fields are optional. Substocks ascribed to a specific workgroup are not visible for users outside the workgroup (they will be greyed out and locked in the Facility view).

The screenshot shows the 'New substock' form in Zebrabase. The form is organized into several sections:

- Stock Section:**
 - Fishline:** A dropdown menu with a search field.
 - Date of birth:** A text field with a 'YYYY-MM-DD' format.
 - Gen.:** A text field.
 - Suffix:** A text field.
- Substock Section:**
 - Number of substocks:** A text field with the value '1'.
 - Fish in each substock:** A text field.
 - Position:** A text field with a search icon and a 'Find a position by name' label.
 - Suffix:** A text field.
- Description:** A large text area for entering details.
- Bottom Section:**
 - Substock owners:** A dropdown menu.
 - Workgroups:** A dropdown menu.
 - Substock project:** A dropdown menu.
- Buttons:** 'Save', 'Save and add another', and 'Back to list' are located at the bottom of the form.

3.1.3 3. Rooms

Facility / Rooms / Add Room

In Zebrabase, you can reflect the layout of your facility by defining Rooms and Racks.

Create the first room of your facility and then press Save and add another to define more rooms or simply Save and continue to the next part of the facility configuration.

3.1.4 4. Racks

Facility / Design Rack

Once you have your rooms defined, you can fill them with racks. In **Step 1**, you can define the number of rows and columns independently for each rack you add to your facility. In **Step 2**, enter the name of your rack and select the room. To use the rack setup for multiple racks, click Add rack form and fill in the new form. Please, note that Zebrabase currently supports tanks occupying up to 4x standard tank width.

When you are finished designing your rack, hit Save and your newly designed racks will appear in the Facility view.

The screenshot displays the 'Design rack' interface in the Zebrafish Facility software. At the top, there are navigation tabs for 'Fish', 'Facility', 'Calendar', 'Projects', and 'Stats'. The current view is 'Design facility' under the 'Design rack' tab. A grid of 5 rows (A-E) and 10 columns of tanks is shown. Each tank has a red 'X' icon for deletion. A red box labeled 'Step 1: Design rack structure' highlights the top-left corner. A red arrow labeled 'Delete tank' points to a red 'X' icon in the first row, second column. Other red arrows point to 'Delete row', 'Add tank', and 'Move row up/down' controls. Below the grid, a 'Generated code' section shows a JSON array of tank configurations. 'Step 2: Specify racks' is highlighted, showing a form with two columns of input fields for 'Name', 'Room', and 'Description', and a 'Save' button.

3.1.5 5. Species parameters

Although the default settings connected to species are pre-set for zebrafish, they can be tweaked based on the facility's practice, or changed to suit a different species. These parameters include the age when the animals become juvenile/adult/retired, or the threshold densities of animals in tanks (used for color-coding tanks in the Facility view). They can be configured in Admin / Configuration under the following keys:

- `substock.age.juvenile` - age in days when fry becomes juvenile (**default value: 7**)
- `substock.age.adult` - age in days when juvenile becomes adult (**default value: 90**)
- `substock.age.retired` - age in days when adult becomes retired (**default value: 549**)
- `substock.count.verylow` - upper limit for the count of fish in a tank to be color-coded as "very low" (**default value: 5**)
- `substock.count.low` - upper limit for the count of fish in a tank to be color-coded as "low" (**default value: 15**)
- `substock.count.standard` - upper limit for the count of fish in a tank to be color-coded as "standard" (**default value: 25**)
- `substock.count.high` - upper limit for the count of fish in a tank to be color-coded as "high" - above this limit, the value is "very high" (**default value: 35**)

If you have the multi-species extension active, you can define these settings for multiple different species. Please see [Multi-species module](#) for more details.

3.1.6 6. Diet, Light

Admin / Codebooks / Diet Admin / Codebooks / Light

If you are planning to assign different diet and light conditions to different substocks, first you need to define them to be able to use them for your fish. Go to Admin / Codebooks, and select either Diet or Light. You can input new conditions by clicking the blue plus button in the upper right corner.

You can change the diet and light setting for each substock in the action Conditions (in the action panel in stock detail). It is also possible to change multiple substocks in bulk using batch actions. Default diet is also one of the parameters associated with your species.

3.1.7 7. Tags

Admin / Codebooks / Tag

Tags are used to associate additional, quickly accessible information with your substocks (such as low productivity, illness, single fish in a tank...). Tags are visible either in substock detail next to the name of the substock, or in facility view.

New tags can be designed in Admin / Codebooks / Tag. You can define the name and the color of the tag, and write a short description. The name should be short (ideally one word). Tags are most useful when only one or two of them are associated with a substock.

The screenshot shows the 'New tag' form in the ZebraBase application. The form has a dark header with navigation links: Dashboard, Configuration, Codebooks, Workgroups, Users, Permission groups, Notifications, Attachments, Newsletters, Tutorials, and Logs. A 'Save' button is in the top right. The form itself has a 'Name' field, a 'Description' text area, and two color selection palettes for 'Text color' and 'Background color'. A 'Preview' button shows the word 'default'. At the bottom are three buttons: 'Save', 'Save and add another', and 'Back to list'.

3.1.8 8. Users

Admin / Users / Add user

1. **Adding a new user account** The user name and a functional email must be defined, as well as the first name and the last name of the user. Type of the user (person/machine) must be selected. After saving the record, an email is sent to the new user's address with the automatically generated password. The users can change their password after logging into their account by clicking on their username and the Change password icon.
2. **Define additional details** Adding the Slack user name is optional, as well as the projects the user participates in. Permission groups determine the access rights the user is going to have. In the default, there are four permission groups: guest, user, facility, and admin. The rights of each permission group can be reviewed by the admin in Admin / Permission groups. If a workgroup has been selected for the user, he/she will only see the data belonging to his/her workgroup. Users without a selected workgroup will only see the substocks and fishlines that do not belong to any workgroup. In the table at the bottom of the form, the notification settings can be defined. Each type of notification can be sent by one or more of the three types of channels: Slack, e-mail, and system notifications (inbuilt ZebraBase messaging system).
3. **Select Save, or Save and add another**

User accounts can be edited in Admin / Users by clicking on the edit icon on the right side of the page or by clicking on the user and clicking on Update. An admin can ban a user by clicking on the Ban user icon in order to prevent the user from logging into the database (this will delete the user profile, however as most deleted items in Zebrabase, it can be re-enabled later).

zebrabase
Help | Test Admin | ZB3-ARM

Dashboard | Configuration | Codebooks | Workgroups | **Users** | Permission groups | Notifications | Attachments | Newsletters | Tutorials | Logs

New user Save

Personals

Username *	Type *
<input type="text"/>	<input type="text" value="person"/>
Email *	Slack
<input type="text"/>	<input type="text"/>
First name *	Last name *
<input type="text"/>	<input type="text"/>

Access

<p>Permission groups</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid #ccc; padding: 5px;"><i>Items to select</i></td> <td style="width: 20px; text-align: center;">⇌</td> <td style="border: 1px solid #ccc; padding: 5px;"><i>Selected items</i></td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 5px;"> <ul style="list-style-type: none"> admin ...cron facility guest _monitoring users </td> <td></td> <td style="border: 1px solid #ccc; padding: 5px;"></td> </tr> </table>	<i>Items to select</i>	⇌	<i>Selected items</i>	<ul style="list-style-type: none"> admin ...cron facility guest _monitoring users 			<p>Work groups</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid #ccc; padding: 5px;"><i>Items to select</i></td> <td style="width: 20px; text-align: center;">⇌</td> <td style="border: 1px solid #ccc; padding: 5px;"><i>Selected items</i></td> </tr> <tr> <td style="border: 1px solid #ccc; padding: 5px;"> <ul style="list-style-type: none"> CD Lab Core Fishlines Kozmik Machon Martasek Novotny Lab </td> <td></td> <td style="border: 1px solid #ccc; padding: 5px;"></td> </tr> </table>	<i>Items to select</i>	⇌	<i>Selected items</i>	<ul style="list-style-type: none"> CD Lab Core Fishlines Kozmik Machon Martasek Novotny Lab 		
<i>Items to select</i>	⇌	<i>Selected items</i>											
<ul style="list-style-type: none"> admin ...cron facility guest _monitoring users 													
<i>Items to select</i>	⇌	<i>Selected items</i>											
<ul style="list-style-type: none"> CD Lab Core Fishlines Kozmik Machon Martasek Novotny Lab 													

Projects

Notifications

Event	Slack	E-mail	System
input.condition Substock conditions changed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.count Input count executed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.productivity Input productivity executed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.productivity.juvenile Productivity changed to juvenile.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.productivity.productive Productivity changed to productive.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.productivity.retired Productivity changed to retired.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.substock_update Substock updated.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
input.transfer Input transfer executed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
message.new User got new system message.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
project.fish_limit_exceeded Fish limit exceeded on some project.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
project.severity_exceeded Maximal severity exceeded on some procedure.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.changed.assigned User was assigned to request by edit request.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.changed.data Request details changed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.changed.status Request status changed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.changed.unassigned User was unassigned from request by edit request.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.commented Request commented.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.created.assigned New request was created - notify assignees.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.declined Request declined.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
request.unassigned Somebody unassigned from request.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
substock.age.adult Substock became adult.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
substock.age.juvenile Substock became juvenile.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
substock.age.retired Substock became retired.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Save
Save and add another
Back to list

3.1.9 9. Workgroups

Admin / Workgroups / Add workgroup

Workgroups allow the separation of fishlines and users into multiple workgroups (these may be for instance different labs). The data belonging to a specific workgroup are then only visible to the corresponding members. **All fishlines assigned to a specific workgroup will be displayed only to its members.** Fishlines with no workgroup assigned will be visible to all.

The screenshot shows the 'New workgroup' form in the zebra base application. The form has a dark header with navigation links: Dashboard, Configuration, Codebooks, Workgroups (active), Users, Permission groups, Notifications, Attachments, Newsletters, Tutorials, and Logs. Below the header, there is a 'New workgroup' title and a 'Save' button. The form contains two input fields: 'Name *' and 'Description'. Below these is a 'Members' section with a dropdown menu labeled 'Start typing...'. At the bottom, there are three buttons: 'Save', 'Save and add another', and 'Back to list'.

Add fishlines to workgroups: Fish / Fishlines / Edit and select one or more workgroups from the options.

Assigning users to workgroups: Admin / Workgroups / Edit, or in the user profiles: Admin / Users / Edit.

The screenshot shows the 'Edit user #989 Test User (test-user)' form in the zebra base application. The form has a dark header with navigation links: Dashboard, Configuration, Codebooks, Workgroups, Users (active), Permission groups, Notifications, Attachments, Newsletters, Tutorials, and Logs. Below the header, there is an 'Edit user #989 Test User (test-user)' title and a 'Save' button. The form is divided into two sections: 'Personals' and 'Access'. The 'Personals' section contains fields for 'Username *' (test-user), 'Type *' (person), 'Email *' (support@zebrabase.org), 'Slack', 'First name *' (Test), and 'Last name *' (User). The 'Access' section contains 'Permission groups' (admin, _cron, guest, _monitoring) and 'Selected items' (facility, users). A red box highlights the 'Work groups' section, which contains 'Items to select' (Core Fishlines) and 'Selected items' (CD Lab, Novotny Lab). Below the 'Work groups' section is a 'Projects' dropdown menu labeled 'Start typing...'.

3.1.10 10. Projects

Admin / Codebooks / Project / Add Project

Projects can be used to specify different project licenses/IACUC codes/permission IDs or simply to separate fish into different projects. Enter the name and optionally also description and attachment for each of your projects and then Save or Save and add another.

Note that if you have the **Animal reporting module** extension active, there is an additional setup needed for your projects. Please navigate to the [Animal reporting module configuration](#) in that case.

The screenshot shows the 'New project' form in the Zebrabase application. The form is titled 'New project' and has a 'Save' button in the top right corner. The form fields are as follows:

- Code ***: A text input field.
- Name ***: A text input field.
- Description**: A text input field.
- Attachments +**: A dropdown menu with 'Start typing...' as the placeholder.
- Users**: A dropdown menu with 'Start typing...' as the placeholder.
- Start**: A date input field with the format 'yyyy-mm-dd'.
- Expiration**: A date input field with the format 'yyyy-mm-dd'.
- Used fish limit**: A text input field.
- Leader ***: A dropdown menu with 'Start typing...' as the placeholder and a red 'x' icon.
- Deputy ***: A dropdown menu with 'Start typing...' as the placeholder and a red 'x' icon.

At the bottom of the form, there are three buttons: 'Save', 'Save and add another', and 'Back to list'.

3.1.11 11. Printing

Zebrabase allows you to print substock labels with your local label printer. You can use the predefined Zebrabase templates, or create your own. A printing template is also displayed in the substock detail - you can change the template used in the substock detail by changing the configuration key `substock.detail.label_template`.

You can find more info in the section [Printing](#).

Substock #17292 - gata1:kalta4⁺,UAS:RFP⁺ 2024-04-04/1 (2)

Delete Update Update stock Back to list
Notify owners Print QR



Species	
SSID	17292
Status	unproductive
Tags	
Stock	gata1.kalta4,UAS:RFP 2024-04-04/1
Num	2 / 8
Fishline	gata1.kalta4,UAS:RFP
Suffix	
Background	
Position	F14:C08
F-gen	
Use	
Funding	
Count	15
Birth	2024-04-04
Death	not yet

Label 🔍

gata1:kalta4 (+),UAS:RFP (+)

DOB: 2024-04-04
ID: 7702-17292
POS: F14:C08

Observation log +

There is nothing here, yet.

Action log 🔍 👁 🗑

confirm burden (fishline) 2022-02-15

3.2 Animal reporting module

added in v. 3.2.0

Extensions needed: animal reporting

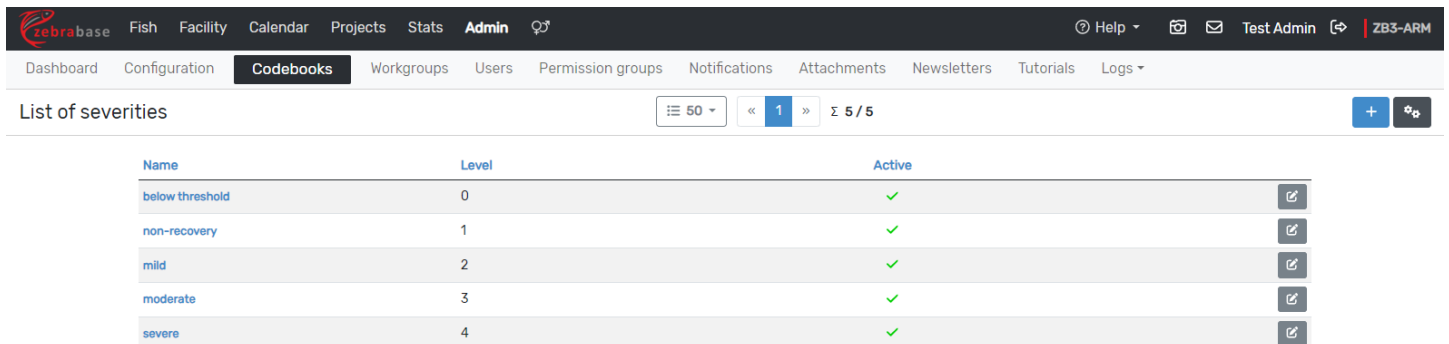
3.2.1 Initial configuration

For the reporting system to function properly, it is recommended to perform the following steps before starting to use it. The configuration needs to be done by an admin.

1. Procedure severities and severity threshold

Admin / Codebooks / Severity

It is possible to define the levels of procedure severities used for reporting (higher number = higher severity). It is recommended **not to change** the predefined options unless legislation in your country requires that.



The screenshot shows the ZebraBase Admin interface. The top navigation bar includes 'zebrabase', 'Fish', 'Facility', 'Calendar', 'Projects', 'Stats', 'Admin', and a search icon. Below this is a secondary navigation bar with 'Dashboard', 'Configuration', 'Codebooks', 'Workgroups', 'Users', 'Permission groups', 'Notifications', 'Attachments', 'Newsletters', 'Tutorials', and 'Logs'. The main content area is titled 'List of severities' and features a table with the following data:

Name	Level	Active	
below threshold	0	✓	
non-recovery	1	✓	
mild	2	✓	
moderate	3	✓	
severe	4	✓	

You can also specify the threshold severity value in Admin / Configuration, by changing the value of `project.min_severe_procedure_level`. By default, the threshold severity is **mild**.

- If the **severity threshold is reached**, animals will be subtracted from the Project capacity (see below for more details).
- If the **severity threshold is not reached**, the procedures will be reported in the report table but animals will not be subtracted from the capacity of the project.

2. Project

Projects / Dashboard

All current projects, running experiments, and the remaining project capacity are displayed in the **Project dashboard**.

The screenshot shows the 'Projects' section of the zebraBase interface. It displays four project cards, each with a header, a list of users, dates, experiment counts, and a color-coded bar at the bottom representing fish usage.

Project Code	Project Name	Users	Start Date	End Date	Closed Experiments	Running Experiments	Fish Usage (Current / Limit)
16/2020	NMDA receptor	alesbalik (alesbalik)	2020-06-22	2024-12-31	0	0	1 / 6000
262/2022	Vývojová biologie na modelu ryb	simonamrstakova (simonamrstakova)	2022-02-04	2024-12-31	0	0	144 / 1000
101/2021	Manipulace	nikol (jana)	2022-02-19	2026-12-31	0	3	3111 / 17000
1/2022	Burdens + New lines (not active)	jana (nikol)	not set	2026-12-31	0	0	5535 / 30000

Go to the new project form at Projects / Projects / Add project and add projects you wish to use for reporting.

It is mandatory to fill in a **project code** and a **name**, which will both be displayed in the project list. The code can be any combination of numbers and letters with a maximum length of 12 characters. Further mandatory fields include the **leader** and the **deputy**. Description, users, and expiration date are optional fields as well as the **used fish limit**. If set up, this limit will be displayed in the project dashboard in the color-coded horizontal bar below the project, where also the current usage of fish can be seen.

The screenshot shows the 'New project' form in the zebraBase interface. It contains several input fields and dropdown menus for creating a new project.

Fields:

- Code ***: Text input field.
- Name ***: Text input field.
- Description**: Text input field.
- Attachments +**: Text input field with a dropdown arrow.
- Users**: Text input field with a dropdown arrow.
- Start**: Text input field with format 'yyyy-mm-dd'.
- Expiration**: Text input field with format 'yyyy-mm-dd'.
- Used fish limit**: Text input field.
- Leader ***: Text input field with a dropdown arrow and a red 'x' icon.
- Deputy ***: Text input field with a dropdown arrow and a red 'x' icon.
- Allowed procedures**: Section with a '+ create new procedure' button.
- Max. prospective severity ***: Dropdown menu with 'below threshold (0)' selected.
- Procedure ***: Text input field with a dropdown arrow and a red 'x' icon.

Buttons:

- Save**: Blue button.
- Save and add another**: Blue button.
- Back to list**: White button.

For each project, it is necessary to add the **allowed experimental procedures** that will take place within the project. Each procedure must have an associated maximum prospective severity selected from the drop-down menu. The number of procedures allowed in each project is unlimited - procedures can be added to a project by selecting **add procedure**. Also, new procedures can be created either straight in the form at **Create new procedure**, or in Admin / Codebooks / Procedure.

The **procedure - severity** combination is saved specifically for the project you are editing. It **does not affect** the severity of the same procedure in other projects.

3. Age threshold

The minimum age when procedures need to be reported in a project can be set in Admin / Configuration in `project.fish_reported_age`. Typically, this is **5 dpf in zebrafish**.

In case an experiment is launched with **fish below the limit**, their number will be subtracted from the corresponding project capacity, when the experiment is in progress. However, fish that have died **before reaching the age threshold** will no longer be counted as used fish after the experiment is finished. They will only be included and marked in the reporting table.

4. Observation period

To ensure that animals that have died as a consequence of a procedure are counted in an experiment, administrators can specify an **observation period**. The observation period is defined as the number of days that have to pass after a procedure, during which any deceased fish are still considered to die due to the experimental treatment.

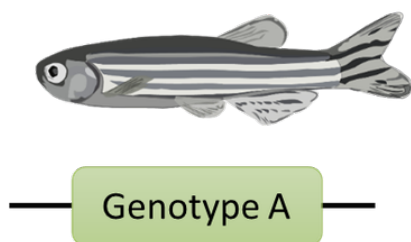
By default, the observation period is set to 0 days. The administrator can change the length of the observation period in Admin / Configuration by changing the value of `project.experiment_observation_period`.

IMPORTANT: It is not possible to end an experiment before the termination of the observation period of all involved substocks.

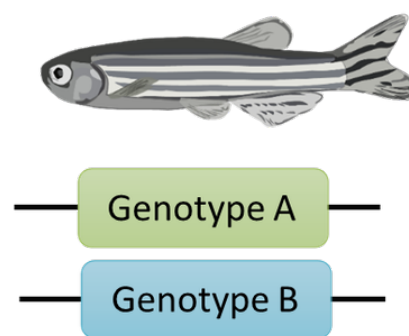
5. Burdens

In Zebrabase, burdens are **genotype-zygosity combinations** that are damaging to the animal, resulting in e.g. immunodeficiency, low survival, or a higher chance of cancer. It is possible to specify which genotype-zygosity combinations represent a burden for each fishline and should be tracked (see an example below).

Example of burdened genotype-zygosity combinations for fishlines with one and two genotypes - only some combinations produce the burden phenotype (marked by an exclamation mark):



example	
A ^{WT/WT}	OK
A ^{WT/MUT}	OK
A ^{MUT/MUT}	!



example	B ^{WT/WT}	B ^{WT/MUT}	B ^{MUT/MUT}
A ^{WT/WT}	OK	!	!
A ^{WT/MUT}	OK	!	!
A ^{MUT/MUT}	!	!	!

To **enter or edit burdens**, go to a specific fishline record and click **Edit burdens** in the row **Burdens**. Then select all the genotype-zygosity combinations that represent a burden for the fishline. The table also contains these two options:

- **?** - A burden is associated with undefined zygosity (i.e. before genotyping).
- **Any** - A burden is associated with all zygositys (but the substock has to be genotyped and positive in at least one allele).

The screenshot shows the 'Edit burdens' dialog box in the Zebrafish database. The dialog is titled 'Edit burdens' and is overlaid on the main interface for fishline #327. The main interface shows the fishline name '-3.5ubi:GFP,b-globin:GFP,flk1:Cre' and various buttons like 'Delete', 'Update', 'Clone', 'Back to list', 'Founder substock', and 'Change genotypes'. The dialog box contains the following fields and options:

- Fishline:** -3.5ubi:GFP,b-globin:GFP,flk1:Cre
- Performed:** 2024-04-19
- Note:** (empty field)
- Select burdened genotype-zygosity combinations:**

Genotype	-	+	+/+	-/-	WT	+/-	?	Any
b-globin:GFP (Tg)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-3.5ubi:GFP (Tg)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
flk1:Cre (Tg)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
- Buttons:** 'Check form' and 'Save action'.

Once you have the burdens defined, all substocks bearing the defined combinations will be marked as burdened by a **red exclamation mark** at the end of their name, so that you can distinguish them easily.

The screenshot shows the 'Substocks' view in the Zebrafish database. The interface displays a list of substocks with the following columns: Name, BG, F-gen, Num, Birth, Status, Count, Position, Owners, Responsible, Workgroups, Project, Funding, Use, Statusbar, and ID. The substocks are listed as follows:

Name	BG	F-gen	Num	Birth	Status	Count	Position	Owners	Responsible	Workgroups	Project	Funding	Use	Statusbar	ID
p53 ^{M214K} -/- ₁			7	2023-10-19	unproductive	20	C03:E10	jovana	CD Lab						7468-16583
p53 ^{M214K} -/- ₁			6	2023-10-19	unproductive	21	C03:D10	jovana	CD Lab						7468-16582
p53 ^{M214K} -/- ₁			5	2023-10-19	unproductive	20	C03:E09	jovana	CD Lab						7468-16581
p53 ^{M214K} -/- ₁			4	2023-10-19	unproductive	20	C03:E08	jovana	CD Lab						7468-16580
p53 ^{M214K} -/- ₁			3	2023-10-19	unproductive	20	C03:E07	jovana	CD Lab						7468-16579
p53 ^{M214K} -/- ₁			2	2023-10-19	unproductive	20	B01:B03	jovana	CD Lab						7468-16578
p53 ^{M214K} -/- ₁			4	2023-03-01	unproductive	10	C03:D03	jovana	CD Lab		1/2022				7098-15292

Burdened fishlines are confirmed in two steps: **1. Initial check** is required for any newly added fishlines. It is accomplished by the steps described above. **2. The second check** acts as a follow-up confirmation of the observations that should be performed after 3 generations of animals (the default interval in Zebrafish is 270 days, but can be

changed). It should serve to capture any phenotypes that were not obvious during the initial stages of the breeding. You can keep track of the currently required checks on the admin dashboard.

The screenshot shows the ZebraBase Admin Dashboard. At the top, there is a navigation bar with the following items: Fish, Facility, Calendar, Projects, Stats, Admin, and a search icon. On the right side of the navigation bar, there are links for Help, Test Admin, and ZB3-ARM. Below the navigation bar, there is a sub-navigation bar with the following items: Dashboard, Configuration, Codebooks, Workgroups, Users, Permission groups, Notifications, Attachments, Newsletters, Tutorials, and Logs. The main content area is divided into two sections. The first section, titled 'Extensions:', contains eight icons representing different features: Customization, Calendar, Batch, Workgroups, Multi-species, Extended stats, Water monitoring, and Animal reporting. The second section contains six cards displaying various metrics: 'Fishline burden initial check' with a value of 3, 'Fishline burden second check' with a value of 0, 'Substocks of deleted fishlines' with a value of 0, 'Projects exceeding expiration' with a value of 0, 'Projects exceeding capacity' with a value of 0, and 'Projects exceeding 75% of the capacity' with a value of 0. Each card has a 'link' button at the bottom right.

In Admin / Configuration under the key `project.default_burden_project`, you can specify the default project for reporting burdened animals. In case a burdened substock does not have any project specified, it will be reported under the default project. If you need to report some animals under another project, these animals should be constantly associated with that particular project (not only during experiments).

You can also specify the key `project.default_burden_severity` to specify the default severity of burdens (the severity can be changed in the report form), and `project.fishline_burden_check_interval`, which is the default interval between the initial and consequent burden checks.

6. Configuration of count types

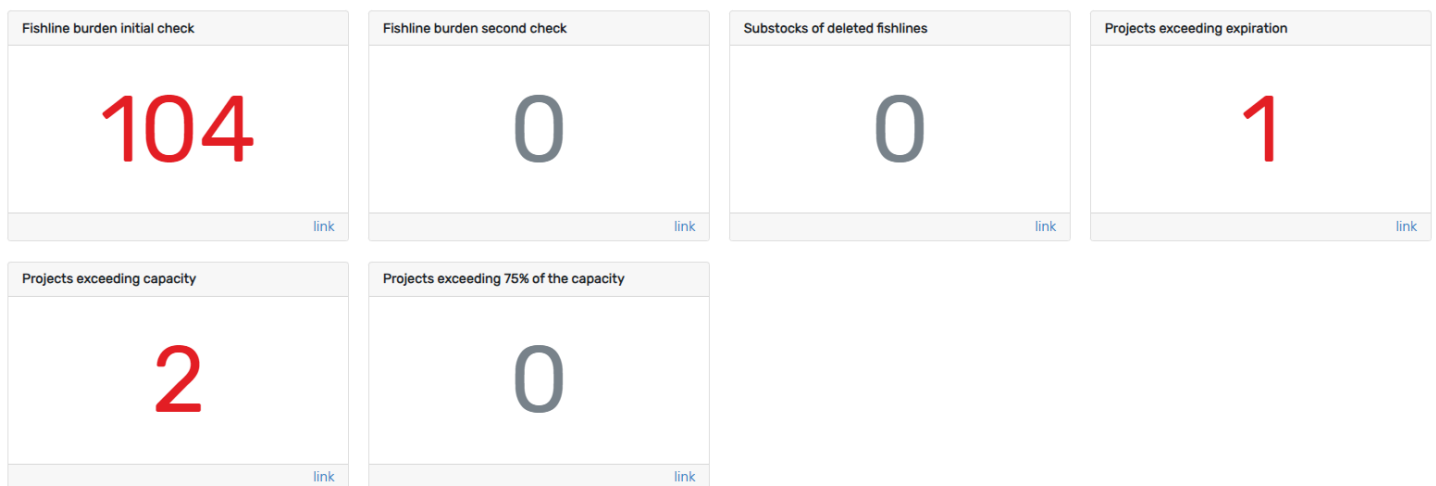
Changing the count type configuration is currently not permitted - this section is intended only for a better understanding of the mechanisms behind it.

Two attributes determine the input counts:

- `count_to_experiment` determines whether an input will be included in the project table (if a substock is not associated with a project, this attribute has no effect)
- `count_to_burden` determines whether fish that died during a given procedure should be added to `used_burdened_fish` of a project. If a substock is associated with an experiment and `count_to_experiment` is also set, then `count_to_experiment` will have priority over `count_to_burden`.

3.2.2 Admin dashboard

The admin dashboard allows for an easy visual check that your data is up-to-date. The five boards show the number of entries that should be reviewed by an administrator and potentially corrected. If there is no data to check, the displayed number will be a gray zero. Any red numbers should be reviewed. Clicking on the blue link at the bottom right corner of the boards will display the list of items to review.



1. Fishline burden initial check

In the initial check, any potential burdens should be entered for each new fishline or it should be confirmed that the fishline is not burdened. In the fishline detail, click the **Edit burdens** icon in the **Burdens** row, and this will allow you to select burdens for specific genotype-zygosity combinations. Clicking the green tick confirms the current status.

2. Fishline burden second check

After 270 days (i.e. 3 generations) of the existence of a new fishline, the fishline displays here, so that the original burdened status may be re-evaluated and corrected (removed or added) if necessary.

The delay between the first and second burden check may be changed in Configuration, key `project.fishline_burden_check_interval`.

3. Substocks of deleted fishlines

Only substocks of existing fishlines should be present in the database. This board will alert the admin in case a fishline has been deleted without the removal of associated substocks.

4. Projects exceeding expiration

Any projects running past the pre-set expiration date will be displayed here. They can be either deleted or their expiration date can be reset.

5. Project exceeding capacity

Any projects exceeding their capacity (i.e. the used fish limit) will be displayed here.

6. Project exceeding 75% of the capacity

Any projects exceeding 75% of their capacity (i.e. the used fish limit) will be displayed here.

3.2.3 Reporting the use of animals (for all users)

This section is intended for all users using the animal reporting module. It describes three scenarios:

1. Reporting of animals used in experimental procedures
2. Reporting of burdened animals
3. Reporting of special actions (e.g. genotyping)

1. Reporting of animals used in experimental procedures

STEP 1 - START A NEW EXPERIMENT

A new experiment can be started either from the **Projects** module or from a substock of your choice.

- **From a substock:** The detail of each Substock contains the **Experiment** action button, clicking on this button opens a drop-down menu where you can select between adding a substock to an existing experiment and designing a new experiment.
- **From Projects:** Go to Projects / Experiments and click on **Add experiment**.

In the new experiment form, the **Name of the experiment** and the **Start date** should be entered, as well as the **Project** this experiment will be a part of. The prefix of the experiment is generated automatically using the **Project code** and the **Experiment number**. Substocks are added to the experiment using the **Add substocks** field.

The screenshot shows the 'New experiment' form in the ZebraBase system. The form is titled 'New experiment' and has a 'Save' button in the top right corner. The form includes the following fields and sections:

- Name ***: A text input field with the value 'no prefix'.
- Project ***: A dropdown menu with the value '-- none --'.
- Begins ***: A date input field with the value '2024-04-19'.
- Ends**: A date input field with the placeholder 'yyyy-mm-dd'.
- Add substocks**: A section with a plus icon and the text 'Add substocks'.
- Substocks**: A search bar with the placeholder 'Start typing...' and a dropdown arrow.
- Performed date for substock actions**: A date input field with the placeholder 'yyyy-mm-dd'.
- Buttons**: Three buttons at the bottom: 'Save', 'Save and add another', and 'Back to list'.

All currently running experiments and associated substocks can be viewed in **Experiments**.

Name	Start Date	End Date	Project	Substocks currently in experiment	Substocks used in experiment	Running
57/2018 [Microinjection]	2020-02-29			☰	☰	✓
57/2018 [Chemical treatment]	2020-02-29			☰	☰	✓
57/2018 [Transplantation]	2020-02-29			☰	☰	✓
57/2018 [Nucleic acid isolation for PCR]	2020-02-29			☰	☰	✓
57/2018 [WKM isolation]	2020-02-29			☰	☰	✓
96/2018 [Electroporation]	2020-02-29			☰	☰	✓
101/2021_E01 - cryo	2022-01-01		101/2021 - Manipulace	☰	☰	✓
101/2021_E02 - sort post 5 dpf	2022-06-28		101/2021 - Manipulace	☰	☰	✓
101/2021_E03 - IVF	2022-10-10		101/2021 - Manipulace	☰	☰	✓
65/2022_E01 - Kidney marrow isolation	2022-12-01		65/2022 - Regulation of normal and malignant hematopoiesis	☰	☰	✓

STEP 2 - ADD SUBSTOCKS TO THE EXPERIMENT

This step needs to be completed only in case a new experiment has been started without adding substocks or if more substocks need to be added to an already running experiment - i.e. the substock you want to use in the experiment is not in the experiment yet. The associated substocks can be found in the **Experiments** overview, in the **Substocks currently in experiment** list.

IMPORTANT: To add a substock to an experiment, animals in the substock **have to be counted**. The Experiment button is hidden for substocks without a start count.

Substocks can be added to experiments by clicking the **Add to experiment** action button in the substock detail and selecting **Add to existing experiment**.

Once the substock enters an experiment, **all the following subtractive count actions** will be reported and assigned to the experiment and project. It is **not possible** to euthanize an animal during the course of the experiment without an association of the count action to the experiment/project. If you wish to omit such an action from the experiment/project statistics, it is necessary to remove the substock from the experiment, euthanize the animals, and then add the substock to the experiment again.

IMPORTANT: Single project per substock

A substock can be associated with only one project at a time. To report in multiple projects, do it sequentially by reassigning the substock from one project to another.

When entering an experiment, all fish from a substock are added to the **Used fish** of the corresponding project. At the end of the experiment, the number is **corrected** based on how many animals were actually used. The animals are counted only if the severity of at least one of the procedures **exceeds the defined threshold**.

STEP 3 - REPORT AN EXPERIMENTAL PROCEDURE

To **report an experimental procedure**, use the action **Count (Procedure)** for the substock that is in the ongoing experiment. The new procedure can be added in the field **Record new procedure**; it is important to choose the correct **Count type** for each procedure based on the table below:

Count type	Description	Is it subtractive?
Non-lethal procedure	Animals are undergoing an experimental procedure (non-lethal)	NO
Euthanized	Animals were euthanized as a part of the procedure or after the procedure	YES
Died	Animals died during the procedure or after the procedure	YES

* Subtractive procedures subtract the number of fish on which they were performed from the total fish count of the substock.

For each procedure, it is necessary to enter:

- **Count** - how many animals have entered the procedure
 - For the option **non-lethal procedure**, it is recommended to enter **All fish** for the clarity of the record.
 - In case only a **part of the substock** is used for an experiment, it is recommended to **split it first**.
 - In case some animals will be marked as **not used** at the end of the experiment, **the substock will be split automatically** into a substock that underwent the procedures (and is reused) and another one that has not been used at all.
- The **actual severity** of the procedure
 - The **Actual severity** can be different than the **prospective severity**
 - Any severity with **lower or the same level** as the prospective severity is displayed in **blue**
 - Any severity with a **higher level** than the prospective severity is displayed in **red**
 - The **default value** corresponds to the **prospective severity** of the procedure

You can use the field **Note** to add a description of the action. The note will appear in the report table.

The screenshot shows the 'Count' form in the zebrafishbase application. The form is titled 'Count' and is used for reporting animal procedures. It includes the following sections:

- Substocks:** A list of substocks with a filter set to 'All [1] Problematic [0]'. One substock is highlighted: '7468-16580 C03:E08 20' with a count of 183.
- Performed:** A date field set to '2024-04-19' and a 'Note' field.
- Warning:** A message states 'This item already has some more inputs for selected date - please choose the exact timing' with a dropdown menu set to '-- last --'.
- Count type:** A dropdown menu set to 'non-lethal procedure' and a 'Count' field.
- all fish:** A toggle switch that is currently turned on.
- Procedure:** A dropdown menu set to 'fin clip'.
- Severity:** A set of radio buttons with options: 'below threshold', 'non-recovery', 'mild' (selected), 'moderate', and 'severe'.
- Buttons:** 'Check form', 'Save action', and 'Save and leave the experiment'.

STEP 4 - REPORT ANIMALS AFTER A PROCEDURE

If animals are found dead or euthanized **after a procedure**, it is possible to associate them with the last procedure in the experiment. In that case, use the **Count** action and navigate to the **“Add to existing procedure”** part of the form.

The screenshot displays the 'Count' dialog box in the zebraBase application. The dialog is titled 'Count' and has a close button (X) in the top right corner. It is divided into several sections:

- Substocks:** A filter section with 'All [1]' and 'Problematic [0]'. Below it is a search bar 'Add a substock...' and a list of substocks: '7468-16580 C03:E08 20' and 'p53^{M214K}-/-; 2023-10-19/1 (4) 183'.
- Performed:** A date field containing '2024-04-19' and a 'Note' field.
- Warning:** A message: 'This item already has some more inputs for selected date - please choose the exact timing' with a dropdown menu set to '-- last --'.
- Count type:** A dropdown menu set to 'died after 90dpf' and a 'Count' field set to '2'. There is also a toggle for 'all fish'.
- Procedure:** A section titled 'In experiment of project 101/2021' with options to 'Record new procedure' or 'Add to existing procedure'. Under 'Add to existing procedure', there is a dropdown for 'Existing procedure action' set to '-- none --'.
- Severity:** A row of five buttons: 'below threshold', 'non-recovery', 'mild', 'moderate', and 'severe'.
- Buttons:** At the bottom right, there are three buttons: 'Check form', 'Save action', and 'Save and leave the experiment'.

The background shows the 'Substocks' tab of a substock record for 'Substock #16580'. It includes a sidebar with various filters (Species, SSID, Status, Tags, Stock, Num, Fishline, Suffix, Background, Position, F-gen, Use, Funding, Count, Birth, Death, Most se) and a list of substocks with their respective dates and counts.

SEVERITY ASSESSMENT

In order to determine the **actual severity** of procedures, and the severity for animals that died or were euthanized after a procedure, the local authorized officer should be consulted or the official guidelines in a given country should be reviewed.

STEP 5 - MULTI-PROCEDURE EXPERIMENTS

In case there are subsequent procedures in the experiment, steps 3 and 4 (reporting of procedures) can be repeated several times before leaving the experiment.

STEP 6 - ENDING AN EXPERIMENT

To remove a substock from an experiment, click the **Leave experiment** button in the substock record. When the only or the last substock is removed from the experiment, the user is asked whether he/she wants to terminate the whole experiment. An experiment cannot be terminated until all substocks leave the experiment first.

When leaving an experiment, it is possible to mark all or a portion of the surviving fish as **unused**. These animals will be **subtracted** from the **Used fish** of the project. A **new unused substock** will be split off from the original substock if only a portion of the fish is unused.

The overall severity of procedures can be reviewed and changed at this point. The default value is the highest severity the animals have experienced in the course of the experiment. However, it is suggested to use the maximum severity experienced by the animals throughout their life.

The screenshot displays the 'Remove from experiment' dialog box in the zebabase Facility view. The dialog box includes a 'Performed' field with the date '2024-04-19' and a 'Note' field. Below these fields is a table with the following columns: Substock, Experiment, Fish count, Overall severity, and Unused fish. The table contains one row with a red 'x' in the Substock column, indicating a selected row. The row details are: Substock: 6801-14489 p53^{M214K}-/-; Experiment: 101/2021_E01 - cryo; Fish count: 17; Overall severity: mild; Unused fish: 0. The background shows a grid of racks (A01-A10, B01-B02, C01-C02, D01-D10, E01-E10) with various fish and their associated dates and genotypes.

SEVERITY ASSESSMENT

In order to assess the cumulative severity after multiple procedures, the local authorities should be consulted or the official guidelines of the country should be used.

If an animal **dies after the experiment** has ended, the animal will not be counted in the experiment by default. However, it is possible to set an **observation period**, which will allow the users to report also animals that died soon after an experiment.

Animals that were used in a procedure with a **severity higher than the threshold** and left the experiment, but are still kept in the facility, are marked as **Reused** (recycle icon in the status bar and the name of the substock).

In case there are any **Reused fish** at the end of the experiment, a special type of procedure is produced in the **Report table**. This procedure includes the action **Count** with **procedure severity** equal to the maximum severity reached by any procedure the animals experienced (i.e. the overall severity).

2. Reporting animals with a genetic burden

It is **not necessary** to start an experiment to report burdened animals under a project. Instead, they can be reported directly under a chosen project and they will appear in an independent section, similar to genotyping reports. Animals will be counted automatically in case a **subtractive count** (died/euthanized) is saved for the burdened substock (see below) and the substock has a **Project** assigned.

When reporting **dead or euthanized burdened animals**, the **severity** of the procedure must be selected. The default severity for burdened animals is **mild** but it is possible to reconfigure this for the whole database of the facility (in Admin / Configuration at `project.default_burden_severity`).

The screenshot displays the Zebrafish database interface. A modal window titled "Count - p53^{M214K} -/-, 2023-02-22/2 (2)" is open over a rack view. The modal contains the following fields and options:

- Substocks**: Filter: All [1] Problematic [0]. A substock entry is visible: `7080-15222` B01:C05 20 [recycle icon] [stop icon] p53^{M214K} -/-, 2023-02-22/2 (2) [recycle icon] 422.
- Performed**: date of execution: 2024-04-19. Note: This item already has some more inputs for selected date - please choose the exact timing. -- last --
- Count type**: died after 90dpf. **Count**: [input field]. all fish.
- Project**: 1/2022 - Burdens + New lines (not active).
- Severity**: below threshold, non-recovery, **mild** (selected), moderate, severe.
- Buttons: Check form, Save action.

Please, consult your **local authorities** or **responsible person** to determine **how to assess the severity** for burdened animals properly.

A new project dedicated to reporting burdened fish can be created and used in these cases.

3. Reporting genotyping

Similarly to burdened animals, genotyping can be reported under a project without creating an experiment. In the form for the **Genotyping** action, there is a section where you can directly select the project which should be used for the reporting, and the method.

The screenshot displays the Zebrafish database interface. The main view is a 'Facility view' grid showing various substocks (A01, B01, C01, D01, E01) and their associated dates. A modal window titled 'Genotyping - p53^{M214K} -/- - 2023-02-22/2 (2)' is open over the grid. The modal contains the following fields and options:

- Substocks:** Filter: All [1] Problematic [0]. A dropdown menu shows '7080-15222 B01:C05 20' and 'p53^{M214K} -/- - 2023-02-22/2 (2)'. Below is an 'Add a substock...' input field.
- Performed:** A date field set to '2024-04-19' and a 'Note' field.
- Genotypes:** A dropdown menu showing 'p53^{M214K}' and 'homozygote - mutant (-/-)'.
- Additional data:** An 'ARM Project' dropdown set to '101/2021 - Manipulace' and a 'Genotyping method (optional)' dropdown set to 'fin clip (PCR) (mild)'.
- Buttons: 'Check form' and 'Save action'.

If the substock does not have a fish count set before genotyping, no count will show up in the genotyping report! Please pay attention to the fish count being set if you need this data in the report.

4. Batch actions for substocks in the experiment and burdened substocks

There are certain limitations of batch actions for substocks with a burden or in an experiment. Using **Count** and **Terminate** should be avoided in the batch mode if the substock is burdened or in an experiment. These actions **will not** be accepted because they require an input of specific details (e.g. procedure severity).

However, it is possible to use the advantages of the batch mode for other actions, e.g. removing substocks from an experiment, changing the productivity, conditions, etc.

3.2.4 Reports

Each project has a variable `used_fish` which sums up all fish used in this project (any procedure severity). The value is displayed on the fish usage indicator bar, and also in the field Fish usage in the project detail. In the project detail, you can also find itemized the 3 constituents of fish usage: experiments, burdened animals, and genotyping.

The screenshot shows the ZebraBase interface for Project #4 - 101/2021 - Manipulace. The project description is 'Kryokonzervace, in vitro fertilizace a genotypování Danio rerio'. The 'Fish usage' indicator bar shows 5211 / 17000. The 'Used fish' table is highlighted with a red box, showing a total of 80 + 20 + 5111. The 'Experiments' table lists three experiments with their actual fish counts.

Name	Actual fish count
101/2021_E01 - cryo	0
101/2021_E02 - sort post 5 dpf	None
101/2021_E03 - IVF	None

For each project, an overview of all experiments and used burdened animals can be seen at the bottom of the page in project detail. It consists of three parts: **Experiment report, Burdened fish report, and Genotyping report**. The reports consist of list of all experiments and procedures, and a summary (for each year and the whole project), which contains the used fish counts according to:

- **EU** - the current EU legislation (each animal is only used once, the highest procedure severity is counted)
- **all** - overall summary (each procedure is counted, one animal can be counted multiple times)

The number of `used_fish` does not have to agree with the sums in the reporting table, because during some activities the number of used fish is updated manually by the user and some fish can also be used repeatedly; ZebraBase is not able to account for this, so please use the fish usage indicator only as an orientational value and when reporting to authorities, always use the data from the report table.

Experiment report

The table of experiments for each project is generated from Count inputs and offers a detailed overview of the use of fish in the project.

Project #4 - 101/2021 - Manipulace

Experiment report

2022

Year summary (EU):	below threshold: 0	non-recovery: 24	mild: 39	moderate: 0	severe: 0
Year summary (all):	below threshold: 0	non-recovery: 24	mild: 39	moderate: 0	severe: 0

2023

Year summary (EU):	below threshold: 0	non-recovery: 17	mild: 0	moderate: 0	severe: 0
Year summary (all):	below threshold: 0	non-recovery: 17	mild: 0	moderate: 0	severe: 0

101/2021_E01 - cryo [→ 2022](#) [2022-01-01 - ?]

Exp. summary (EU):	below threshold: 0	non-recovery: 17	mild: 0	moderate: 0	severe: 0
Exp. summary (all):	below threshold: 0	non-recovery: 17	mild: 0	moderate: 0	severe: 0

5625-12013 Xla.Tubb:DsRed^{+/+}.grin2aa^{540072 +/-} 2021-09-02/1 (1)

→ [testis dissection](#) [2023-10-19]

2023-10-19 [777]	-6 / 17	euthanized (healthy)	non-recovery	facility	---	
------------------	---------	----------------------	--------------	----------	-----	--

leaving experiment [2023-10-19] unused fish: 11

5797-12406 3' globin enhancer_pME-b-globin:EGFP⁺.gata1:DsRed⁺ 2021-11-09/2 (2)

→ [testis dissection](#) [2023-10-19]

2023-10-19 [709]	-2 / 12	euthanized (healthy)	non-recovery	facility	---	
------------------	---------	----------------------	--------------	----------	-----	--

leaving experiment [2023-10-19] unused fish: 10

5797-12407 3' globin enhancer_pME-b-globin:EGFP⁺.gata1:DsRed⁺ 2021-11-09/2 (3)

→ [testis dissection](#) [2023-10-19]

2023-10-19 [709]	-2 / 2	euthanized (healthy)	non-recovery	facility	---	
------------------	--------	----------------------	--------------	----------	-----	--

The numbers of animals undergoing different levels of procedure severity are summarized for each experiment and also for the whole project. The number of unused animals is also specified for each experiment - this only applies if the animals are set as unused when the substock is exiting the experiment and there are still some animals alive.

IMPORTANT! If the report table is too long, only a part of it will be displayed on the page. The full report table can be viewed by exporting the table and opening it externally.

Burdened fish report

Breeding fish with burdened phenotypes should be always reported (according to the current EU legislation).

This table shows all burdened animals that died under the project and the associated severity. If the burdened lines are not already entered in another project, we recommend setting up a dedicated project for reporting burdened fishlines. The default project for reporting burdens can be set in Admin / Configuration under the key `project.default_burden_project`.

Year summary:							below threshold: 0	non-recovery: 19	mild: 0	moderate: 1	severe: 0
Date		Substock	Count	Type	Severity	User	Note				
2022-10-10		5628-12054 mpx:GFP+.csf1ra ^{V614M} -/-; 2021-09-03/1 (5)	-17 / 17	euthanized (sick)	non-recovery	facility	---				
2022-10-10		5501-11816 CASPER-rag2 E450fs; 2021-07-08/3 -/- (2)	-2 / 2	euthanized (sick)	non-recovery	facility	---				
2022-10-11		5573-11949 rag2 ^{E450fs} -/-; 2021-08-10/1 (7)	-1 / 11	died after 90dpf	moderate	facility	---				

Genotyping report

Similarly to burdened fish, all genotypings are automatically reported in a separate table. For fish that are not a part of any specific project, we recommend setting up a genotyping project, under which the genotyped fish will be reported. You can change the default project for reporting genotypings in Admin / Configuration under `project.default_genotyping_project`.

Year summary:							below threshold: 297	non-recovery: 0	mild: 23	moderate: 0	severe: 0	not set: 966
Date		Substock	Count	Method	Severity	User	Note					
2023-03-21		7119-15414 mpx:GFP +/- 2023-03-15/1 (6)	25	not set	not set	martin	---					
2023-03-21		7123-15388 flk1:NTR-mCherry +/- 2023-03-16/2 (1)	not set	not set	not set	facility	---					
2023-03-22		7119-15410 mpx:GFP +/- 2023-03-15/1 (2)	25	sorting before 5 dpf (microscope)	below threshold	martin	---					
2023-03-22		7119-15411 mpx:GFP +/- 2023-03-15/1 (3)	25	sorting before 5 dpf (microscope)	below threshold	martin	---					
2023-03-22		7119-15412 mpx:GFP +/- 2023-03-15/1 (4)	25	sorting before 5 dpf (microscope)	below threshold	martin	---					
2023-03-22		7119-15413 mpx:GFP +/- 2023-03-15/1 (5)	25	sorting before 5 dpf (microscope)	below threshold	martin	---					
2023-03-22		7119-15384 mpx:GFP +/- 2023-03-15/1 (1)	25	sorting before 5 dpf (microscope)	below threshold	martin	---					

3.3 Custom fields

Added in v. 3.4.1

Extensions needed: customization, (water monitoring)

Custom fields are meant to be used for any additional information about **users** (such as the license number etc.) and for water quality related data that should be part of the **waterlog**. The fields are named `custom_field_[1-N]` but can be renamed by the facility admin according to the facility's needs.

Once you make the changes, the new field will appear as an interactive part of the user profile, respectively waterlog.

User custom field:

The screenshot shows the 'Edit user #989 Test User (test-user)' page in the ZebraBase Admin interface. The page is titled 'Personals' and contains several input fields for user information. A custom field named 'custom field 1' is highlighted with a red border. The fields are as follows:

Field Name	Value
Username *	test-user
Type *	person
Email *	support@zebrabase.org
Slack	
First name *	Test
Last name *	User
custom field 1	

Waterlog custom field:

The screenshot shows a 'New waterlog' modal form. The form contains the following fields: Date (yyyy-mm-dd), Time (---), Temperature, pH, Conductivity, NH₃ (mg/l), NO₂ (mg/l), and NO₃ (mg/l). A red box highlights a 'custom field 1' input field. A 'Save' button is located at the bottom left of the form. The background shows a table of waterlog entries with columns for Name, Description, Temperature, pH, Conductivity, NH₃, NO₂, and NO₃.

Custom fields can be used for **filtering and ordering** in list views but for that, their **internal name** `custom_field_<N>` has to be used!

3.3.1 Configuration

Step 1: Specify field name

In Admin / Configuration, find the key you want to use:

`custom_fields.<table>.custom_field_<number>`. Currently, there are two types of tables, `user` and `waterlog`. The name supports HTML and FontAwesome 6 icons, so you can use:

`H₂SO₄` -> H₂SO₄

`<i class="fa-solid fa-bomb"></i>` -> bomb icon (FontAwesome)

Step 2: Select fields to be displayed

Then admin has to select which columns should be visible in Admin / Configuration, key `custom_fields.<table>.used_fields`.

When the configuration changes, users need to re-log to see the changes!

3.4 Notification system

There are three **notification methods** in Zebrabase:

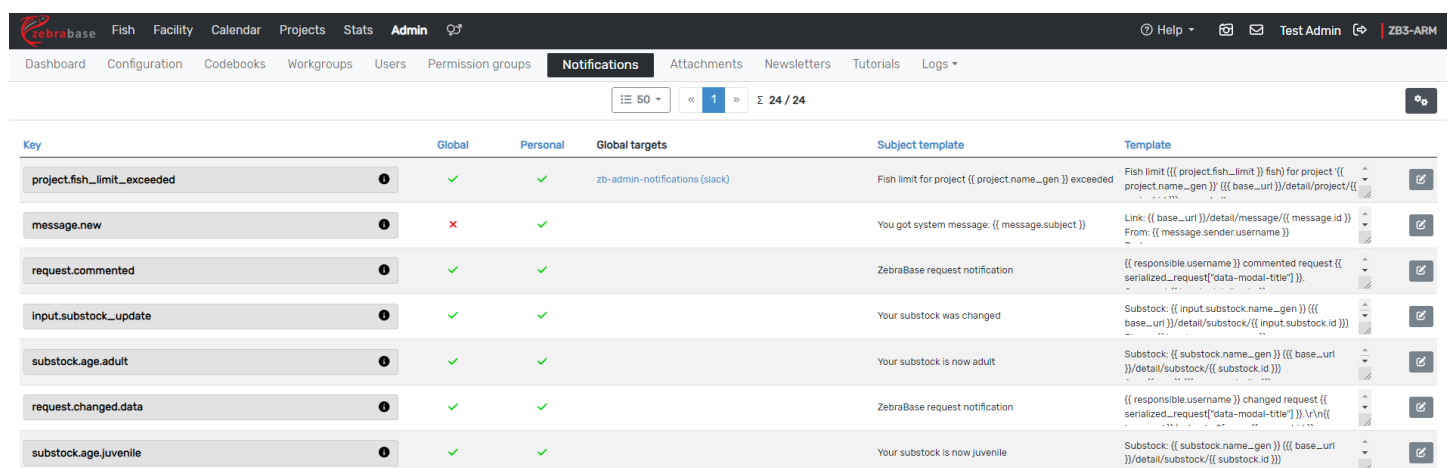
- **System message** (details in [Basic navigation and messaging system](#))
- **E-mail**
- **Slack**

The two **notification types** used in Zebrabase are:

- **Personal notifications:** Messages targeted at a specific user (i.e. the substock owner or a requested assignee etc.).
- **Global notifications:** Messages to every target (a Slack channel for example) about every event (i.e. a notification about a new unassigned request sent to an email conference or to a Slack channel accessible to multiple users).

3.4.1 Events

Zebrabase notifications are invoked by specific **events**, which are listed in Admin / Notifications. Here, the administrator can edit the templates for the subject and the message body for each event (the subject is not displayed by Slack). There are two important read-only flags: **Global** and **Personal**. These indicate which type of notification can be assigned to this event.



Key	Global	Personal	Global targets	Subject template	Template
project.fish_limit_exceeded	✓	✓	zb-admin-notifications (slack)	Fish limit for project {{ project.name_gen }} exceeded	Fish limit {{{ project.fish_limit }}} fish for project '{{ project.name_gen }}' {{{ base_url }}/detail/project/{{ ... }}
message.new	✗	✓		You got system message: {{ message.subject }}	Link: {{ base_url }}/detail/message/{{ message.id }} From: {{ message.sender.username }}
request.commented	✓	✓		Zebrabase request notification	{{ responsible.username }} commented request {{ serialized_request['data-modal-title'] }}.
input.substock_update	✓	✓		Your substock was changed	Substock: {{ input.substock.name_gen }} {{{ base_url }}/detail/substock/{{ input.substock.id }}
substock.age.adult	✓	✓		Your substock is now adult	Substock: {{ substock.name_gen }} {{{ base_url }}/detail/substock/{{ substock.id }}
request.changed.data	✓	✓		Zebrabase request notification	{{ responsible.username }} changed request {{ serialized_request['data-modal-title'] }}.\n{{ ... }}
substock.age.juvenile	✓	✓		Your substock is now juvenile	Substock: {{ substock.name_gen }} {{{ base_url }}/detail/substock/{{ substock.id }}

3.4.2 Global notifications

Global notifications cannot be sent via the inbuilt messaging system.

In case an event occurs, a notification message is generated and sent to all the global targets associated with the given event and to all the users who are involved and have the event notification enabled. The desired targets for global notifications can be added in the event form.

Event description
Fish limit exceeded on some project.

Subject template * Format: Jinja
Fish limit for project {{ project.name_gen }} exceeded

Template * Format: Jinja
Fish limit ({{ project.fish_limit }} fish) for project '{{ project.name_gen }}' ({{ base_url }}/detail/project/{{ project.id }}) exceeded!
Current fish usage: {{ project.total_used_fish }}

Global targets +

Items to select

- zb-notifications
- fish-facility email

Selected items

- zb-admin-notifications

Save Back Back to list

New global targets can be set up in Admin / Codebooks / Notifications: Global Target.

A global target has three attributes: unique **name**, **type** (Slack/email), and **target** (email address OR Slack channel name or personal ID), and can be associated with any event flagged **Global**.

Name *

Type *
Slack

Target (slack channel name / e-mail address) *

Save Save and add another Back to list

3.4.3 Personal notifications

For Slack notifications, a Slack ID must be set up in the user profile first. A tutorial is presented in this [video](#).

Each event that flagged **Personal** can be switched on or off for a specific user. The notifications are set in the user profile settings, either by the specific user (click on the user name in the top right corner, then click Update), or by admin in Admin / Users / Edit. It is possible to choose more options of notifications for one type of event, the message will be sent through all the selected pathways.

Notifications			
Event	Slack	E-mail	System
<code>input.condition</code> Substock conditions changed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.count</code> Input count executed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.productivity</code> Input productivity executed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.productivity.juvenile</code> Productivity changed to juvenile.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.productivity.productive</code> Productivity changed to productive.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.productivity.retired</code> Productivity changed to retired.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.substock_update</code> Substock updated.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>input.transfer</code> Input transfer executed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>message.new</code> User got new system message.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>project.fish_limit_exceeded</code> Fish limit exceeded on some project.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>project.severity_exceeded</code> Maximal severity exceeded on some procedure.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.changed.assigned</code> User was assigned to request by edit request.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.changed.data</code> Request details changed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.changed.status</code> Request status changed.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.changed.unassigned</code> User was unassigned from request by edit request.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.commented</code> Request commented.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.created.assigned</code> New request was created - notify assignees.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.declined</code> Request declined.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>request.unassigned</code> Somebody unassigned from request.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<code>substock.age.adult</code> Substock became adult.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>substock.age.juvenile</code> Substock became juvenile.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<code>substock.age.retired</code> Substock became retired.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Some notifications (for example a notification to the admin about all input counts) cannot be set up this way; using the global notifications is necessary for this.

Event types and endpoint (targeted) users

- **input** - the substock owners
- **request** - the request submitter and all request assignees
- **fishline** - no events with personal handlers available
- **project** - the project leader and the deputy
- **message** - the system message recipients

3.4.4 Video tutorial: Slack notifications



3.5 Printing

- Zebrabase allows printing customizable QR labels with substock information.
- It is recommended to use a Zebra ZD 420 series printer as it was used for testing during the development. Other Zebra printers will most likely function as well. All non-Zebra printers should have ZPL emulation (Zebra Programming Language - https://en.wikipedia.org/wiki/Zebra_Programming_Language) in order to function properly! While it is possible to use other printers and their programming language (for example Intermec), there is no support granted for these printers.
- To ensure good printing quality, use 300 dpi printers.
- You can customize your templates using the code snippets offered here.
- Zebrabase support team can help out with creating templates (only if your printer operates in ZPL II language).

3.5.1 Printwes

To print labels from Zebrabase, you will need [printwes2](#) installed on your printing computer. Printwes provides communication between Zebrabase and the printer. It can work in 2 modes, local and remote.

Local mode

Local mode is easier to install but allows you to print labels only from the computer connected to your printer. To set up Printwes in local mode, install it and set the configuration key `general.printwes_url` to `ws://localhost:8777` (to change the key, go to Admin/Configuration/general.printwes_url). This is the default value of the configuration.

Remote mode

Remote mode allows you to send print requests to your printer from any device connected to the internet. To set up Printwes in the remote mode, you have to:

1. install Printwes
2. register on <https://printwes.org> - the username should be in the form `zebrabase_<facility_prefix>` (e.g. `zebrabase_img`)
3. after successful registration, you will get the "Printwes2 uri":
`wss://printwes.org/ws/pwserver/<username>/?token=<access_token>` (no brackets!)
4. save this URI to Zebrabase configuration (key `general.printwes_url`)
5. save this URI to Printwes client configuration: Printwes2 GUI --> Setup
6. restart the Printwes2 client (if still not working, restart the computer)

3.5.2 Printing templates

There are three codebooks related to printing in the Codebooks section (Admin/Codebooks):

- Printer type
- Printing template
- Printer

The field **Printer type** is just a label that serves for better orientation if you are using multiple different printers. It should be used to distinguish the brand or the programming language of your printers (e.g. Zebra and Intermec). Printer type can be assigned to your printers in the **Printer** codebook. It is also required for printing templates.

The **printing template** is printer-type specific. The template itself uses the templating system [Jinja](#) and you can use all the built-in filters and structures like loops, conditions, macros, etc., and some of our custom filters.

Printer: Printwes2 allows users to connect multiple printers and choose which printer will be used for printing. You don't have to register any printer in Zebrabase if you are using only one.

There is also a configuration key `substock.detail.label_template` that determines which template will be used for the label preview in substock detail. Go to Admin/Configuration/substock.detail.label_template to change it.

Data obtainable from Zebrabase

- `base_url` - URL of your Zebrabase instance (i.e. <https://demo.zebrabase.org>); useful for generating QR codes

- `substock` - all the data we know about the substock:
 - `id` - substock ID (SSID)
 - `stock_id` - stock ID (LSID)
 - `date_of_birth` - date of birth
 - `name_gen` - full generated name
 - `facility_name_gen` - shorter generated name used in facility view
 - `list_name_gen` - shorter generated name used in list view
 - `genotypes_zygosities` - list of genotypes and their zygosities: `genotype_name_gen`, `zygosity_name`
 - `fishline_name` - fishline alias
 - `fishline_name_gen` - fishline generated name
 - `owners` - list of owners: `username`, `first_name`, `last_name`
 - `responsible` - responsible user
 - `workgroups` - list of workgroup details: `name` and some other
 - `funding_name` - funding
 - `project` - project detail or `null` - project detail: `code`, `id`, `name`, `name_gen` (code + name), `leader`, and some other
 - `status` - productivity status
 - `tags` - list of tags
 - `light_name` - light status
 - `diet_name` - diet status
 - `substock_num` - substock number (within sibling substocks)
 - `generation` - number of generation
 - `position_fullname` - full position including room and rack
 - `position_name` - position within rack
 - `room_name` - room
 - `rack_name` - rack
 - `species_name` - species
 - `sex_name` - sex
 - `count` - fish count
 - `suffix` - substock suffix
 - ... and some more. You can use anything contained in API response on endpoint `/api/substock/<some_id>/?detail` (example in API docs), or just write `{{ substock }}` into the template field and the full response will be returned
- `parents` - list of parent substocks

Zebrabase specific filters

- `substock_name_no_html` - this filter removes all html tags and puts the upper index into brackets

- `unicode` - transcribes UTF characters to the most similar ASCII
- `smart_split(max_row_length, max_row_count=10, delimiter=',')` - attempts to split a string value by the `delimiter` and to build rows of text out of the parts; if the resulting row count extends `max_row_count`, standard split by `N` characters is used instead (see example bellow); available from 3.1.2

ZPL basics

This is the basic information needed to customize your QR labels written in ZPL. For more details, go to [ZPL programming guide](#).

- `^XA`, `^XZ` - start/end of label
- `^CF` - sets font for the whole label (syntax: `^CF0,N,25,20`) (font 0 is well scalable and recommended)
- `^A` - sets font only for the next entry (syntax: `^A0N,25,20`, orientation can be either N = normal, R = 90 degrees, or B = 270 degrees)
- `^F0` - position of the following field - it is in absolute coordinates from the left and top edge (x, y)
- `^FD` - text string for the field
- `^FS` - end of field definition
- `^PW` - sets width of label
- `^BQ` - QR code (syntax: `^BQU2,5` - the last variable indicates the size of the code and can possess values 1-10)

The parts between square brackets - `[...]` - are optional, they can be omitted. If these parts are specified, the square brackets are omitted.

Code snippets

These snippets are examples of how you can get some basic data for the labels. You can mix and match them to create your custom label, just define the position of each field in front. It is also possible to specify the font. The full code describing one field could look like this:

```
^A0,40,40 ^F0200,170 ^FDD0B: {{ substock.date_of_birth }}^FS font position text
```

Code snippets

Declaration of printer-friendly version of substock name

```
{% set ss_name = substock.facility_name gen | substock_name no_html | unicode %}
```

Split substock name to more lines (3 rows with 40 characters in each)

```
{{ ss_name[:40] }}`, `{{ ss_name[40:80] }}`, `{{ ss_name[80:] }}
```

Declaration of printer-friendly version of substock name + splitting using `smart_split` (40 characters per row, 3 rows)

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode | smart_split(40, 3) %}
```

List of owners

```
^FDOW: {{ substock.owners | map(attribute='username') | join(', ') }}^FS
```

QR code (size 5)

```
^BQU,2,5 ^FDLA,{{ base_url }}/substock/{{ substock.id }}^FS
```

Date of birth

```
^FDDOB: {{ substock.date_of_birth }}^FS
```

Full ID (LSID-SSID)

```
^FDID: {{ substock.stock_id }}-{{ substock.id }}^FS
```

Position

```
^FDPOS: {{ substock.position_fullname }}^FS
```

Project code

```
^FDPRJ: {{ substock.project.code }}^FS
```

F-generation

```
{% if substock.generation == None %} ^FDF: {{ substock.generation }}^FS {% else %} ^FDF: F{{ substock.generation }}^FS {% endif %}
```

Substock description (35 characters maximum)

```
^FDDES: {{ substock.description[:35] }}^FS
```

Fish count

```
^FDN: {{ substock.count }}^FS
```

List of tags

```
^FDTAG: {{ substock.tags | map(attribute='name') | join(',') }}^FS
```

List of genotypes

```
^FDGNT: {{ substock.genotypes_zygosities | map(attribute='genotype_name_gen') | join(',') | substock_name_no_html }}^FS
```

First genotype in list and its zygosity

```
^FD{{ substock.genotypes_zygosities[0].genotype_name_gen | substock_name_no_html }}^FS ^FD{{ substock.genotypes_zygosities[0].zygosity }}
```

Printing of images

If you want to have an image included in your label, you can create a ZPL command for the image for example by using [Labelary](#). Paste your label template into the viewer, click "Add image" and select your image from a file.

Conditions and loops

You can use any functions available in [Jinja](#) in your label templates, including if statements and for loops:

```
{% if substock.generation == None %} ^FDF: no input^FS {% elif substock.generation == 0 %} ^FDF: F0 (primary generation)^FS {% e
^FD {% for genotype in substock.genotypes_zygosities %} {{ genotype.genotype_name_gen | substock_name_no_html }} {{ genotype.zyg
```

Examples of use

How smart_split works

Example

Result

```
# second and third sections are short enough to fit in one row {% set x = '1234567,123,456,123456' | smart_split(10, 3) %} {{ x[0]
, x[1], x[2]}}
```

```
1234567, 123,456, 123456
```

```
# second section is longer than 10 characters --> it's split {% set x = '123,12345678987,123,456' | smart_split(10, 3) %} {{ x[0]
, x[1], x[2]}}
```


```
123, 1234567898 7,123,456
```

```
# `x` would have 4 rows in this example # so smart split is replaced by normal split {% set x = '123,1234567898,12345,456,789' |
split(10, 3) %} {{ x[0]
, x[1], x[2], x[3]}}
```

```
123,123456 7898,12345 ,456,789
```

Example (very simple)

**gata1:DsRed (?),gfi1b1_pME – b – globin:EGFP
(?)**



DOB: 2023-02-14
ID: 7061-15174 [9]
POS: F10:C05
OW:
PRj:

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode %} ^XA ^CFN,20 ^PW450 ^LL0300 ^F021,130 ^BQN,2,5
```

Example with smart_split

**gata1:DsRed (?),
gfi1b1_pME-b-globin:EGFP (?)**



**DOB: 2023-02-14
ID: 7061-15174 [9]
POS: F10:C05
OW:
PRj:**

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode | smart_split(40, 3) %} ^XA ^CFN,20 ^PW450 ^LL03
```

Please note that in this case, `ss_name` is not a string but an array of strings. Standard indices `0, 1, 2, ...` are used instead of range indices `:40, 40:80, 80:` used in the previous example.

Example (custom 1)

**gata1:DsRed (?),
gfi1b1_pME – b – globin:EGFP (?)**



**DOB: 2023 – 02 – 14
F: None
Ethic code:
N animals: 30
Project code:**

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode | smart_split(40) %} ^XA ^CF0,22 ^PW450 ^LL0300
```

Example (custom 2)

**gata1:DsRed (?),gfi1b1_pME – b – glob
in:EGFP (?)**




ID = 7061 n = 30 POS: F10:C05
TAG: Chov
DES: GMO APPROVED

GEN: None
DOB: 2023 – 02 – 14 OW: pavla

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode %} ^XA ^CFN,22 ^PW456 ^LL0256 ^A0,23,25 ^F011,10
```

Example (custom 3)

**gata1:DsRed (?),
gfi1b1_pME – b – globin:EGFP (?)**

 **DOB: 2023 – 02 – 14**
 **ID: 7061 – 15174**
 **POS: F10:C05**

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode | smart_split(30, 3) %} ^XA ^CF0 ^PW800 ^LT20 ^F
```


Example (custom 4)

**gata1:DsRed,
gfi1b1_pME – b – globin:EGFP**

jovana

**DOB (Y – M – D): 2023 – 02 – 14
ID:7061 – 15174**

```
{% set ss_name = substock.facility_name_gen | substock_name_no_html | unicode | replace(" (?)", "") | smart_split(30, 3) %} ^X
```

3.6 Printing troubleshooting

Here you can find a troubleshooting checklist for printing QR labels from Zebrabase. If you are unable to print QR code labels, try to locate the error using this checklist. If you answer "no" to a question, try to fix the problem at that level first and continue with the next step only if the problem is fixed.

1. Is the label printer switched on and connected to the computer?
2. Is the label printer listed in the available printers on the computer?
3. Does the label printer print labels from other programs?
4. Is the Printwes2 Client program installed and running?
5. Is the Printwes2 Client Info dialog showing host=localhost, port=8202, websocket=connected? (This does not apply if the Printwes2 Client is in remote mode; in that case, the host and port values will be different. Launch the Info dialog by right-clicking the Printwes2 Client icon in your taskbar and selecting "Printwes Client - Info".)
6. Is the label printer visible in the list of printers in Printwes2 Client/Setup? (Launch the Setup from the taskbar icon by selecting "Open GUI" and clicking on the Setup link on the GUI webpage.)
7. Can you print a label from the Printwes2 Client GUI web page "Print"? (As your raw data, you can use the example below (it is designed for label size 300x300 dots)
8. Are you running Zebrabase on the computer where the Printwes2 Client is installed? (This does not apply if the Printwes2 Client is installed in the remote mode; in that case, it is possible to print also from other devices.)
9. If you have more printers available, have you specified the printer that should be used by Zebrabase? This can be done in several ways:
 - a) Define the label printer as the default computer printer.
 - b) On the Setup page in Printwes2 Client, select the Printer and click [Save setup].
 - c) At <https://zebrabase.org/list/printer>, click the [Add printer] button and specify your printer in the following form. The printer(s) specified here will be then available in the Zebrabase "Print QR" dialog.
10. Are you selecting the correct template (and printer, if more printers are defined according to point 9.c) for printing the label?

ZPL example

ZPL language works on Zebra printers and some compatible models. It will not work on Intermec printers.

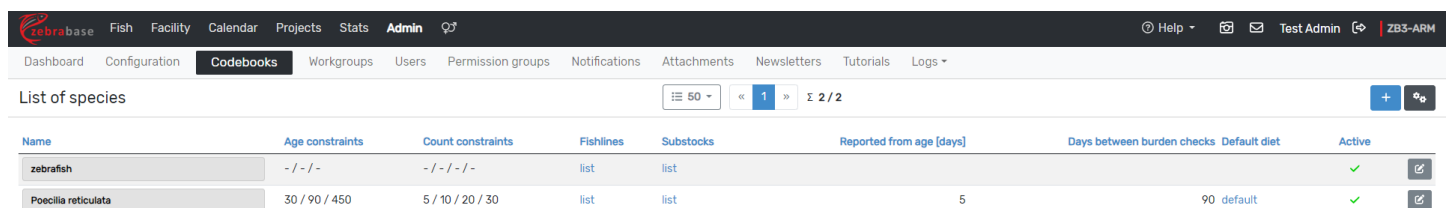
```
^XA ^CF0,20,14 ^PW300 ^LL0300 ^F030,50 ^BQN,2,5 ^FDLA,https://zebrabase.org^FS ^F011,20 ^FSubstock generated name^FS ^F0160,50
```

3.7 Multi-species module

Extensions needed: multi-species

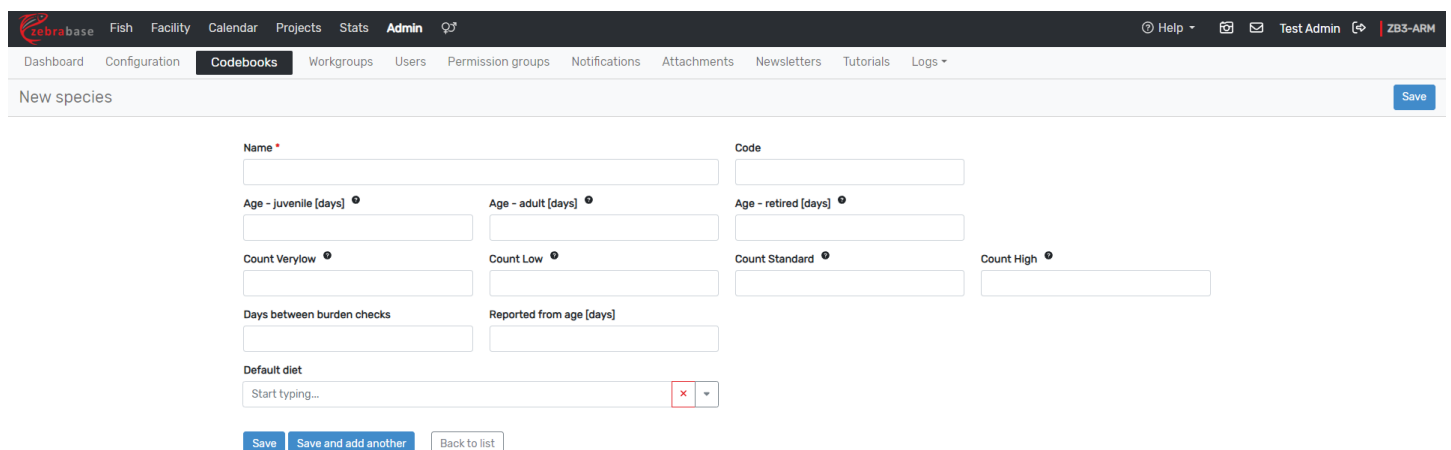
Zebrabase is suitable for any species of fish or any animals kept in groups. You can detail your species-specific parameters as age constraints, or liminal animal densities for tanks. You can also filter animals according to their species, or color-code them based on the species in the Facility view.

If a parameter is not filled in, it automatically acquires Zebrabase default values (which are pre-set for *Danio rerio*). The default values can be inspected in Admin / Configuration. Some of these default values can be changed by the admin, however, great care should be taken when changing any values in the Configuration.



Name	Age constraints	Count constraints	Fishlines	Substocks	Reported from age [days]	Days between burden checks	Default diet	Active
zebrafish	- / - / -	- / - / - / -	list	list				✓
Poecilia reticulata	30 / 90 / 450	5 / 10 / 20 / 30	list	list	5		90 default	✓

New species can be added in Admin / Codebooks / Fishline / Species / Add species.



Name *

Code

Age - juvenile [days] •

Age - adult [days] •

Age - retired [days] •

Count Verylow •

Count Low •

Count Standard •

Count High •

Days between burden checks

Reported from age [days]

Default diet

Start typing... [x] ▾

Save Save and add another Back to list

Species code

When adding a new species, it is useful to add an abbreviation or a code for the species. This may be up to six numbers, letters, or a combination of the two.

Life stages

The life stages are automatically updated in Zebrabase for each species. **Age: juvenile, adult, and retired**, refers to the number of days from the birth of the individual until the animal transforms to the stated developmental stage. This means the age (in days) when the fry becomes juvenile, adult, and finally retires. The retirement age is the age when the fish are usually removed from the breeding program.

Animal density

For welfare reasons, fish should be kept in densities between a minimum and a maximum. Zebrabase will color-code tanks according to their animal density in the Facility view.

The screenshot displays the Zebrabase Facility view interface. At the top, there is a navigation bar with tabs for Fish, Facility, Calendar, Projects, Stats, and Admin. Below this, a sub-navigation bar shows 'Facility view' and options for Rooms, Racks, Design rack, and Water system. A search bar and several action buttons (Count, View, Transfer, Batch Select, Block, Price) are visible. The main area is a grid of tanks, each represented by a colored square with a label and a date. The tanks are color-coded based on animal density, with a legend on the right side. The legend shows color-coded boxes for density ranges: 1 (blue), 2-5 (orange), 6-10 (green), 11-20 (yellow), 21-30 (red), and ≥ 31 (dark red). The tanks are labeled with various genetic backgrounds and dates, such as 'gata1:kalt4^{+/+};UAS:kae^{de}' and 'WT (AB) 6794'. A 'Legend' panel on the right also shows 'Count' and 'Tags' for 'Poecilia reticulata', including 'dead', 'low eggs', 'unproduc.', and 'genotyping'.

The **count constraints** are the limits for the color-coding of the tanks. The color-coding levels are: very low, low, standard, high, and very high. Hence there are the four values of count constraints that define the ranges. Any tank filled with more fish than what is the limit for count high is considered very high. The color-coding is visible when selecting the option **Count** in the **Status** drop-down menu in the Facility view.

Days between burden checks

In genetically modified lines, a possible genetic burden should be confirmed after 3 generations of breeding. Currently, this is represented by the number of days by which typically 3 generations have been bred.

Reported from age

This is the age by which the animals must be reported when used in experiments. In *Danio rerio*, this is typically 5 days.

Default diet

Any default diet for each species can be selected. Diets can be defined in Admin / Codebooks / Diet.

3.8 Batch actions

Added in v. 3.4.0

Extensions needed: batch

User permissions needed: batcher

It is possible to perform certain actions on a batch of substocks. You can do so **1) in Substocks**, by clicking on Batch actions and selecting the substocks directly from the list, or **2) in Facility view**, where clicking on Batch select initiates the batch action mode. Once you have selected the substocks, choose an action from the offered options.

In the action form, the selected substocks are listed on the left panel. If there are any substocks where the count action cannot be performed (for example a substock with undefined start count in a count action) then they will be highlighted in red. You will need to remove these substocks from the batch by clicking on the cross in the upper right corner of each substock.

Problematic substocks

Burdened substock

Count - batch

Substock *

Filter: All [1] Problematic [2]

Add a substock...

7510-16672	*	B01:A06	???	155
gata1:kalta4 2023-11-15/1 (1)				
7509-16671	*	B01:A07	???	161
gata1:kalta4 2023-11-09/2 (1)				
7468-16578		B01:B03	21	182
p53 ^{M214K} 2023-10-19/1 (2)				
6157-16306		B01:C03	9	814
gata1:kalta4 2022-01-25/2 (8)				
6794-14493	WT	B01:B04	20	533
(AB) 2022-11-02/3 6794 (3)				
6600-14039		B01:C04	8	630
gata1:kalta4 2022-07-28/1 (10)				
6548-13920	WT	B01:B05	20	645
AB (DR) 2022-07-13/2 (6)				

Performed * date of execution Note

2024-04-18

Count type * Count *

died after 90dpf

all fish *

1 Burdened substocks

Project *

1/2022 - Burdens + New lines (not active)

Severity

below threshold mild moderate non-recovery severe

Check form Save action

The count can be performed even on burdened substocks. A burdened substock is marked with a pictograph of a weight. The form contains a section where the project of the burdened substock must be selected and the procedure severity confirmed.

If there are any substocks in experiments included in the batch, the form will automatically include sections for each project where the procedure and its severity must be recorded.

The form should be reviewed for potential discrepancies by clicking on the Check form button. If no issues have been found the form can be saved.

The action confirmation message will return a link to the list of substocks for which the action has succeeded or failed.

3.9 Attachments

A list of all attachments is available for viewing and downloading in Admin / Attachments.

Attachment type (such as "Standard Operating Protocol", etc.) can be specified by clicking on Update. New attachment types can be defined in Admin / Codebooks / Attachment Type.

Also, a confirmation request (none/optional/mandatory) can be set up to confirm that the document has been read.

The screenshot shows the 'Edit attachment' form in the ZebraBase Admin interface. The breadcrumb trail is 'Admin / Attachments'. The attachment title is 'Rozhodnutí o schválení PP (AVCR 7345 2022 rozhodnuti.pdf)'. The form contains the following fields:

- File (max. size: 50MB)**: A text input field with the value 'Rozhodnutí o schválení PP'. A button 'Vybrat soubor' (Choose file) is next to it, with the text 'Soubor nevybrán' (File not selected) below it.
- Attachment type**: A dropdown menu with the value '-- none --'.
- Confirmation**: A dropdown menu with the value 'no confirmation'.
- Description**: A large empty text area.

At the bottom of the form, there are three buttons: 'Save', 'Back', and 'Back to list'.

3.10 How to use the animal reporting module

This is an introduction to using the Animal reporting module (ARM). The module is only available to **Premium customers** and is **enabled after sending a request**, ideally via helpdesk.

The animal reporting module is designed for reporting the usage of animals for experimental procedures according to the Felasa recommendations and the **2010/63/EU directive** on the protection of animals used for scientific purposes. According to this directive, each animal capable of independent feeding (in zebrafish, this applies to animals older than 5 dpf; days post fertilization) that has undergone one or more procedures with a severity level higher than threshold, must be counted and reported. The results are displayed in the Experiment report table as summary (EU). The prospective severity of the procedure must be defined before each experiment, and updated if changes in severity occur during or after the experiment. If an individual undergoes more procedures with various levels of severity during one experiment, only the highest experienced severity is reported. If more animals undergo procedures and it is not possible to distinguish the exact animals that were actually treated, then all the animals in the group are reported as having undergone a procedure with the highest severity of all experienced severities.

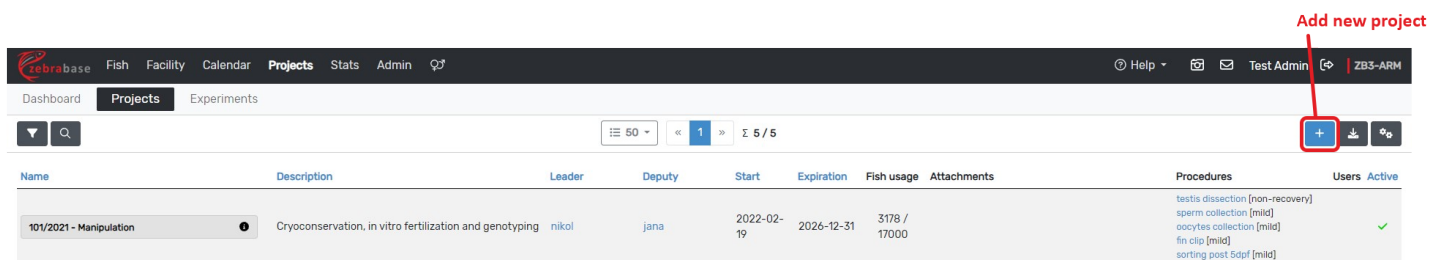
To cater to non-EU countries, the reporting system also records all procedures that the animals have gone through including all the severities. This means that one animal can be reported multiple times, once for each procedure it has undergone. This statistic is displayed in summary (all) in the Experiment report.

To start logging the animal usage, the facility admin first needs to set up a project or projects under which the animals will be reported.

3.10.1 Projects

Prior to starting the experimental work, outlining such projects is usually required by the authorities. These projects may run over several years and usually include the maximum number of animals to be used during this project and the list of experimental procedures and their estimated prospective severities. The experiment can start once the project has been approved.

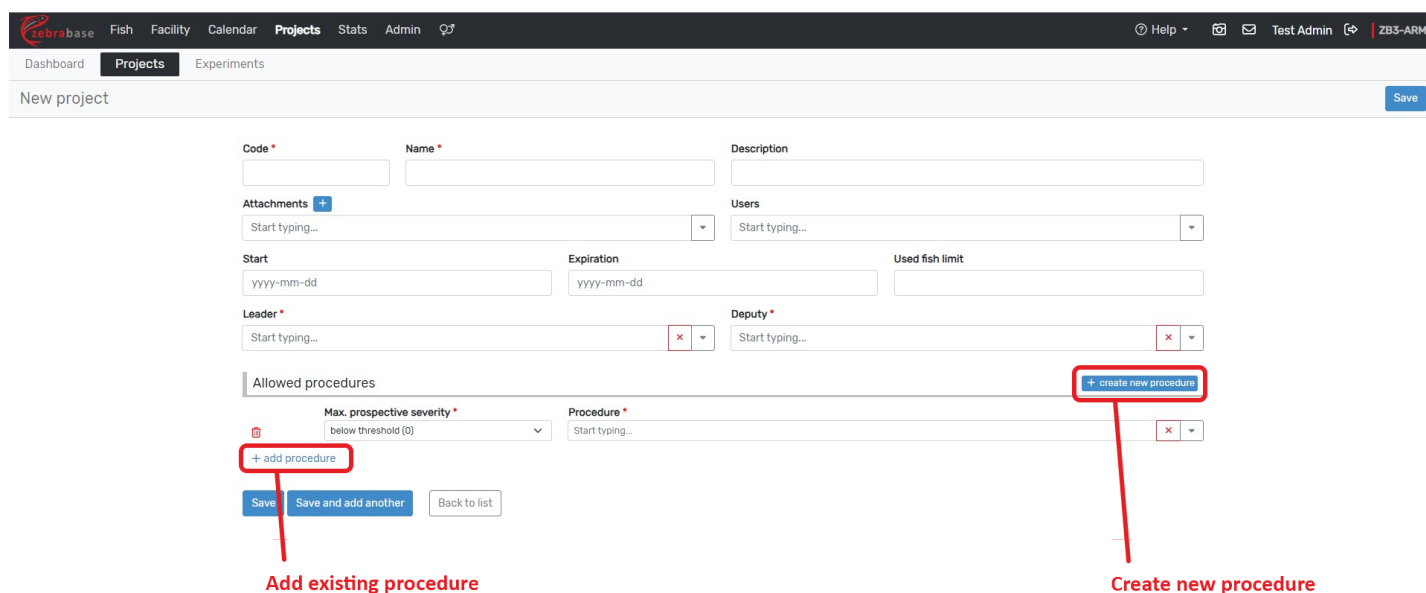
Adding a new project to Zebrabase is possible in Projects/Projects (or in Admin/Codebooks/Projects/Project). To add a project, click on the blue plus button.



The screenshot shows the Zebrabase interface with the 'Projects' tab selected. A table lists existing projects, including one named '101/2021 - Manipulation'. In the top right corner, a blue plus button is highlighted with a red box and a red arrow pointing to it, labeled 'Add new project'.

This opens a **New project** form. The mandatory fields are a name and a code unique to the project, the project leader and the deputy (the same user can be filled in both fields). The optional fields are the Start and the Expiration dates of the project, Users and Used fish limit. The field **Users** serves as an informative list with no connected functionalities. The **Used fish limit** number is necessary for the capacity indicator bar to work. The actual project documentation or approval can be uploaded as a PDF attachment.

Finally, the **Procedures** that will be performed within the experiments in this project must be included. It is possible to use procedures that were already defined, or to create new procedures. The **Maximum prospective severity** of each procedure can be selected from the drop-down menu. The default values for procedure severity are: below threshold (0), non-recovery (1), mild (2), moderate (3), and severe (4). These settings can be changed in Admin/Codebooks/Severity. Procedures can be managed independently on projects in Admin/Codebooks/Procedures.



The screenshot shows the 'New project' form. At the bottom, there is a section for 'Allowed procedures'. A red box highlights the '+ add procedure' button, with a red arrow pointing to it and the label 'Add existing procedure'. Another red box highlights the '+ create new procedure' button, with a red arrow pointing to it and the label 'Create new procedure'.

After the project is successfully set up, experiments can be added.

NOTE: According to the regulations, breeding transgenic **fish with burdened phenotype** must be reported as well as the **genotyping of all fish**. For this purpose, we recommend setting up two separate projects that will serve as default projects for genotyping and reporting actions on burdened substocks that are not used in another experiment. For these projects, the procedures do not need to be defined; the procedure field can be deleted when creating the project by clicking on the bin icon on the left. If a substock is already subscribed to an experiment, the user can decide in the genotyping form under which project this genotyping should be reported.

3.10.2 ARM configuration

In the admin dashboard, there is the ARM configuration table which is designed to help with the configuration and give a warning if some of the important elements are missing. If an element is missing, there will be a red warning button, and if the element has not been set up yet, the button will be yellow. Ideally, all buttons should appear green. In the next step, after you have designed the default projects for genotyping and burdened fish breeding, please check this table thoroughly.

The screenshot shows the ZebraBase Admin dashboard with the 'Animal Reporting Module (ARM)' configuration page. The page is divided into two main sections: 'Mandatory configuration' and 'Recommended configuration'. Each section contains a list of configuration items with a 'set' button and a status indicator (green, yellow, or red).

Configuration Item	Status
Pre-set. Do NOT touch! defaults.actions.experiment_end_procedure	Green
Pre-set. Do NOT touch! defaults.actions.experiment_end_counttype	Green
Set default project for reporting all genotyping actions. project.default_genotyping_project	Green
Set default method for reporting all genotyping actions. defaults.actions.genotyping_default_method	Yellow
Set default project for reporting burdened substocks (animals). project.default_burden_project	Green
Define default severity for keeping burdened fish. project.default_burden_severity	Green
Define default count type for actions in experiment (already preset to the recommended non-lethal procedure). defaults.actions.experiment_default_counttype	Yellow
Set the observation period (in days) if you wish to extend the reporting period past the experiment termination date. project.experiment_observation_period	Green
Set the age threshold (in dpf) for subtracting fish from the project capacity (already preset to 5 dpf for zebrafish). project.fish_reported_age	Green
Set the interval (in days) between the first and second burden check. project.fshline_burden_check_interval	Green

Configuration explanations:

- **Mandatory configuration** - Should not be changed!
- **Observation period** - The number of days after the final experimental procedure after which any death in a substock that has been in the experiment is attributed to the last experimental procedure. For the duration of the observation period, it is impossible to terminate the experiment. It is important NOT TO ADD ANY NEW PROCEDURES after the final one, because this would reset the observation period. Any deaths after the final procedure must be attributed to one of the already existing procedures.
- **Burden checks** - In every new fishline, there should be a check for burdened phenotype, the first check should be performed as soon as possible. This should be noted in the fishline detail (Edit burdens icon) and after a defined period, a second check should be performed to confirm the burden. The interval between the burden checks is set to 270 days by default (3 generations, i.e. F2), and can be changed in Configuration/project.fishline_burden_check_interval.

When the ARM configuration is set up, you can start defining the individual experiments.

3.10.3 Experiments

Each experiment runs under a previously defined project and contains one or more procedures. A list of all experiments can be found at Projects/Experiments. To add a new experiment, click the blue plus button.

When a new experiment is being set up, substocks can be added right away, or they can be added later. At the moment a substock is added to an experiment, the fish count is included in the project capacity indicator bar. If there are some unused fish at the end of the experiment, they are subsequently removed from the project capacity indicator.

NOTE: Once the project capacity is filled to 75% or more, the color of the indicator bar will change from green to red.

3.10.4 Quick reporting walkthrough

In some cases, the substock doesn't need to stay subscribed to an experiment, and can be removed from it right away. To make such reporting of individual procedures smoother, there is the option to go through all the necessary entries at once in a series of consequent action windows.

1. Add to experiment

In the substock detail, click the flask icon (Project experiment). This will open the **Add to experiment** action window, where you can choose under which project and which experiment you want to subscribe to the substock. The **Continue to count** button will save the action and open the next window in the sequence.

2. Count

In this window, you can specify the procedure and how many fish from the substock underwent it. The toggle button **all fish** will assign all fish in the substock. The severity of the procedure can be altered at this point. If you want to unsubscribe the stock from the experiment in the following step, click **Save and leave the experiment**.

If you need to make this action for a substock that was already assigned to an experiment, go to substock detail and click **Count (Procedure)** on the upper panel.

3. Remove from experiment

This action unsubscribes the substock from the current experiment. You can finalize the severity of the procedures (based on the actual observed severity) and specify the number of unused fish (a new substock consisting of the unused fish will be created, and the used fish will keep the original substock name and position).

Remove from experiment

Performed ^{*} [●] date of execution Note

2023-12-14

Substocks ^{*}

Substock	Experiment	Fish count	Overall severity [●]	Unused fish
7479-16631 CASPER	101/2021_E02 - sort post 5 dpf	13	mild	0

Check form Save action

If you need to make this action outside the sequence, go to substock detail and click **Remove from experiment** on the upper panel.

Before an experiment is finished, **all substocks must be removed from the experiment.**

3.10.5 Reporting tables

In the project detail, below the summary information, there are three reporting tables, **Experiment report**, **Burdened fish report**, and **Genotyping report**. These are detailed reports listing the substocks that went through a procedure, the date of the procedure, the name of the procedure, the associated severity of the procedure, the count of animals that were used, the name of the user, and any eventual note.

EU summary counts each used fish only once, regardless of how many procedures it has been through.

Full (all) summary counts each fish occurrence in a procedure.

Experiment report

2023

Year summary (EU):	below threshold: 0	non-recovery: 0	mild: 10	moderate: 18	severe: 0
Year summary (all):	below threshold: 5	non-recovery: 3	mild: 7	moderate: 5	severe: 5

5/2022_E01 - ARM experiment [2023-05-09 - ?]

Exp. summary (EU):	below threshold: 0	non-recovery: 0	mild: 10	moderate: 18	severe: 0
Exp. summary (all):	below threshold: 5	non-recovery: 3	mild: 7	moderate: 5	severe: 5

7014-15089 gata1:SW⁹; [2023-01-31/1 (3)]

→ oocytes removal [2023-05-09] [98]	5 / 15	non-lethal procedure	below threshold	violapavlova	---
→ testis dissection [2023-05-09] [98]	3 / 15	non-lethal procedure	non-recovery	violapavlova	---
→ transplantation [2023-05-09] [98]	2 / 15	non-lethal procedure	mild	violapavlova	---
→ electroporation [2023-05-09] [98]	1 / 15	non-lethal procedure	moderate	violapavlova	---
→ leave experiment [2023-05-09] [98]	1 / 15	non-lethal procedure	moderate	violapavlova	---
→ leave experiment [2023-05-09] [98]	15 / 15	left experiment	moderate	violapavlova	---

leaving experiment [2023-05-09] unused fish: 0

- When animals die during an experiment, the number of individuals is highlighted in red and appears with a minus. When the procedure severity exceeds the prospective severity, it is also highlighted in red.
- When a substock is added to the project **before reaching 5 dpf**, the number of fish is subtracted from the project capacity. Actions on or deaths of fish younger than 5 dpf are shown in the report, but the row is colored grey and the counts are not included in the project summary calculations.
- Substock actions on both burdened or not burdened substocks that happen within an experiment are reported in the **Experiment report** table. Actions on burdened substocks that are not involved in an experiment, but are subscribed to a project, are included in the **Burdened fish report** table in the project detail.

	In experiment	Not in experiment
Burdened substocks	Experiment report	Burdened fish report
Not burdened substocks	Experiment report	Not reported

The yearly EU summary in the Experiment report is the sum of all animals that died or left the experiment after being treated in a procedure with **higher than below threshold severity** (i.e. mild or more severe procedures). Animals that have undergone at least one procedure with higher than threshold severity are marked as used by a **recycle icon** behind the name in the substock detail, or the word **reused** in

the pop-up detail. If animals from one substock have gone through more procedures, only the highest severity is reported in summary (EU) and all procedures are reported in summary (all).

Used in experiment

Substock #16631 - CASPER 2023-10-31/1 (3)

Species: 16631
 SSID: unproductive
 Status: unproductive
 Tags: CASPER 2023-10-31/1
 Stock: 3 / 10
 Fishline: CASPER
 Suffix: AB

Label: CASPER
 DOB: 2023-10-31 [FNone]
 ID: 7479-16631 [3 / 10]
 POS: C04:E10
 OW:

The fish usage is calculated as the sum of fish used in an experiment (all substocks added to the experiment minus the fish that were removed from the experiment unused or if they died before reaching 5 dpf - in the list of count types, this is represented as counts.to.experiment), plus the deaths or euthanizations of burdened fish (counts.as.burdened.death), plus the genotyped fish (every time genotyping is performed, even with the same substock). The notification after the limit was exceeded can be enabled by setting on project.fish_limit_exceeded.

Fish usage indicator

Project #4 - 101/2021 - Manipulation

Fish usage: 3223 / 17000

Description: Cryoconservation, in vitro fertilization and genotyping
 Active:
 Leader: nikol
 Deputy: pavla
 Start: 2022-02-19
 Expiration: 2026-12-31
 Fish usage: 3223 / 17000
 Used fish: 80 + 20 + 3123
 Fishlines:
 Substocks (ARM):
 Attachments: 101-2021-P
 Procedures: testis dissection [non-recovery], sorting post 5dpf [mild], fin clip [mild], oocytes collection [mild], sperm collection [mild]

Experiments

Name	Actual fish count
101/2021_E01 - cryo	0
101/2021_E02 - sort post 5 dpf	None
101/2021_E03 - IVF	None

NOTE: The numbers in the experiment report and on the project capacity indicator bar may slightly differ because of the order of operations and calculations performed in Zebrabase. This is not a bug. The project capacity indicator presents a rough estimate of used fish and is designed for the sole purpose of giving a warning and sending a notification when the project capacity has been exceeded.

WARNING: The numbers in the project capacity bar can also become inaccurate after some experimental action is subsequently **deleted by the admin**. If the project capacity is exceeded by a user error or there is generally a large difference between the capacity bar and the actual fish usage, please **contact the help desk** to have the numbers clarified and identify possible faulty actions. Always use the experiment report table for reporting fish usage to authorities, and not the capacity indicator.

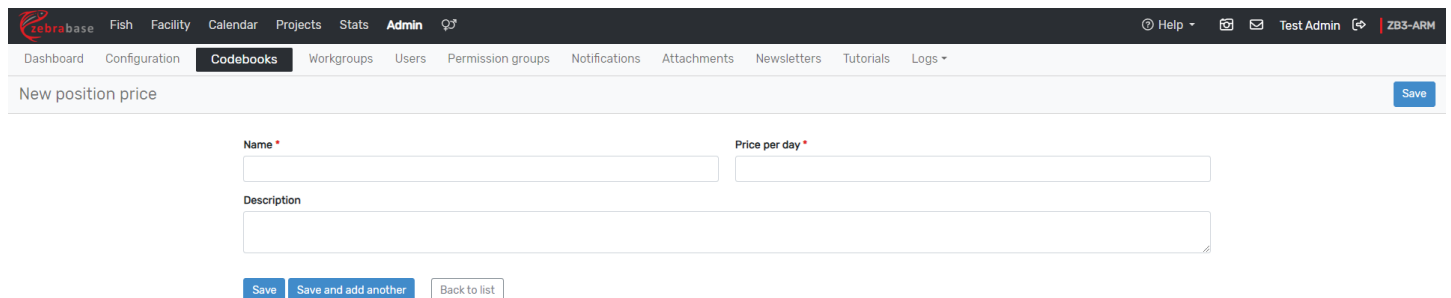
3.11 Pricing and billing

Extensions needed: tank pricing

In Zebrabase, it is possible to calculate billing for facility users based on their tank usage. A price is **associated with individual tank positions** and in the billing calculations, it is multiplied by the number of days the position was occupied, within the defined billing period. You can calculate billing for workgroups, and individual users, based on the defined funding, and more.

Step 1: Define prices

To set up, first define the tank prices you will use in Admin / Codebooks / Position Price. For each price, it is mandatory to fill in its name (for example "Academic", "Juvenile fish", ...) and price per day. Optionally, you can add a description.



The screenshot shows the 'New position price' form in the Zebrabase Admin interface. The form is located under the 'Codebooks' menu. It has a title bar 'New position price' with a 'Save' button on the right. The form contains three input fields: 'Name *' (required), 'Price per day *' (required), and 'Description'. Below the form are three buttons: 'Save', 'Save and add another', and 'Back to list'.

Step 2: Assign prices to positions

Once your prices are defined, you can assign them to individual positions in your facility. Go to Facility view and in the upper right corner, select Price to get into the pricing mode. Then you can select one or more positions you want to assign the price to, and select the price you want to use in the panel on the right. Confirm the action by clicking Assign price.

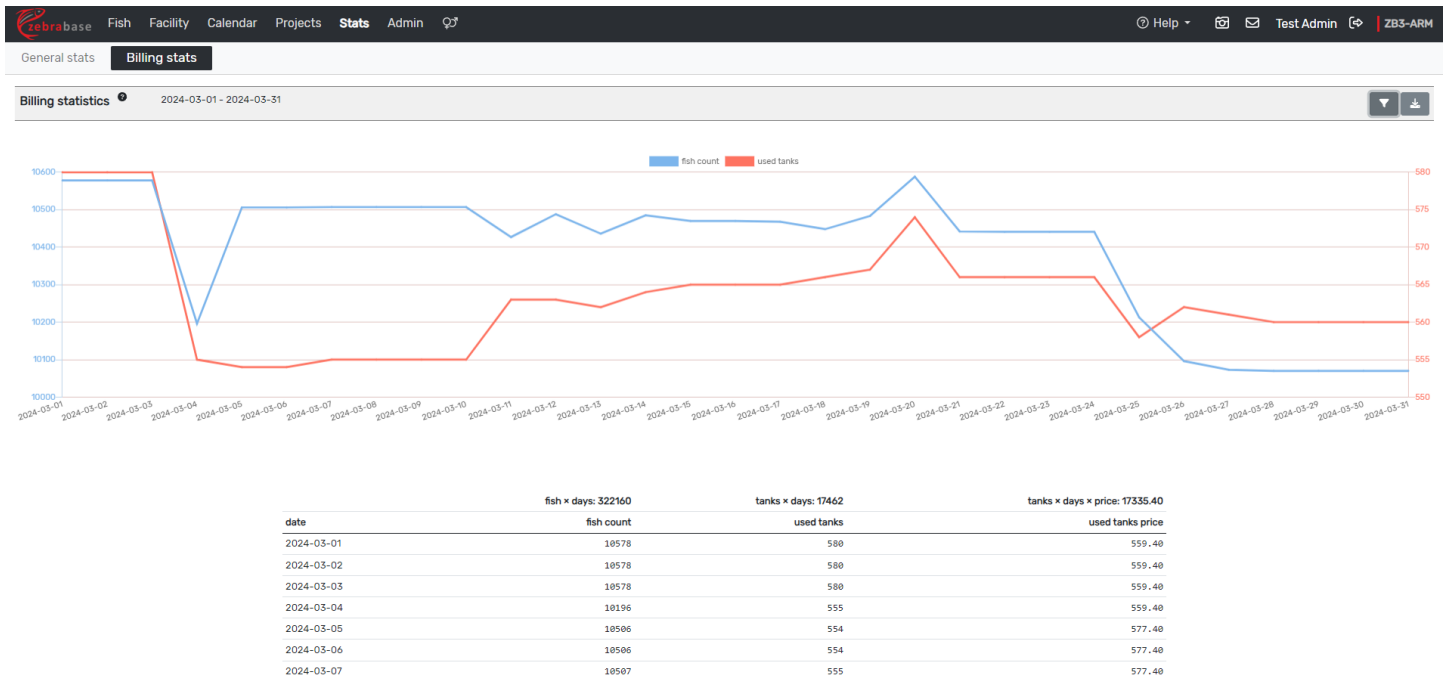
You can view the prices by switching the color coding of your tanks. Click on Status and select Price from the drop-down menu. When you place a substock into a priced position, it will be added to the billing statistics.

Step 3: Billing statistics

At the end of each day, billing statistics for that day are calculated. You can view them in Stats / Billing stats (this page is only visible to admins). In the table on the bottom of the page, each row represents one day and displays three values:

- **fish count** - total number of fish in positions with a defined price
- **used tanks** - total number of used tanks, i.e. substocks assigned to positions with defined price
- **used tanks price** - total price for used tanks - your defined prices will be used in the calculation

In the top row, there are the sums calculated for the whole billing period (i.e. all rows of the table summed up).



Clicking on the filter icon in the top right corner opens filtering options, where you can define the billing period and which tanks should be included in the calculation (for example, only tanks belonging to a certain workgroup or with a certain type of funding can be counted).

You can download the billing table by clicking the download icon in the top right corner.

Please notice!

- If a substock is owned by 2 workgroups and billing stats are computed for workgroups separately, the substock will be counted twice (once for each of the workgroups)
- Filters are applied on current data only - i.e. filtering by room counts only fish placed in the selected room
- Wider tank counts as one
- Split or shared tank counts as one

3.12 Crons

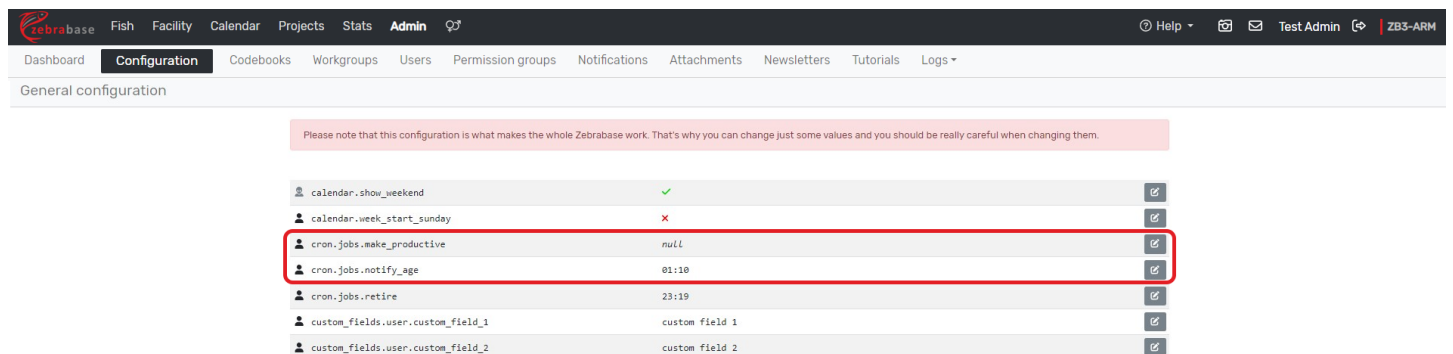
The cron utility is a job scheduler on Unix-like operating systems. Cron is used to scheduling jobs (commands or shell scripts usually intended for system maintenance or administration) to run periodically at fixed times.

In Zebrabase, two jobs can be executed by cron:

- "retire" - retires animals older than a defined threshold
- "make productive" - the productivity of juveniles older than a defined threshold is updated to "Productive"

The cron logs the job's name, timing, outcome, and the number of updated substocks (if any). If the job cannot be executed an error message and additional details are logged. You can check the cron logs in Admin / Logs / Cron log.

The jobs are executed daily at a set hour. Four times a day the cron checks whether the configuration has been changed and (if needed) updates the action plan. If the configuration is set to null the job will not be executed (i.e. is inactive). You can find the cron configuration in Admin / Configuration.



The screenshot shows the 'Configuration' page in Zebrabase. A navigation bar at the top includes 'Fish', 'Facility', 'Calendar', 'Projects', 'Stats', and 'Admin'. Below the navigation bar, there are tabs for 'Dashboard', 'Configuration', 'Codebooks', 'Workgroups', 'Users', 'Permission groups', 'Notifications', 'Attachments', 'Newsletters', 'Tutorials', and 'Logs'. The 'Configuration' page is titled 'General configuration' and contains a table of configuration items. A red box highlights the rows for 'cron_jobs.make_productive' and 'cron_jobs.notify_age'. A warning message at the top of the configuration table states: 'Please note that this configuration is what makes the whole Zebrabase work. That's why you can change just some values and you should be really careful when changing them.'

Configuration Item	Value	Status
calendar.show_weekend		✓
calendar.week_start_sunday		✗
cron_jobs.make_productive	null	
cron_jobs.notify_age	01:10	
cron_jobs.retire	23:19	
custom_fields.user.custom_field_1	custom field 1	
custom_fields.user.custom_field_2	custom field 2	

3.13 Water systems

Extensions needed: water monitoring

Facility / Water system / Add watersystem

Water systems are used to monitor water quality in your facility. Each independent water circulation system should be defined separately and can possess one or more racks.

The screenshot shows the 'New watersystem' form in the ZebraBase application. The form is titled 'New watersystem' and has a 'Save' button in the top right corner. The form fields are as follows:

- Name ***: A text input field.
- Description ***: A text input field.
- temperature range**: Two text input fields.
- pH range**: Two text input fields.
- conductivity range**: Two text input fields.
- NH₃ range**: Two text input fields and a 'mg/l' unit dropdown.
- NO₂ range**: Two text input fields and a 'mg/l' unit dropdown.
- NO₃ range**: Two text input fields and a 'mg/l' unit dropdown.
- custom field 1 range**: Two text input fields and a 'mg/l' unit dropdown.
- Racks**: A dropdown menu with the placeholder text 'Start typing...'.

At the bottom of the form, there are three buttons: 'Save', 'Save and add another', and 'Back to list'.

The mandatory fields are Name and Description. The description can be longer and should precisely define the exact water system (e.g. location, type).

The predefined water quality parameters are temperature, pH, conductivity, NH₃, NO₂, and NO₃. You can also define up to 5 custom parameters using the [custom fields](#). For each parameter, you can define the safe range within which the values should be.

Finally, define the racks that are part of the water system.

Once defined, you can create new logs by going to the water system detail and clicking on Log water status.

Watersystem #2 - Fish facility I

[Delete](#) [Update](#) [Clone](#) [Back to list](#) [Log water status](#)

Name	Fish facility I
Description	mistnost 01.169
Temperature	30.0
Temperature range	26.0 - 29.0
pH	7.4
pH range	6.0 - 8.0
Conductivity	694.0
Conductivity range	300.0 - 1500.0
NH ₃ [mg/l]	0.5
NH ₃ [mg/l] range	0.0 - 0.1
NO ₂ [mg/l]	7.2
NO ₂ [mg/l] range	0.0 - 0.3
NO ₃ [mg/l]	692.0
NO ₃ [mg/l] range	0.0 - 25.0

Last data

[see full list](#)

Timestamp	Temp.	pH	Cond.
	NH ₃	NO ₂	NO ₃
2022-11-17 17:00 [UTC]	28.8	7.6	692.0
	0.41	134.89	13.3
	27.9	7.2	700.0
2022-11-24 13:30 [UTC]	0.05	0.063	46.2
	27.8	7.5	691.0
2022-11-30 13:45 [UTC]	0.0	0.095	52.3
	27.9	7.1	697.0
2022-12-08 13:30 [UTC]	0.0	0.099	40.5
	27.8	7.2	700.0
2022-12-15 09:30 [UTC]	0.115	0.012	19.0
	27.8	7.4	703.0

4. Api

4.1 Basics

Zebrabase is fully driven by REST API so you can do any action without using the web interface. API is accessible on the URL `<your domain>/api`, e.g. `demo.zebrabase.org/api`. The domain part of the URL is omitted in the examples.

All examples in this documentation are shown in the format

`<method> <url> [optional DATA: data]` or as a curl command (see next section).

4.1.1 Authentication

To communicate with API you have to obtain an access token first:

```
POST /api/auth/token DATA: {"username": "<username>", "password": "<password>"}
```

```
curl /api/auth/token -X POST -d '{"username": "<username>", "password": "<password>"}'
```

(If using Windows command prompt for curl commands, there may occur syntax problems with using single quotes (') in the statement. You can try to run the command as is in Git Bash, use the Microsoft WSL tool, or save the input data as an external JSON file and import it into the curl statement.)

Response: `{ "token": "<your access token>" }`

You have to send this token in the headers of any further request:

```
-H 'Authorization: Token <your token>', e.g.
```

```
curl /api/substock/536 -H 'Authorization: Token <your token>'. This header is mandatory for all API requests so it is omitted in all examples.
```

In the current version, the access token does not expire. Token expiration may be added in the future, therefore, please, consider implementing some fallback solutions in your scripts.

4.1.2 Security

Your access token should be kept secret. It's not advisable to put it in the code and upload it to a public repository.

If you want to build your own script, please create a special user for it (`user type: machine`) and make sure to grant it just the permissions it will need. Create a separate permission group if necessary.

4.1.3 Excel spreadsheet

You can use this [Excel spreadsheet](#) for accessing and downloading certain lists (substocks, fishlines, ...) from your database instance. It works based on APIs, i.e. you will need to input your credentials and generate a token to be able to access the data.

4.2 Get

updated: 2024-04-11 (v. 3.4.18)

Using the GET method, you can get lists of any listable classes, filter and order them, or look up an object with a specific ID.

Example:

`GET /api/<class>/?<parameters>`, e.g. `GET /api/user/?detail`

4.2.1 Listable classes

- `affectedgene` - genotype affected genes
- `attachment` - list of attachments
- `attachmenttype` - shortened list of attachments - ID + attachment type
- `background` - fishline background
- `config` - facility configurations
- `counttype` - count types
- `diet` - diets
- `driver` - gene drivers for genotypes
- `event` - events for notifications
- `experiment` - experiments
- `fishline` - fishlines
- `genotype` - genotypes
- `globaltarget` - notification targets
- `group` - permission groups
- `input` - action log
- `input_condition` - action log, type: Conditions
- `input_count` - action log, type: Count
- `input_crossing` - action log, type: Crossing
- `input_genotyping` - action log, type: Genotyping
- `input_productivity` - action log, type: Productivity
- `input_split` - action log, type: Split
- `input_transfer` - action log, type: Transfer
- `light` - light conditions
- `message` - your received messages
- `method` - count type methods
- `modcategory` - genotype modification categories

- `origin` - fishline origins
- `permission` - list of permissions - not permission groups
- `position` - list of positions
- `positionblocktype` - types of blocked positions
- `printer` - printers
- `printertype` - printer types
- `printingtemplate` - printing templates
- `printlog` - printing log
- `project` - projects
- `rack` - racks
- `request` - requests
- `room` - rooms
- `sex` - sexes
- `source` - fishline sources
- `species` - species
- `stock` - stocks
- `substock` - substocks
- `substockuse` - substock uses
- `triggeredgene` - triggered genes used in genotypes
- `user` - users
- `waterlog` - water system logs
- `watersystem` - water systems
- `workgroup` - workgroups
- `zygosity` - zygosity

4.2.2 GET parameters

All get parameters are optional.

Detail

Usually, when accessing one of the classes above, the result contains only basic information. If you need more detailed data (e.g. some M:N related objects expanded), you have to use parameter `detail`. But use it carefully - the response size and processing time may significantly increase.

Example:

```
GET /api/substock/?detail
```

Deleted objects

Deleted objects are excluded from all list requests by default. Parameter `all[=true]` includes deleted objects in the list.

Example:

```
GET /api/substock/?_all
```

Paging

If the returned list exceeds a length limit, it is split into multiple pages and without any additional parameters, only the first page of the list is returned. You can change the page size and go through individual pages using the parameters below. The default value for page size is in the instance configuration under `defaults.list.page_size`.

- `pg_size` - number of objects returned in one response, `pg_size=0` returns all objects - this can lead to request timeout, in general, it is not ideal to use it without filters
- `pg` - number of the requested page, counted from 1

Examples:

```
GET /api/substock/?pg_size=10&pg=3
```

```
GET /api/triggerredgene/?pg_size=0
```

Order

Lists can be ordered according to one or more attributes. The order can be set with a parameter `o=<comma-separated list of attributes>`. To sort data in descending order use `-` before the attribute.

Django lookups (double-underscore-separated path through the DB model) can be also used to sort, e.g. substocks can be sorted by `stock_fishline_background_name`.

Examples:

```
GET /api/substock/?o=-count,-stock_date_of_birth
```

```
GET /api/triggerredgene/?o=name
```

Filters

Every other parameter is considered to be a filter. If it is not a valid Django lookup ([Django docs](#)) the request will fail.

Look-up consists of a double-underscore-separated path through the DB model and an optional double-underscore-separated operator, e.g.:

- `/api/substock/?stock_date_of_birth_gte=2015-12-01` (`gte` is operator greater or equal),
- `/api/background/?name=ABC` (missing operator means exact match).

Please note that the filter conditions are applied in conjunction (`filter_1 AND filter_2 AND filter_3`) and there is no way to put `OR` between them. Yet some special use cases of OR are available: operator `in` and combined lookup (see further sections).

Operators

Here is a list of some commonly used operators (those with leading `i` are case insensitive):

- `exact`, `ixact` - exact match = equal
- `contains`, `icontains` - contains given substring
- `containsall`, `icontainsall` - contains all given substrings (comma-separated list)
- `like`, `ilike` - SQL like, allows using some wildcards
- `startswith`, `startswith`, `endswith`, `iendswith` - starts or ends with given substring
- `gt`, `gte` - greater than, greater than, or equal
- `lt`, `lte` - less than, less than or equal
- `in` - belongs to the given list of possible values
- `isnull` - the attribute is not set

Negative filters

Lookups ending with `!` are negative, e.g. `in!` means "not in", `exact!` means "≠".

Combined lookup

If you are looking for a sub-string (or whatever else) and you are not sure in what attribute should it be, you can search a few attributes at once with this feature. Just put comma-separated look-ups into square brackets:

```
[name_gen__icontains,description__icontains]=<something>
```

You can also mix operators:

```
[count_gte,description_icontains]=<something>
```

Or put the common operator out of the brackets (this is equivalent to the first example):

```
[name_gen,description]__icontains=<something>
```

Examples:

Square brackets have a special function in curl so you need to backslash them to work properly ("\[", "\]"). Similar issues may occur with other symbols such as "&".

GET /api/substock/?count__gt=10&id__in=12,13,15,16 - count > 10 and ID belongs to the list

GET /api/substock/?id__in!=12,13,15,16&status__in!=dead,merged,split out&count=10

GET /api/substock/?[id_exact,stock_id_exact,stock_local_stock_id_exact,name_gen_icontains,po

- this is a substock quick search filter GET /api/fishline/?genotypes__isnull - wild type

fishlines GET /api/substock/?stock_fishline_genotypes__isnull - wild type substocks

GET /api/substock/?name_gen__icontainsall=abc,B52,E55 - substocks name_gen has to contain all of the given substrings

4.2.3 Response

The response contains 3 sections:

- `data`
- `meta` - metadata
 - `filters` - filters used to get this list, split into 2 sections:
 - `positive` - e.g. `count=10`
 - `negative` - e.g. `id__in!=1,3,5`
 - `order`
 - `filtered_count` - number of objects matching used filter
 - `total_count` - total number of objects in DB (not including deleted unless `_all` is used)
 - `page`
 - `page_size`
 - `max_page` - maximal valid value for `page`
- `warnings` - list of warning messages

Example 1

```
GET /api/sex/
```

```
{ "data": [ { "id": 1, "name": "unknown", "tag": null, "active": true }, { "id": 2, "name": "male", "tag": "<i class=\"fa-fw fas" }
```

Example 2 - Substock simple

```
GET /api/substock/?pg_size=1
```

```
{ "data": [ { "id": 16950, "count": 45, "substock_num": 12, "name_gen": "ubi:GFP<sup>+</sup> 2023-12-13/1 (12)", "facility_name_
```

Example 3 - Substock detailed

```
GET /api/substock/?pg_size=1&detail
```

```
{ "data": [ { "id": 16950, "count": 45, "substock_num": 12, "name_gen": "ubi:GFP<sup>+</sup> 2023-12-13/1 (12)", "facility_name_
```


4.3 Create & Update

4.3.1 Describe - get a list of mandatory and optional parameters

```
GET /api/<class>/describe/
```

This request returns a generated list of mandatory and optional parameters for record creation and updating, as well as a list of permissions needed for any action. If a specific action is not listed, it cannot be used manually (e.g. create `stock`, etc.).

Please note that lists of mandatory and optional parameters are generated from the inner code structure and there can be some additional rules that may not be visible here, for example, when creating a fishline you have to set either `alias` or `genotypes`, so technically there are no mandatory parameters. However, if you omit both fields, the request will fail.

If you need any help or if you find a bug (e.g. `describe` that is uselessly inaccurate), don't hesitate to [ask for help](#).

Response

Example 1 - Fishline

```
{ "model": { "class": "FishLine", "table": "fishline" }, "create": { "mandatory": [], "optional": [ "attachments", "projects", "
```

Example 2 - Project

```
{ "model": { "class": "Project", "table": "project" }, "create": { "mandatory": [ "name", "leader_id", "code" ], "optional": [ "
```

4.3.2 Create

```
POST /api/<class>/ + DATA
```

Creation returns the detail of the created object if successful.

The mandatory and optional data for each class are listed in the response for `/api/<class>/describe/`. If not satisfied, the response will be an error message stating which parameters are bad or missing.

Example

One missing mandatory parameter and one excessive parameter:

```
POST /api/fishline/ DATA: {"responsible_user_id": 1000, "something": "nothing"}
```

```
Curl: curl -d '{"responsible_user_id": 1000, "something": "nothing"}' -X POST /api/<class>/
```

```
{ "detail": "At least one of these params must be specified: 'alias', 'genotypes'", "info": {}, "status_code": 400, "exception_c
```

Let's add the mandatory parameter:

```
POST /api/fishline/ DATA: {"alias": "test", "responsible_user_id": 1000, "something": "nothing"}
```

```
{ "detail": "These parameters are not necessary: {'something'}", "info": {}, "status_code": 400, "exception_class": "Unnecessary
```

Non-existent value:

```
POST /api/fishline/ DATA: {"alias": "test", "responsible_user_id": 1009}
```

```
{ "detail": "Request lead to database integrity error: insert or update on table \"fishline\" violates foreign key constraint \"
```

Correct command:

```
POST /api/fishline/ DATA: {"alias": "test", "responsible_user_id": 1000}
```

```
{ "id": 428, "name": "test", "name_gen": null, "name_base": null, "alias": "test", "in_facility_since": null, "description": nul
```

4.3.3 Multicreate

```
POST /api/<class>/multicreate/ + DATA
```

This feature allows the creation of multiple instances with one request. The main difference between this solution and just repeating the create request is transaction safety: if any creation from multicreate fails, the whole batch is rolled back.

Data has to be a list of data for each instance, e.g.:

```
POST /api/affectedgene/multicreate/ DATA: [{"name": "A"}, {"name": "B"}, {"name": "C"}]
```

The response is a list of new objects:

```
[ { "id": 4, "name": "A", "active": true }, { "id": 5, "name": "B", "active": true }, { "id": 6, "name": "C", "active": true } ]
```

4.3.4 Update

`PUT /api/<class>/<object id>/ + DATA`

Update returns the detail of the created object if successful

Parameters available for the update are listed in `/api/<class>/describe/`.

4.4 Delete

```
DELETE /api/<class>/<object id>/[?cascade]
```

Deletion of an object corresponds to setting the attribute `active` to `false`, in almost all classes. These deleted objects can be shown in a list (see chapter [Get - deleted objects](#)), in detail and most of them can be re-enabled (there are exceptions, e.g. inputs).

The parameter `cascade` cascades the deletion to child objects. Works just for these classes:

```
affectedgene / triggeredgene / driver → genotype → fishline → stock → substock
```

4.4.1 Re-enable (undelete)

```
PUT /api/<class>/<object id>/enable/
```

This function is not available for any `input`.

4.4.2 Exceptions

- These classes are physically deleted from the database:
 - `position`
 - `rack`
 - `attachment`
- These classes can't be deleted:
 - `message`
 - `input` - types `update_*` and `request_*`
- These classes can't be re-enabled:
 - `input`

4.5 Substock actions

updated: 2024-04-11 (v. 3.4.18)

General parameters

- `performed` - real date of the action, format: `"YYYY-mm-dd"`
- `description` - user's note related to the action

These parameters are common for all substock actions. Below, you can find a full list of parameters for each action type. **Parameters in square brackets are optional.**

4.5.1 Count

```
PUT /api/substock/<substock_id>/count/ + DATA
```

Schema

```
[count: int], [all: bool], counttype_id: int, [experiment_id: int], [method_id: int], performed: str, [description: str]
```

- Either `count` or `all` must be set
- `all` works only for counttypes of type `sub` (subtractive) (GUI function "terminate" uses `all=true`)
- `counttype` setting `public` has to be set to `true` in order to use the count type
- `experiment` and `method` are mandatory depending on `counttype` settings `bound_to_experiment`, respectively `bound_to_method`

Example data

Terminate substock

```
{ "all": true, "counttype_id": 3, "performed": "2021-02-12" }
```

3 fish died

```
{ "count": 3, "counttype_id": 2, "performed": "2021-02-12" }
```

4.5.2 Transfer

```
PUT /api/substock/<substock_id>/transfer/ + DATA
```

Schema

```
position_id: null or int, [switch: bool], [shuffle: bool], [remove_colliding: bool], [target_width: int 1-4], [source_width: int
```

- `position_id = null` - substock will be removed from its position in the facility
- `switch`, `shuffle` and `remove_colliding` specifies what to do if the target position is already occupied; only one can be set true at the same time
- `target_width` and `source_width` allow to set the width of the target, resp. source tank

Example data

Switch position with substocks on target position

```
{ "position_id": 156, "switch": true, "performed": "2021-02-12" }
```

4.5.3 Genotyping

```
PUT /api/substock/<substock_id>/genotyping/ + DATA
```

Schema

```
genotyping_data: list( genotype_id: int, zygotity_id: int ), performed: str, [description: str]
```

- You don't have to set zygosities for all substock's genotypes

Example data

Set zygotity for 2 genotypes

```
{ "genotyping_data": [ { "genotype_id": 1, "zygotity_id": 3 }, { "genotype_id": 2, "zygotity_id": 2 } ], "performed": "2021-02-12" }
```

4.5.4 Productivity

```
PUT /api/substock/<substock_id>/productivity/ + DATA
```

Schema

```
productivity: str, performed: str, [description: str]
```

- productivity values: `unproductive`, `productive`, `retired` (needs special permission `substock_retired`)

Example data

Set productivity to productive

```
{ "productivity": "productive", "performed": "2021-02-12" }
```

4.5.5 Merge

```
PUT /api/substock/<substock_id>/merge/ + DATA
```

Schema

```
substocks: list(int), performed: str, [description: str]
```

The substock specified in the curl call will be the target substock for the merge (the other substocks will be merged into it). All substocks must belong to the same stock.

Example data

Merge specified siblings into a substock

```
{ "substocks": [125, 126, 127], "performed": "2021-02-12" }
```

4.5.6 Conditions

```
PUT /api/substock/<substock_id>/conditions/ + DATA
```

Schema

```
[light_id: int], [diet_id: int], performed: str, [description: str]
```

- At least one of `light_id`, `diet_id` must be set

Example data

Set all

```
{ "diet_id": 2, "light_id": 1, "performed": "2021-02-12" }
```

4.5.7 Split

```
POST /api/substock/<substock_id>/split/ + DATA
```

Split has 2 variants, here are schemas for each of them

Schema 1 - simple

```
splits: int, [owners: list(int)], [projects: list(int)], [workgroups: list(int)], performed: str, [description: str]
```

- Note that split uses the POST method as opposed to other substock actions
- `splits` - number of new substocks with an unspecified count to split off (count value will be `not set`)
- `owners`, `projects`, `workgroups` - parameters to be set to the new substocks

Schema 2 - detailed

```
split_data: list( count: int, [terminate: bool], [position_id: null or int], [tank_width: int 1-4], [productivity: str], [productivity_detail_id: int])
```

- `split_data` - list of specifications of the new substocks
 - `count` - fish count
 - `terminate` - should the substock be terminated just after the split?
 - `position_id`, `tank_width` - parameters for automatic transfer
 - `productivity`, `productivity_detail_id` - parameters for automatic productivity action
 - `genotyping_data` - parameters for automatic genotyping
 - `owners`, `projects`, `workgroups` - parameters to be set to the new substock; override the same parameters on the top level for this specific substock
- `owners`, `projects`, `workgroups` - parameters to be set to the newly created substocks (can be overridden for each substock)

Example data

Simple split

```
{ "splits": 2, "performed": "2021-02-12" }
```

Detailed split

```
{ "split_data": [ { "count": 3, "owners": [5], "suffix": "xyz" }, { "count": 5, "terminate": true }, { "count": 4, "projects": [1, 2] } ] }
```

4.5.8 Crossing

```
POST /api/substock/crossing/ + DATA
```

Crossing is a very complex action, especially choosing the fishline for the progeny substock. Using the GUI interface is recommended in this case.

Available fishlines

```
GET /api/fishline/by_genotypes/?genotypes=<comma-separated list of genotype_id>
```


This function returns 3 lists of fishlines available for progeny substock when crossing substocks with genotypes (parameter `genotypes`):

- The exact match category contains fishlines with all genotypes of all parents.
- The partial match category contains fishlines with a subset of genotypes of the parents.
- The wild-type category contains fishlines without genotypes.

Schema

```
parents: list( id: int, [sex_id: int] ), [count: int], [owners: list(int)], [projects: list(int)], [fishline_id: int], [force_new_fishline: bool]
```

- Note that split uses the POST method as opposed to other substock actions
- `parents` - you need at least 2 parents for crossing, set the same parent twice for incross
 - `id` - id of the parent-to-be substock
- `count`, `owners`, `projects`, `fishline_id` - optional parameters for the progeny substock
- `fishline_id` - not every fishline can be set, see crossing documentation
- `fishline_data` - parameters for the new fishline, `force_new_fishline` must be set to `true`

Example data

Two parents (automatic fishline)

```
{ "parents": [ { "id": 14349, "sex_id": 3 }, { "id": 17221, "sex_id": 2 } ], "owners": [1008], "performed": "2024-04-12" }
```

4.6 Accessing and downloading backups

You can download full backups of the database, or use this [Excel spreadsheet](#) to download certain data tables from Zebrabase. Zebrabase produces backups every 12 hours; at 12 AM and 12 PM UTC. To download the backups, it is necessary to obtain the permission `download_backup`. It is possible to access and download the backups by following these steps:

1. step

A **new user account** (type = machine) should be created for this action. Please, do not use your own credentials in order to keep your password protected.

2. step

The **access token** for Zebrabase API can be obtained by following the instructions in the manual: [API/Basics](#). This action is straightforward when using Linux. For occasional access, a REST client or similar can be used.

3. step

A **list of all available backups** (JSON format) can be displayed by using:

```
curl https://<prefix>.zebrabase.org/backup/ -L -H 'Authorization: Token <access-token>'
```

```
**example response:**
```

```
[ { "name": "zb3_<prefix>-2021-09-22--13:40:10.tar.gz", "type": "file", "mtime": "Wed, 22 Sep 2021 11:40:11 GMT", "size": 258300 },
```

4. step

In the last step, a **file** can be selected and **downloaded** using:

```
curl https://<prefix>.zebrabase.org/backup/zb3_<prefix>-2021-09-22--13:40:10.tar.gz -L -H 'Authorization: Token <access-token>'
```


5. Newsletters

5.1 3.1.9: Changes in action inputs

5.1.1 Fish count before every action

- every action has now information on how many fish were in substock when the action has begun (= how many fish were involved in this action)
- this number can be found as `count before` in the input list and on the substock detail page (in the input sidebar)
- it's working now for all actions **after this update**, older counts will be computed later; we will notify you when it's finished

update ↗	2021-08-09	i
split [2] ↗	2021-08-18	i
👑 count ↗	2021-08-18	i
transfer ↗	2021-08-18	i

user: root
 count before: 2 
 note:
 transfer: void --> B01:C03

5.1.2 GUI action form

- the form is split into two tabs: `Performed date & identifiers` and `Form`
- whenever the performed date or the substock list changes, substock availability is refreshed - **that can take some time if you are doing batch action on the maximum count of 500 items**
- in performed date picker you can see some dates colored orange - these dates are available just for a subset of selected substocks; after picking an "orange" date substocks in the list are highlighted green or red, depending on their availability; the substock list can be filtered
- **substocks that are not available for the action on selected date don't have to be removed from the list manually** - they will be skipped during the action execution

Count - batch

Performed date & identifiers

Form

Performed * ⓘ date of execution

2021-08-10|

August 2021

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

4]

/1 (11) ⚡

/1 (9) ⚡

(ozmik) 2021-08-09/1

4608-10190 casper (DR) 2021-06-01/1 (1)

4603-10167 mutant^{iy57+} 2021-05-08/1 (1)

5.1.3 Input order can be set implicitly

- available from API and GUI if registering an action for a single substock
- when saving an action for a substock which has another actions already registered for the same date user can select the precise order of the actions; **default is last**
- there are, of course, some restrictions to keep the data consistent:
 - no action (except crossing) can be performed before another action of the same type
 - `split`, `merge`, `count` blocks each other

Productivity ✕

Performed date & identifiers [Form](#)

Performed [*] [🔔] <small>date of execution</small>	Note
<input type="text" value="2021-08-18"/>	<input type="text"/>

This item already has some more inputs for selected date - choose the exact timing

-- last --

before COUNT by jana

before TRANSFER by root

before COUNT by root

-- last --

Save action

5.2 3.1.10: Reducing number of projects assigned to substock

What's going on?

We are preparing a new system of projects and experiments, and therefore **we need you to reduce the number of projects assigned to a substock to just one**. Then Zebrabase will be able to gather information about projects' responsibilities among dead fish.

Is it your problem?

This action is needed just for some facilities. If a yellow ribbon with a warning message appeared on the top of Zebrabase webpage, your facility is among them.

How to do it

Just update all substocks with more than one project and remove some project assignments. You can also batch-update them (batch-update just substocks with the same desired project).

In case you need any assistance, please contact us at support@zebrabase.org or via [helpdesk](#).

Deadline

Complete the reduction by **October 31, 2021**, please.

The substock-project relation will change during the first November upgrade. If there are still substocks with more than one project at that time, only the last one will be retained as a substock's project. (So if you don't have to fix it twice if you are using testing instance.)

5.3 3.3.0: Zebrabase News (May/2022)

New: The look

This time we focused on the interface and improved the layout, color-scheme, and buttons functionality. You will notice a lot of changes that should improve your experience when using Zebrabase.

- Colors: the color scheme is now more unified and the color palette is harmonized, blue is the color for all actions and saving, red for warnings, and deleting
- Buttons: instead of words self-explanatory unified pictographs are used everywhere possible
- Layout: filtering setup is now hidden in a single icon, the filtering form, and substock action forms are reorganized for easier navigation

New: Custom filters

It is now possible to create and save a number of various filtering settings of your choice, label them, and share them with your colleagues.

New: Tags

Instead of the feature Productivity detail, which is no longer supported, additional information about your substocks can be stored as a tag (sick fish, single fish...). The tag is simple, colorful and visible in the facility view just like a paper tag would be on a real tank. It is recommended to limit the length of each tag to one word only. (Note: Productivity status can now longer be used for this kind of information and will be reserved for information on fish productivity only.)

New: Batch action permission

In order to perform certain batch actions, the permission batcher is now required. This will prevent non-authorized actions on a large number of substocks by inexperienced users. The permission can be granted to a specific user group by admin in Admin/Permission groups/Update. For admins, this permission is a default.

New: Batch or marked list export

Exporting of the manually selected content is enabled by using marked list export or batch actions.

New: Generation number

The stocks can now have a generation number (F0, F1...). The generation is set to 0 when creating a new fishline. In-crossing will increase the generation number by one. In other cases, it can be set manually. The generation number can be updated in Fish/Update Stocks.

New: Automatic label printing update

A change in substock properties will reset the label printing status to: To print and the printer icon will turn orange.

New: ZFIN nomenclature

ZFIN nomenclature is now supported to promote a clear and correct way of naming fishlines (example: Tg(gata1:DsRed)). This feature is enabled on request, please, contact us if you are interested.

5.4 3.4.0: Zebrabase News (October/2022)

New: Multi-species functionality (Premium feature)

Zebrabase is now suitable for multiple species of fish or any invertebrates that are kept in tanks!



The species can be marked with a code or an abbreviation and the life-stage will be updated automatically according to the species-specific parameters.

Read more in the manual:

https://docs.zebrabase.org/admin/10_multispecies/

New: Tutorials

From now on, it is possible to train new users by using interactive tutorials. Our library of tutorials will be continuously developed.

New: License ID field

The user's license ID is a new optional parameter to be defined in the User's details form. Very useful in any country, where users are required to obtain a license for animal handling!

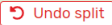
New: Funding source field

A new optional parameter intended for grant codes or names of the grant recipients can be defined for each Fishline.

New: Undo split

The split action can be easily reversed by clicking on the Undo split button in the action-success confirmation message.



Substock split successful: 1 new substock(-s) 

New: Batch count update

The count action can be now performed on batches of substocks even if they are in various projects or if they are burdened. The new action form has an improved user-friendly lay-out.

Read more in the manual:

https://docs.zebrabase.org/admin/11Batch_actions/