trapper-project Documentation

Release 1.1.0

Open Science Conservation Fund (OSCF)

Jun 15, 2023

Contents:

CHAPTER 1

Overview

Trapper is an open source, django based web application to manage camera trapping projects. Motion-triggered camera trapping is increasingly becoming an important tool in ecological research. Because of the nature of collected data (multimedia files) even relatively small camera-trapping projects can generate large and complex datasets. The organization of these large collections of multimedia files and efficient querying for a particular subsets of data, especially in a spatio-temporal context, is often a challenging task. Without an appropriate software solution this can become a serious data management problem, leading to delays and inaccessibility of data in the long run. We propose a new approach which, in contrary to available software solutions, is a fully open-source web application using spatially enabled data that can handle arbitrary media types (both pictures and videos), supports collaborative work on a project and data sharing between system users. We used state of the art and well-recognized open-source software components and modern, general purpose programming language Python to design a flexible software framework for data management in camera trapping studies.

1.1 TRAPPER in a nuthsell

- it is open-source
- provides a spatially-enabled database backend
- can handle both pictures & videos
- provides a flexible model of classifications
- promotes data re-use
- supports collaborative work on a project
- provides API

1.2 Demo

https://demo.trapper-project.org

1.3 Read more

Bubnicki, J. W., Churski, M. and Kuijper, D. P. (2016), TRAPPER: an open source web-based application to manage camera trapping projects. Methods Ecol Evol, 7: 1209-1216. doi:10.1111/2041-210X.12571

https://besjournals.onlinelibrary.wiley.com/doi/10.1111/2041-210X.12571

For more news about TRAPPER please visit the Open Science Conservation Fund (OSCF) website:

https://os-conservation.org

CHAPTER 2

Installation

TRAPPER is provided as Docker container and can be deployed on both Linux- and Windows-based servers. However, we recommend a Linux distribution based on Debian e.g. Ubuntu Server. TRAPPER uses Docker composer, a tool for defining and running multi-container Docker applications. With this technology all basic services needed to run TRAPPER (including database and FTP server) can be automatically deployed and configured on a dedicated server with a possibility left open to configure particular services manually (e.g. a scenario with an external database server).

2.1 Installing Docker Community Edition (CE) & Docker Compose

1) Follow the steps from the official Docker documentation:
   - Ubuntu: https://docs.docker.com/install/linux/docker-ce/ubuntu/
   - Windows: https://hub.docker.com/editions/community/docker-ce-desktop-windows
2) Install Docker Compose: https://docs.docker.com/compose/

Note: To learn more about Docker technology including the management of Docker images and containers please read the official Docker documentation.

2.2 Getting the source code of TRAPPER

1) First, clone the source code of TRAPPER to your local repository:

```
$ git clone https://gitlab.com/trapper-project/trapper.git
```

2) To get the most up-to-date version of TRAPPER switch to a branch `development`:

```
$ git checkout development
```

Note: It is much easier to use Git Bash for cloning git repositories on Windows instead of using a standard Windows command line: https://git-scm.com/download/win

2.3 Running TRAPPER

Note: If you need a custom configuration of TRAPPER (e.g. external database, ftp and email services) please check the *Configuration* section for details.

1) Copy the `trapper.env` file to the `.env` file, you can do it using the command:

   ```
   $ cp trapper.env .env
   ```

2) Adjust the variables in the `.env` file if you use a non-standard installation, e.g. the external postgresql database.
3) Run TRAPPER with the following command:

   - Production

     ```
     $ ./start.sh -pb prod
     ```

   - Development

     ```
     $ ./start.sh -pb dev
     ```

Note: Flags `-p` and `-b` tell the script to `pull` all required docker images and to `build` the `trapper` image locally. To learn more about all possible running options simply type `./start.sh`. For example, to run TRAPPER with an external postgresql database you can use flag `-d` (but first check the *Using an external database* section).

Warning: The process of pulling, building & starting all docker containers can take a while!

4) Now your TRAPPER instance should be up!

2.4 Creating the admin (superuser) account

Enter the `trapper` docker container and create the `superuser`:

```
$ docker exec -it trapper bash
$ python /app/trapper/trapper-project/manage.py createsuperuser
```

After providing a username, email and password you should be able to login to TRAPPER.

2.5 SSL certificates

If you already have the SSL certificates for your TRAPPER website you can use them by providing proper paths in the *.env* configuration file. See the *SSL certificates* section for details. These certificates will be used to configure the

HTTPS `nginx` proxy-server within a dedicated container. When there is no SSL certificates provided they will be generated automatically using the openssl toolkit.

Note: For example, you can get free SSL certificates from Let's Encrypt.

CHAPTER 3

Configuration

You can override many default settings in TRAPPER by adjusting the environmental variables listed in the file *.env*. Further in this section we provide some examples of how to customize selected aspects of TRAPPER.

3.1 Basic settings

Provide a name of your TRAPPER instance domain:

```
DOMAIN_NAME=demo.trapper-project.org
```

The `SECRET_KEY` is used to provide cryptographic signing, and should be set to a unique, unpredictable value (see django docs):

```
SECRET_KEY='cudv86xkzva)o1lkt-ow72)y__*xwsb1@43wtp1cc@zak=i@@e'
```

Define users that are admins of your TRAPPER instance. If the *Email service* is properly configured they will be notifying about all server errors (500), new users registrations and requests for approving new research projects.

```
ADMIN1=admin1,admin1@oscf.org # {username},{email}
ADMIN2=admin2,admin2@oscf.org
```

HTTP & HTTPS ports mapping definition is used to configure the trapper-nginx docker container (see nginx docker docs):

```
HTTP_PORTS=80:80
HTTPS_PORTS=443:443
```

3.2 SSL certificates

If you already have the SSL certificates for your TRAPPER website you can use them by setting the following variables:

```
# default configuration
SSL_CERTIFICATE=${PWD}/ssl/cert.pem
SSL_CERTIFICATE_KEY=${PWD}/ssl/key.pem
```

The example with the Let's Encrypt certificates:

```
SSL_CERTIFICATE=/etc/letsencrypt/live/trapper/fullchain.pem
SSL_CERTIFICATE_KEY=/etc/letsencrypt/live/trapper/privkey.pem
```

These certificates will be used to configure the HTTPS nginx proxy-server within a dedicated container (see trapper-nginx) and to secure the FTP connections (TLS) to the pure-ftpd docker container.

Note: When there is no SSL certificates provided they will be generated automatically using the openssl toolkit (self-signed certificates).

3.3 Using an external database

TRAPPER comes with a pre-configured postgresql docker image (see trapper-posgresql) and has been proved to work well with the recent postgresql-12 and postgis-3.0. The basic credentials needed to access this dockerized postgresql database volume can be configured with these parameters:

```
POSTGRES_USER=trapper
POSTGRES_PASSWORD=trapper
POSTGRES_DB=trapper
```

However, for a production-grade TRAPPER instance we recommend to configure an external postgresql database either running on the same virtual machine or on different remote server. In the *.env* file there are the following variables that can be used to configure a connection with an external database:

```
DB_NAME=trapper
DB_USER=trapper
DB_PASS=trapper
DB_HOST=172.17.0.1
DB_PORT=5432
```

Note: The host address `172.17.0.1` is for the case when the postgresql database is running on the same machine as TRAPPER docker container. This host ip should work for most of the standard Linux distributions, including the recent Ubuntu Server LTS 20.04.

Now you can run TRAPPER using the script *start.sh* and the extra flag *-d*:

```
$ ./start.sh -pbd prod
```

Note: You will still need to properly configure your external postgresql server which is out of the scope of this documentation. Here we only give you a couple of quick cues to help you make the connection between TRAPPER and postgresql working as expected (but be aware that likely you need more restrictive configuration!):

1) In the file `/etc/postgresql/12/main/postgresql.conf` update the following line:

```
listen_addresses = '*'
```

2) In the file /etc/postgresql/12/main/pg_hba.conf add the following entry:

```
# TYPE   DATABASE   USER   ADDRESS       METHOD
host     all        all    172.0.0.0/8   md5
```

172.0.0.0/8 matches all ip addresses starting with 172.X.X.X

3.4 Storage

The multimedia files uploaded to TRAPPER (i.e. camera trapping images & videos) can be stored either locally, i.e. using a storage infrastructure available at a virtual machine running TRAPPER docker container, or remotely in a cloud storage (see the subsection *Cloud storage* for details about supported cloud storage providers).

To use a local storage (default option) leave the following parameter empty:

```
DEFAULT_FILE_STORAGE=
```

Optionally, you can provide paths to local directories which will be used to mount the corresponding docker volumes:

```
VOL_MEDIA=/storage/trapper_data/media
VOL_EXTERNAL_MEDIA=/storage/trapper_data/external_media
```

Note: In a directory mounted as the volume VOL_MEDIA all images & videos uploaded to TRAPPER are stored in the following format:

```
# original files
protected/storage/resources/{user_id}/{date_uploaded}/

# preview files (for images)
protected/storage/resources/{user_id}/{date_uploaded}/previews/

# thumbnails (for both images & videos)
protected/storage/resources/{user_id}/{date_uploaded}/thumbnails/
```

The volume VOL_EXTERNAL_MEDIA provides a storage for users' data including FTP data (e.g. users' uploaded .zip files with multimedia collections) and all data packages exported from TRAPPER.

3.4.1 Cloud storage

The TRAPPER support for a cloud storage is based on a wonderful django package django-storages. Here is the list of cloud storage providers currently supported by TRAPPER and their corresponding configuration parameters:

1) Azure

```
DEFAULT_FILE_STORAGE=azure

AZURE_ACCOUNT_NAME=trapper
AZURE_ACCOUNT_KEY=key
AZURE_CONTAINER=trapper_data
```

(continues on next page)

(continued from previous page)

```
AZURE_CUSTOM_DOMAIN=trapper.blob.core.windows.net
MEDIA_LOCATION=media
```

2) Amazon S3

AWS_ACCESS_KEY_ID and *AWS_SECRET_ACCESS_KEY* - credentials for authorizing access to S3 storage.

AWS_STORAGE_BUCKET_NAME - S3 bucket name for storing all media files

AWS_S3_REGION_NAME - Name of AWS S3 region to use. Optional.

AWS_S3_ENDPOINT_URL - S3 Endpoint, required in order to use alternative S3-compatible storages such as Minio or Backblaze B2

Setting default file storage as *amazon_s3* is required in order to use S3 integration in TRAPPER

```
DEFAULT_FILE_STORAGE=amazon_s3
```

3.5 Email service

The working email service is required to make the notification framework working correctly. This feature is turned off by default. To configure the email notification backend you should set the following parameters (example with gmail):

```
EMAIL_NOTIFICATIONS=True
EMAIL_NOTIFICATIONS_RESEARCH_PROJECT=True
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_HOST_USER=project@gmail.com
EMAIL_HOST_PASSWORD=password
EMAIL_USE_TLS=True
```

Note: If local SMTP server is used it has to be configured to be able to send email messages. Configuration of the SMTP server is out of the scope of this documentation.

3.6 Users registration

Here you can define if users registration (signing-up) is open or not and optionally secure a user registration form with Google's recaptcha authentication.

```
USER_REGISTRATION_OPEN=True

# Google recaptcha settings
USE_RECAPTCHA=True
RECAPTCHA_PUBLIC_KEY=public_key
RECAPTCHA_PRIVATE_KEY=private_key
```

3.7 Django Debug Toolbar

If you are a developer of TRAPPER you can turn on a very cool tool for debugging called Django Debug Toolbar:

```
USE_DEBUG_TOOLBAR=True
DEBUG_TOOLBAR_USERS=admin1,admin2
```

3.8 Advanced system-level variables

These variables are set automatically when starting TRAPPER with the script `start.sh`. You can override them, but please keep defaults when you are not really sure what you can change here.

1) Set UID & GID for a user that will be used to run TRAPPER Gunicorn server within a docker container (by default these values are taken from a user starting docker containers using the `start.sh` script). The example custom configuration:

   ```
   TRAPPER_USER_UID=1000
   TRAPPER_USER_GID=1000
   ```

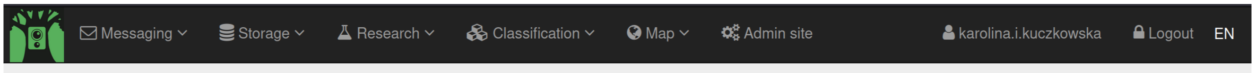
2) Set GID for a docker group which will be created within a TRAPPER docker container. The example custom configuration:

   ```
   DOCKER_GID=131
   ```

CHAPTER 4

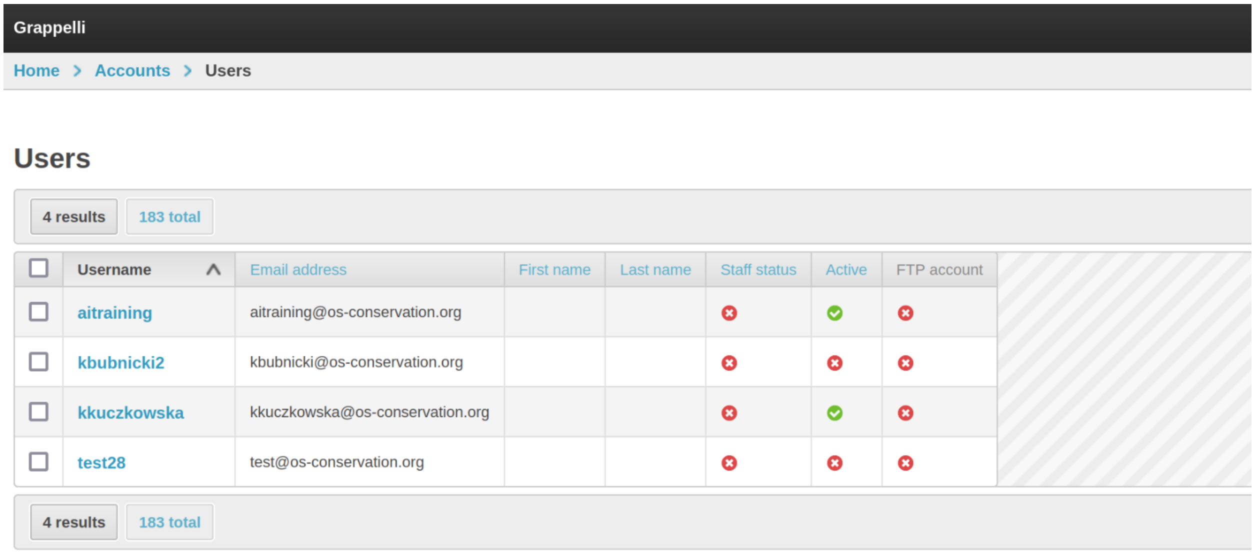
Administration

The Admin site is available in the top menu and it is only visible for users with admin permissions.



4.1 Admin actions for User model

Registered users will be listed on the Admin site. They need to be activated in order to log in, and will need project roles to be able to work on projects. All that and more can be achieved using admin actions, listed at the bottom left corner of User changelist admin view. Before running any action, select *Users* by marking checkboxes by usernames.

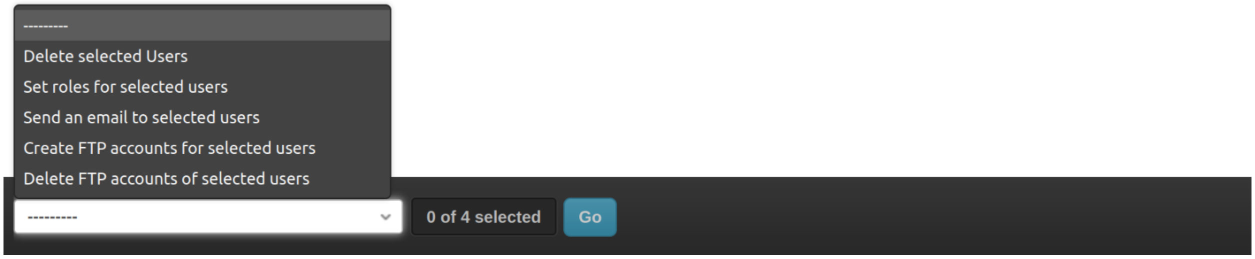


Users

4 results 183 total

□	Username	Email address	First name	Last name	Staff status	Active	FTP account
□	aitraining	aitraining@os-conservation.org			✗	✓	✗
□	kbubnicki2	kbubnicki@os-conservation.org			✗	✗	✗
□	kkuczkowska	kkuczkowska@os-conservation.org			✗	✓	✗
□	test28	test@os-conservation.org			✗	✗	✗

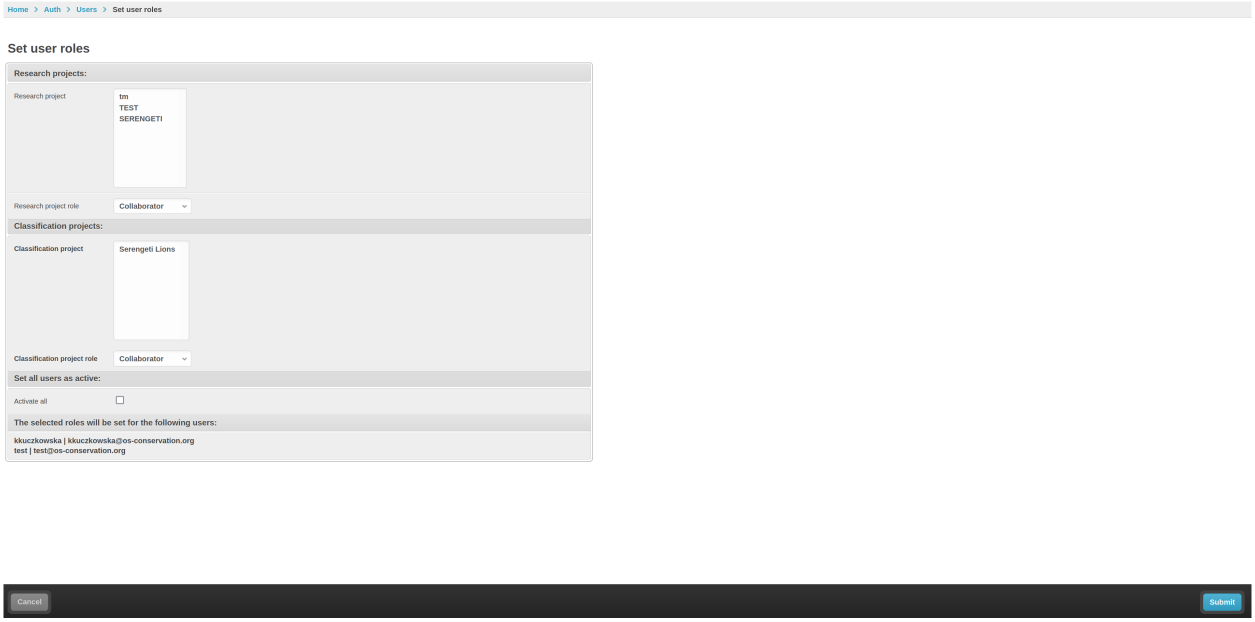
4 results 183 total



Delete selected Users
Set roles for selected users
Send an email to selected users
Create FTP accounts for selected users
Delete FTP accounts of selected users
----
0 of 4 selected
Go

4.1.1 Set roles for selected users action

After selecting *Users*, action and clicking the blue *Go* button next to the action select, you will access the form.



Set user roles

Research projects:

Research project: tm, TEST, SERENGETI

Research project role: Collaborator

Classification projects:

Classification project: Serengeti Lions

Classification project role: Collaborator

Set all users as active:

Activate all ☐

The selected roles will be set for the following users:

kkuczkowska | kkuczkowska@os-conservation.org
test | test@os-conservation.org

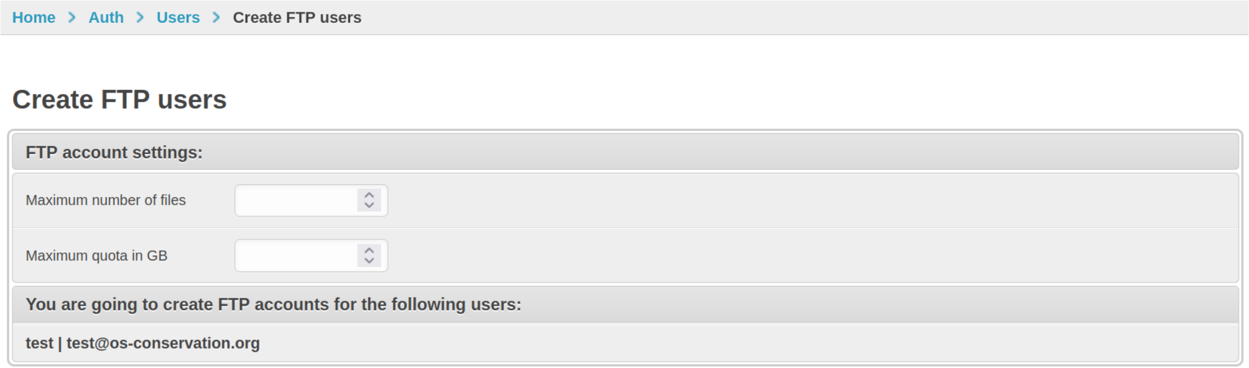
Here you can specify to which *Research projects* and *Classification projects* to add *Users*, and as which roles. By marking the checkmark, you can also activate all selected users.

After filling the form, hit the *Submit* button in the bottom right corner.

4.1.2 Create/delete FTP accounts

Users will need FTP accounts in order to upload data to the Trapper server. FTP accounts for *Users* can be created using admin actions.

After selecting *Users* and running the action, you will see a short form.



Create FTP users

FTP account settings:

Maximum number of files

Maximum quota in GB

You are going to create FTP accounts for the following users:

test | test@os-conservation.org

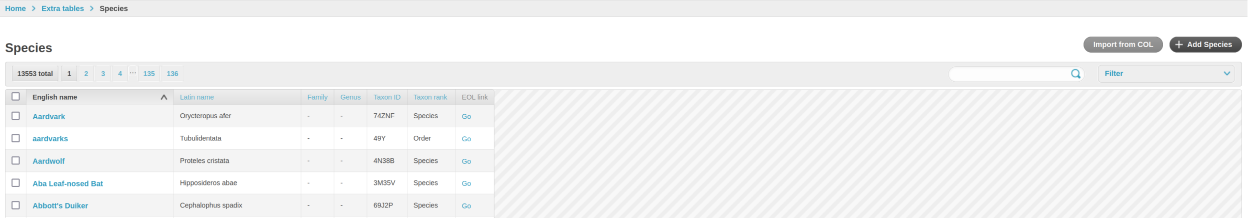
Fill the form and submit.

Creating FTP accounts both allows *Users* to log in to FTP server using Trapper account credentials, and creates the necessary directory structure in your external media directory.

4.2 Download Species from Catalogue of Life

Species table needs to be filled by the admin, so that selected *Species* can be added to *Classificators*.

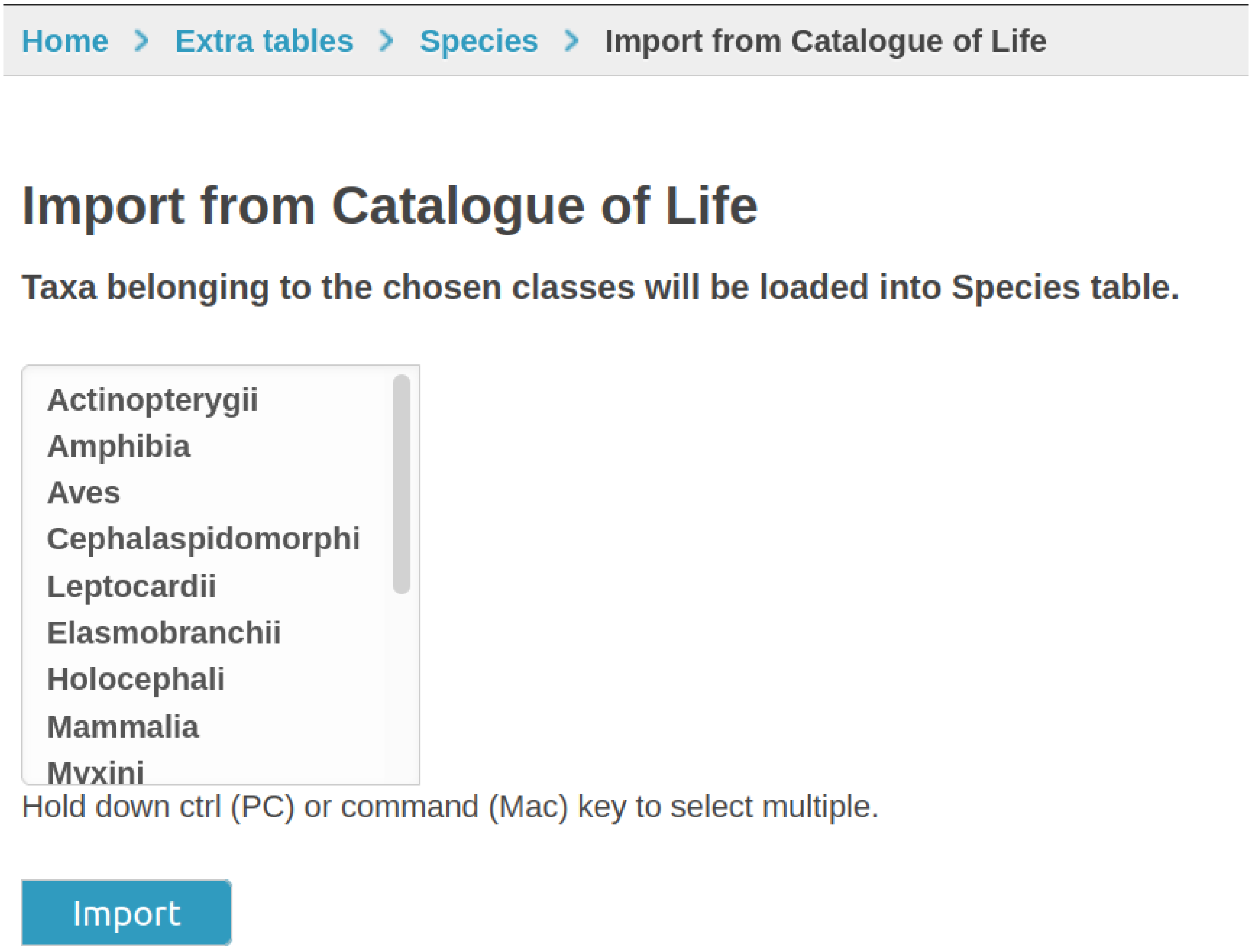
You can add them manually one species at a time, use a csv file to import them, or you can run an admin action to download taxa for chosen classes of animals from Catalogue of Life. The action is available in the *Species* changelist admin view (*Import from COL* button in top right corner).



Species

□	English name	Latin name	Family	Genus	Taxon ID	Taxon rank	EOL link
□	Aardvark	Orycteropus afer	-	-	74ZNF	Species	Go
□	aardvarks	Tubulidentata	-	-	49Y	Order	Go
□	Aardwolf	Proteles cristata	-	-	4N38B	Species	Go
□	Aba Leaf-nosed Bat	Hipposideros abae	-	-	3M35V	Species	Go
□	Abbott's Duiker	Cephalophus spadix	-	-	69J2P	Species	Go

In the form, choose classes to import taxa from and submit.


Home > Extra tables > Species > Import from Catalogue of Life
Import

Import from Catalogue of Life

Taxa belonging to the chosen classes will be loaded into Species table.

Actinopterygii
Amphibia
Aves
Cephalaspidomorphi
Leptocardii
Elasmobranchii
Holocephali
Mammalia
Myxini

Hold down ctrl (PC) or command (Mac) key to select multiple.

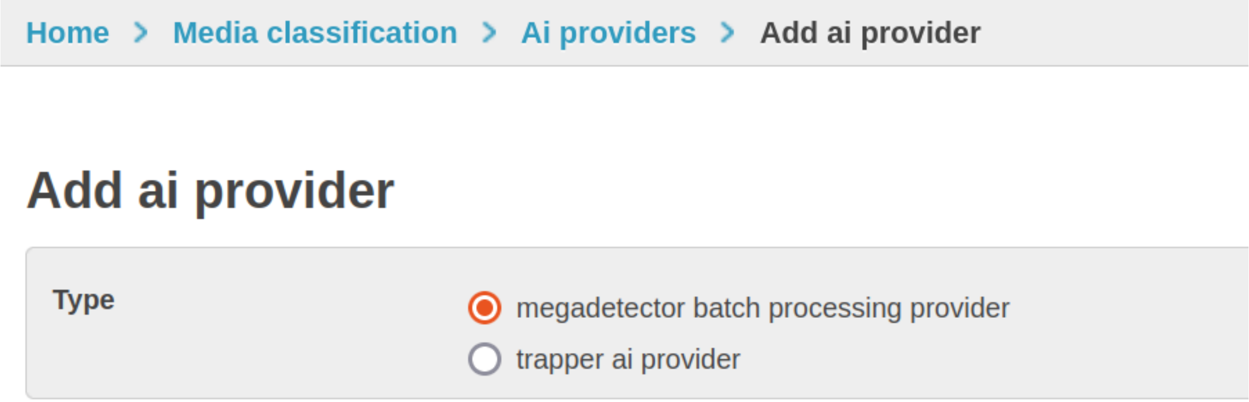
Note: Species import is run asynchronously, so the imported species may not be visible right away after running the action.

4.3 Add and configure AIProvider

To be able to use AI methods for classification, you need to add an *AIProvider* (for example Megadetector) to your *Research project*.

AIProvider needs to be manually added and configured via admin panel first.

Go to admin panel -> *Media Classification* -> *AIProvider* -> + *Add ai provider*.


Home › Media classification › Ai providers › Add ai provider

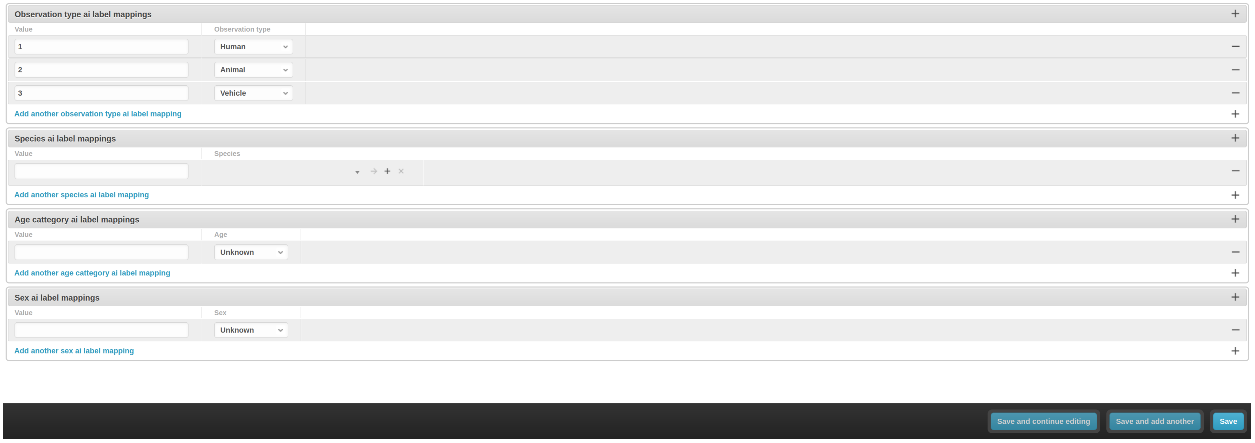
Add ai provider

Type ● megadetector batch processing provider
○ trapper ai provider

In the first part if the form, choose type of *AIProvider* to add. Regardless of the chosen type, several fields are required to fill:

- name for your *AIProvider* instance
- minimum confidence: default value is 0.9, meaning that predictions from AI with confidence lower than that will not be saved
- trapper instance url: url of your Trapper server
- API url: url of AI API

Last required part of *AIProvider* configuration is adding mappings.



Observation type ai label mappings

Value	Observation type
1	Human
2	Animal
3	Vehicle

Add another observation type ai label mapping

Species ai label mappings

Value	Species

Add another species ai label mapping

Age cattegory ai label mappings

Value	Age
	Unknown

Add another age cattegory ai label mapping

Sex ai label mappings

Value	Sex
	Unknown

Add another sex ai label mapping
Save and continue editing
Save and add another
Save

The mappings are used to translate AI classification results to Trapper classification attributes - add the values returned by your chosen AI API and map them to attribute values.

CHAPTER 5

Tutorial

by Tim R. Hofmeester (Swedish University of Agricultural Sciences, Department of Wildlife, Fish and Environmental Studies)

adapted by Karolina Kuczkowska and Jakub Bubnicki (Open Science Conservation Fund)

The guides below describe how to use TRAPPER core functionalities.

The first manual describes the process of uploading your data to the Trapper server. To start classifying *Resources*, you need to make a *Classification project* and a *Classificator*, as described in a second manual, and a third one for classifying *Resources* once you have a running *Classification project*.

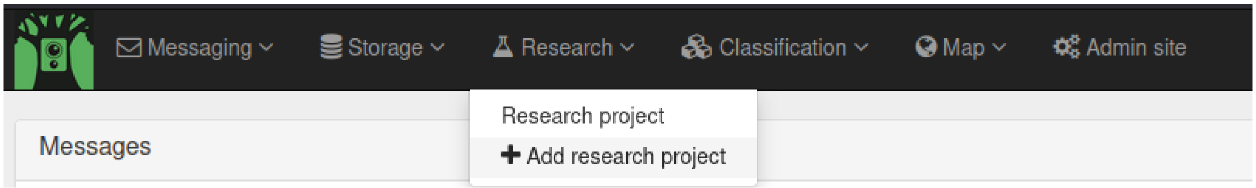
5.1 Opening and logging in to TRAPPER

Go to demo instance and register an account at the TRAPPER server. After your account is activated by admin, you can login on the same page using your email address and password.

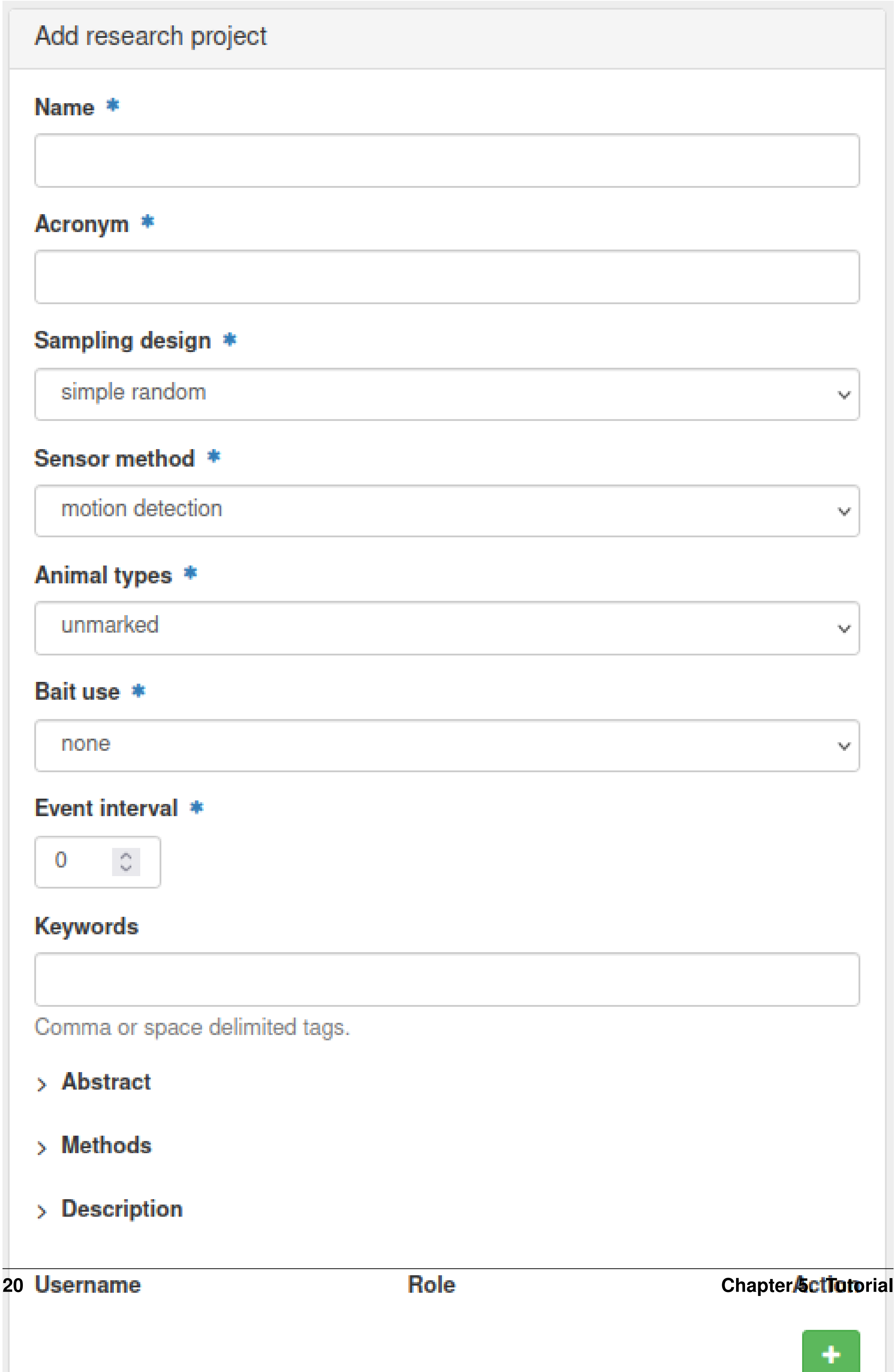
5.2 How to upload data into TRAPPER

5.2.1 1. Create a research project

In the top menu, select *Research* > *Add research project*.



You now get a menu with different options to fill in:



Add research project

Name *

Acronym *

Sampling design *

simple random

Sensor method *

motion detection

Animal types *

unmarked

Bait use *

none

Event interval *

0

Keywords

Comma or space delimited tags.

Fill in all information for your research project and click *Add research project*. Make sure you choose an acronym that is more than 3 characters long, which will make it easier to find it with the auto-fill function in Trapper.

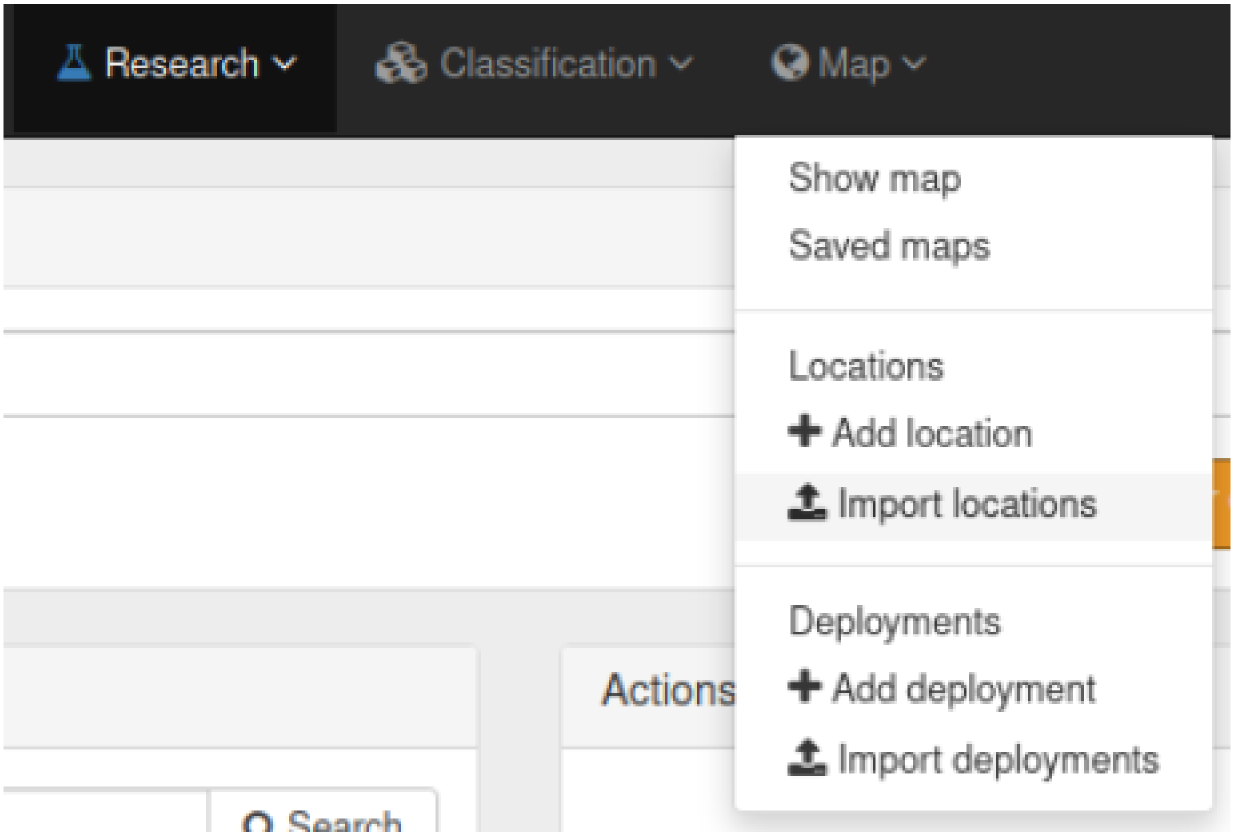
Note: Right after adding your research project you will see the notification that your project has been successfully added but needs to be activated by an admin. In practice, it means that your project request has been placed in the Admin dashboard, and has to be manually approved by an admin. This is one of the security measures and ways to control the amount and content of the projects in your institution/team. If you are an admin yourself you can go to the Admin dashboard and approve the project.

As soon as an Admin approves your new project, it will appear in your *Research* view in a *Research projects* table presenting the project Name, Acronym, Keywords, and Actions. In *Research projects* all the projects, you have access to, are listed. You can filter your projects according to the content and/or the role you play in them.

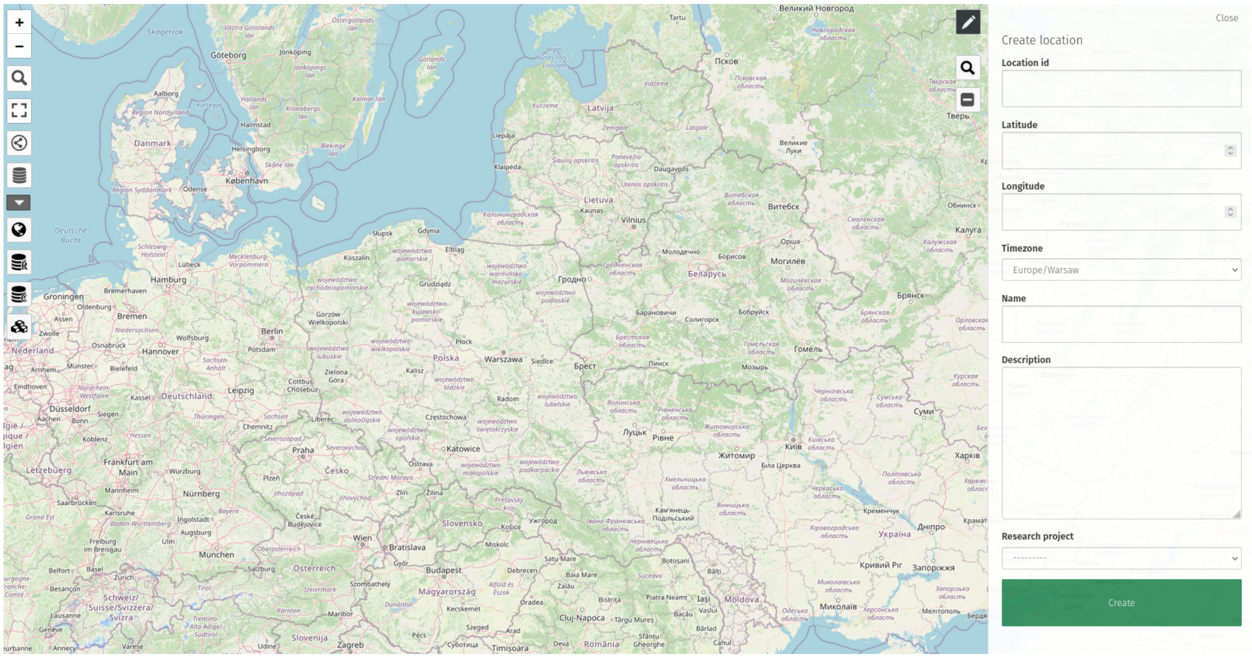
5.2.2 2. Add locations to TRAPPER

Once you have created a *Research project*, you can add *Locations* to your project. There are two ways to do this in TRAPPER: You can add locations manually, or you can upload a file with location data.

To manually add locations, select *Map* > *Locations* > *Add location*.



You now go to the map view where you can manually add information for a camera trap location.



Close

Create location

Location id

Latitude

Longitude

Timezone

Europe/Warsaw

Name

Description

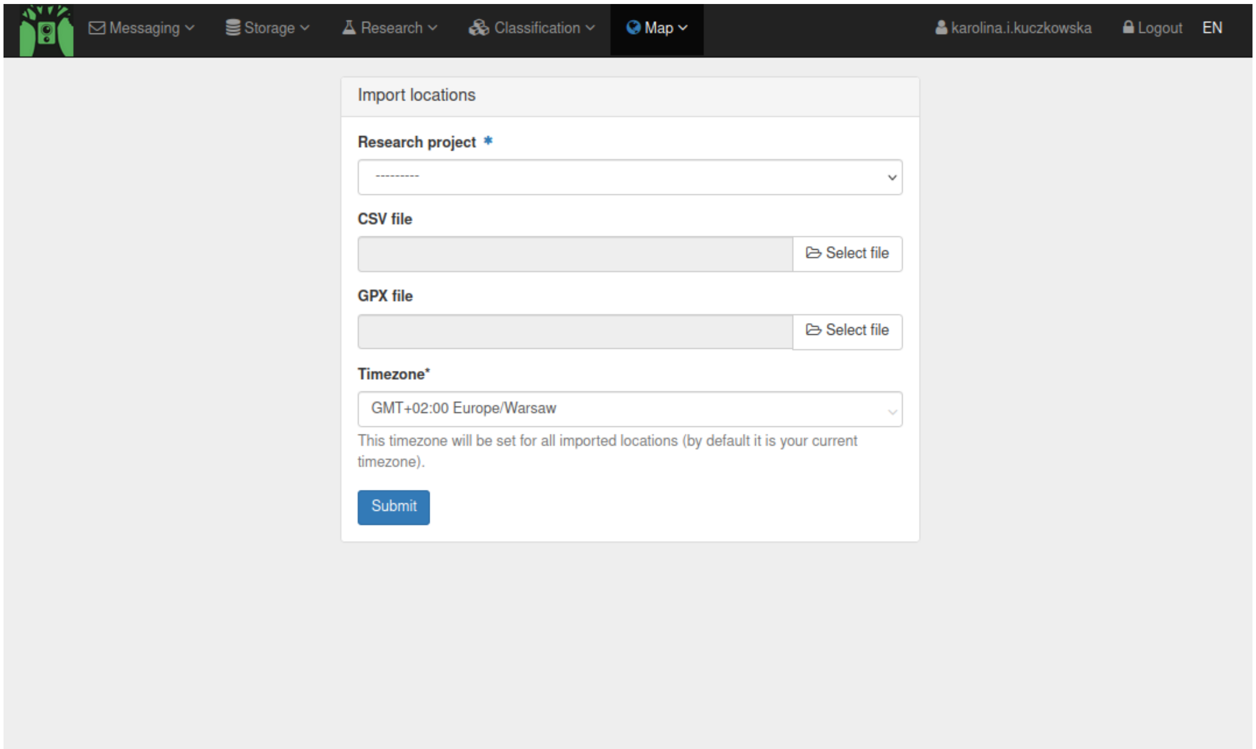
Research project

Create

Most of the time, you want to add multiple locations at once, which can be done using a .csv or a .gpx file with location information. It is easiest with a .csv file, which should be comma- separated and should contain the following columns:

- `locationID`: the ID/name of the location
- `longitude`: the longitude of the location in WSG84 decimal degrees (e.g., 19.58536776)
- `latitude`: the latitude of the location in WSG84 decimal degrees (e.g., 63.55417296)

Once you made the file, you can add the locations to TRAPPER using *Map* > *Locations* > *Import locations*.



Import locations

Research project *

CSV file

Select file

GPX file

Select file

Timezone*

GMT+02:00 Europe/Warsaw

This timezone will be set for all imported locations (by default it is your current timezone).

Submit

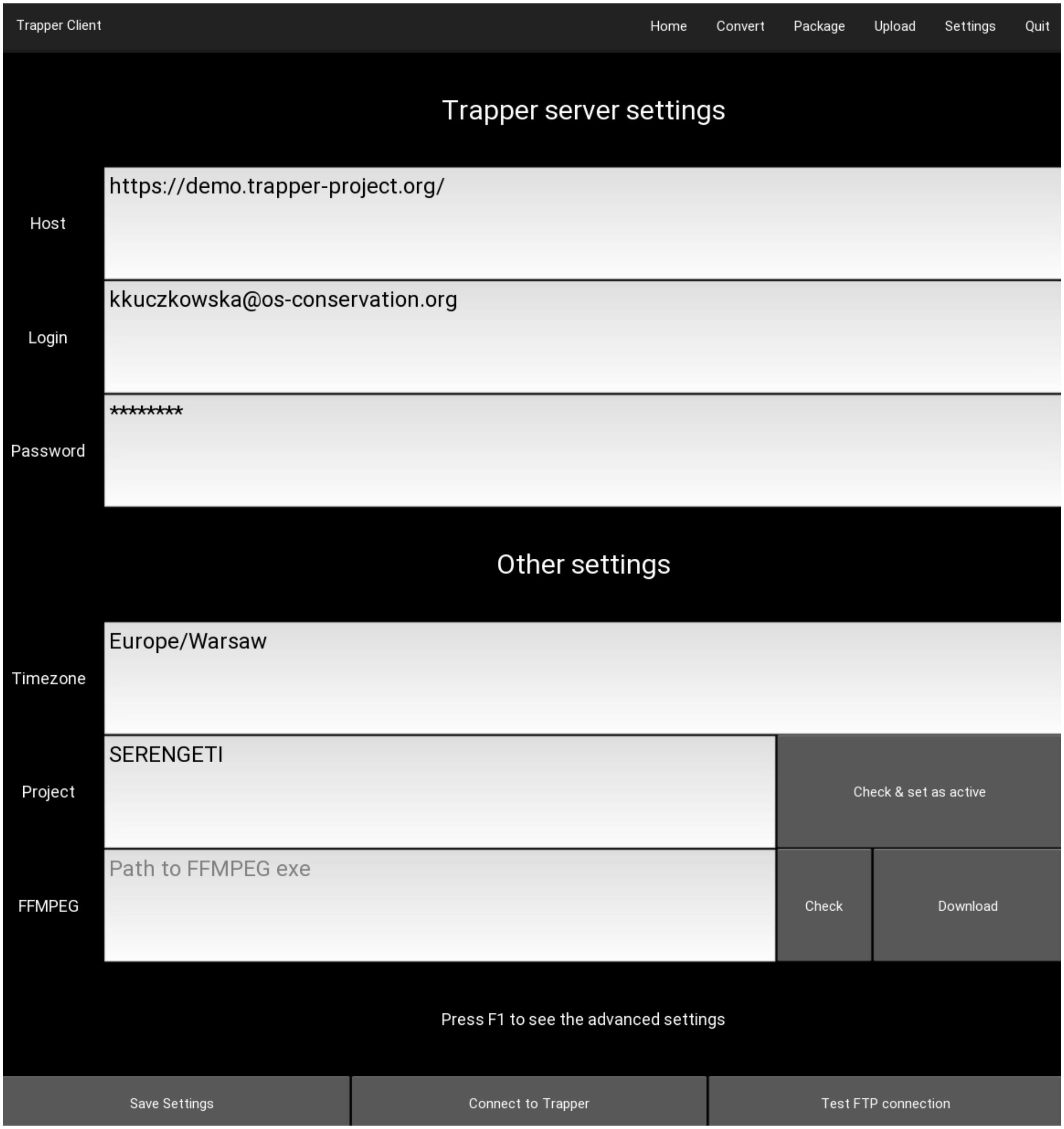
Select the .csv file, make sure you select the right Timezone (e.g., Europe/Stockholm) and select the *Research project* to which you want to add the locations. Once you have done this, you can select *Upload* to add the *Locations* to your *Research project*.

5.2.3 3. Set up Trapper-Client

The developers of TRAPPER provide a standalone software package to deal with the different steps of uploading data called 'Trapper Client'. Trapper Client can also be used to create the correct file to upload *Deployments* based on the camera trap data. Trapper Client is available for Windows and Linux from the GitLab of the Open Science Conservation Fund (the foundation behind Trapper): https://gitlab.com/oscf/trapper-client-bin.

The download consists of a .zip file that you need to extract anywhere on your computer where you have writing rights. Make sure you download the latest version and extract the complete .zip file in one location and keep the folder structure to ensure that the application works.

After unzipping, you can start the Trapper Client by double clicking the TrapperClient.bat file. First go to the 'Settings' page. Here you can add the credentials of your Trapper account.



Trapper server settings

Host: https://demo.trapper-project.org/

Login: kkuczkowska@os-conservation.org

Password: ********

Other settings

Timezone: Europe/Warsaw

Project: SERENGETI

FFMPEG: Path to FFMPEG exe

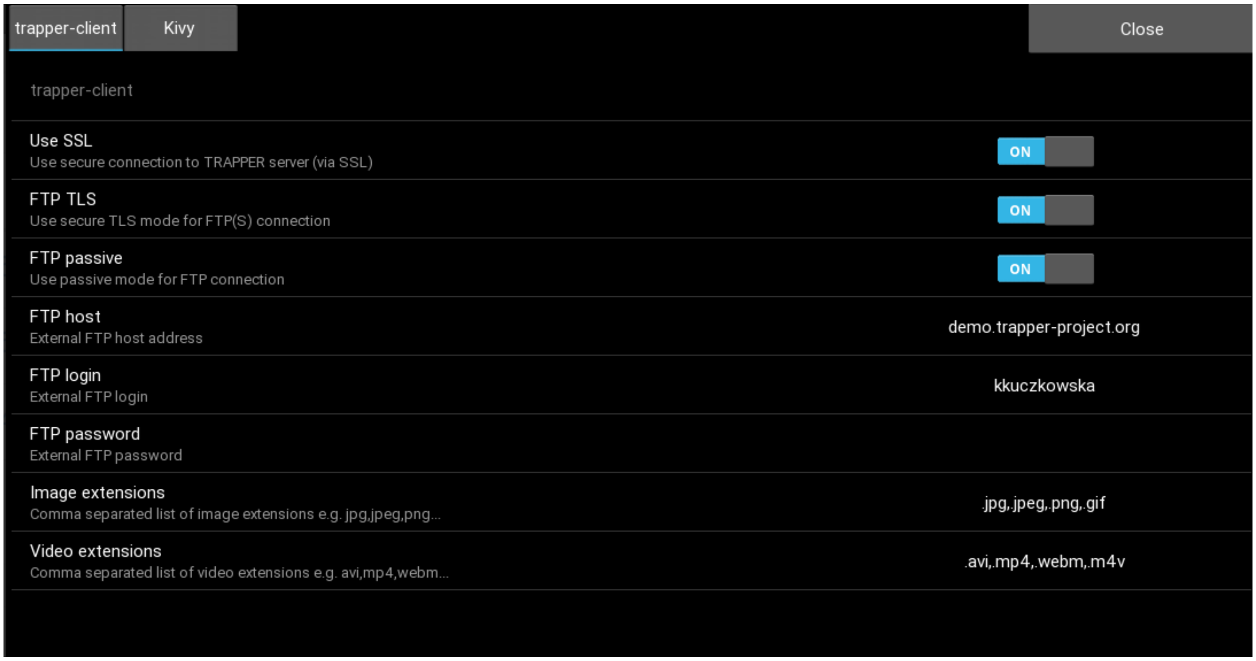
Press F1 to see the advanced settings

Note: Make sure you add `http(s)://` before the URL of the Trapper server and have your full email as login.

Make sure you add the acronym of the research project you want to upload data to in the 'Project' field. You can test if you have the right acronym by clicking 'Check & set as active'. You can also check if the Trapper Client can connect to the server by clicking 'Connect to Trapper'.

As the last step make sure you have an FTP account connected to your Trapper account (ask the admin of your Trapper server for this).

In order to upload data to an external FTP server linked to your TRAPPER instance, to switch between passive and active FTP mode or to modify the list of supported image and video extensions, press F1 to see the advanced settings.



trapper-client

Use SSL
Use secure connection to TRAPPER server (via SSL) — ON

FTP TLS
Use secure TLS mode for FTP(S) connection — ON

FTP passive
Use passive mode for FTP connection — ON

FTP host
External FTP host address — demo.trapper-project.org

FTP login
External FTP login — kkuczkowska

FTP password
External FTP password

Image extensions
Comma separated list of image extensions e.g. jpg,jpeg,png... — .jpg,.jpeg,.png,.gif

Video extensions
Comma separated list of video extensions e.g. avi,mp4,webm... — .avi,.mp4,.webm,.m4v

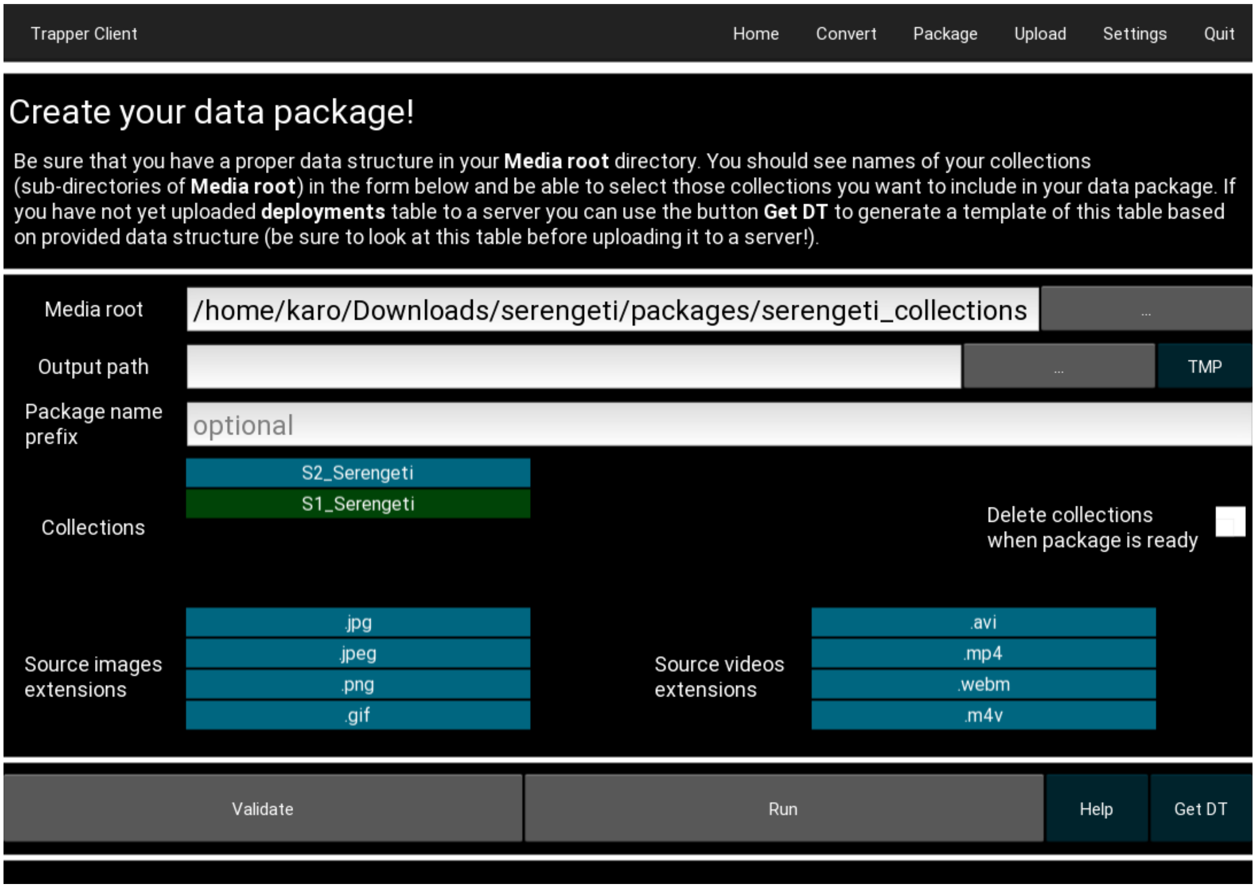
5.2.4 4. Add deployments to TRAPPER

Now that you have added your locations to trapper, you can start adding *Deployments*. The main differences between Locations and Deployments is that *Locations* only specify the coordinates at which a camera trap was located, while *Deployments* also specify the start and end date and time of a camera trap in the field. A *Location* can have several *Deployments*. Usually, a *Deployment* is determined as one session of putting out the camera and picking it up, or a session between changing cards and batteries.

Again it is possible to manually add *Deployments*, but this is rather error prone. Therefore, in this manual I will only deal with adding *Deployments* based on the camera trap data itself. This will make sure that deployment times are correct for all *Deployments*, also the ones with cameras that stopped working before they were picked up.

You can create a .csv file with your *Deployment* information in Trapper Client. For this to work, the data has to be ordered in the correct way: A main directory which you can select in the client, with sub-directories for each *Collection*. A *Collection* can for example be one round of *Deployments*. Within each *Collection*, there are further sub-directories for each *Deployment*. These sub-directories should be labelled in congruence with other information provided to TRAPPER. The *Deployment* label is a combination of a unique code and the *Location* label. For example, you can use a unique code `R1` (for round 1) for the first *Deployment* at a certain *Location* e.g. `32_01`. The Deployment label then becomes: `R1-32_01`.

Go to the 'Package' tab in your Trapper Client. Select the main directory as described above in the 'Media root' box. Also, select a folder where you want the .csv file with the *Deployment* table to be stored as 'Output path'. Now select the *Collection* for which you want to create the *Deployment* table and select the file type for the images (for Reconyx cameras this is .jpg). You can then click `Get DT` to create the *Deployment* table.



Create your data package!

Be sure that you have a proper data structure in your **Media root** directory. You should see names of your collections (sub-directories of **Media root**) in the form below and be able to select those collections you want to include in your data package. If you have not yet uploaded **deployments** table to a server you can use the button **Get DT** to generate a template of this table based on provided data structure (be sure to look at this table before uploading it to a server!).

Media root: /home/karo/Downloads/serengeti/packages/serengeti_collections

Output path:

Package name prefix: optional

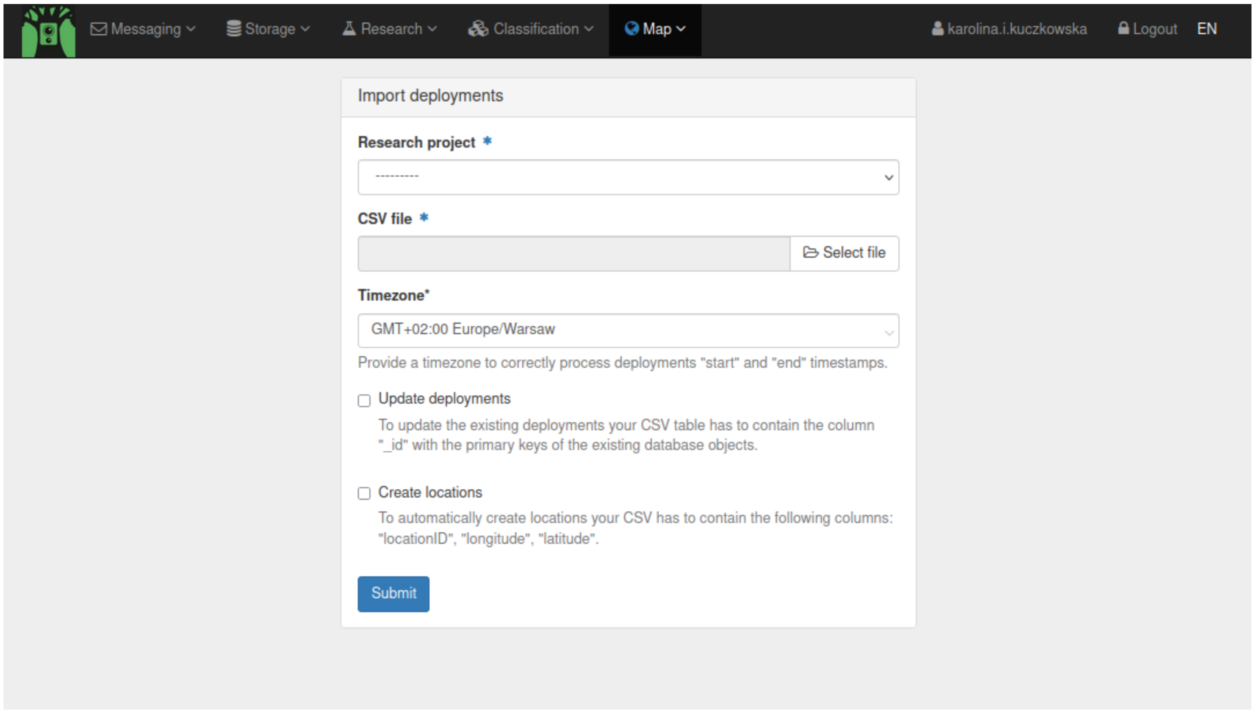
Collections: S2_Serengeti, S1_Serengeti

Delete collections when package is ready □

Source images extensions: .jpg, .jpeg, .png, .gif

Source videos extensions: .avi, .mp4, .webm, .m4v

You can now rename the .csv file (as the standard output doesn't include the collection name which means that it will be overwritten if you make a new *Deployment* table) and upload the .csv to TRAPPER by selecting *Map > Deployments > Import deployments*.



Import deployments

Research project *: ---------

CSV file *:

Timezone*: GMT+02:00 Europe/Warsaw

Provide a timezone to correctly process deployments "start" and "end" timestamps.

□ Update deployments

To update the existing deployments your CSV table has to contain the column "_id" with the primary keys of the existing database objects.

□ Create locations

To automatically create locations your CSV has to contain the following columns: "locationID", "longitude", "latitude".

Note: The automatically generated *Deployment* table contains only a minimum set of metadata fields needed to import deployments into TRAPPER. See https://tdwg.github.io/camtrap-dp/data/#deployments for a full list of supported metadata fields that can be imported into TRAPPER.

Note: Please note that you if your *Deployment* table contains `longitude` and `latitude` fields, location objects associated with imported deployments can be created automatically and you can skip the previous step of importing the *Location* table (*2. Add locations to TRAPPER*).

Select the previously generated .csv file and the *Research project* to which the round of *Deployments* belongs and click *Submit*.

5.2.5 5. Add resources to TRAPPER

Once the *Deployments* are added, the *Resources* (e.g. images) made by the camera traps can be uploaded into TRAPPER. This process consists of three steps:

1. Create a data package
2. Upload the data package via FTP
3. Load the data into Trapper

Create a data package

Go to the ‘Package’ tab in your Trapper Client. Select the main directory as described above in the ‘Media root’ box. Also, select a folder where you want the .csv file with the *Deployment* table to be stored as ‘Output path’. Now select the *Collection* for which you want to create the *Deployment* table and select the file type for the images (for Reconyx cameras this is .jpg).

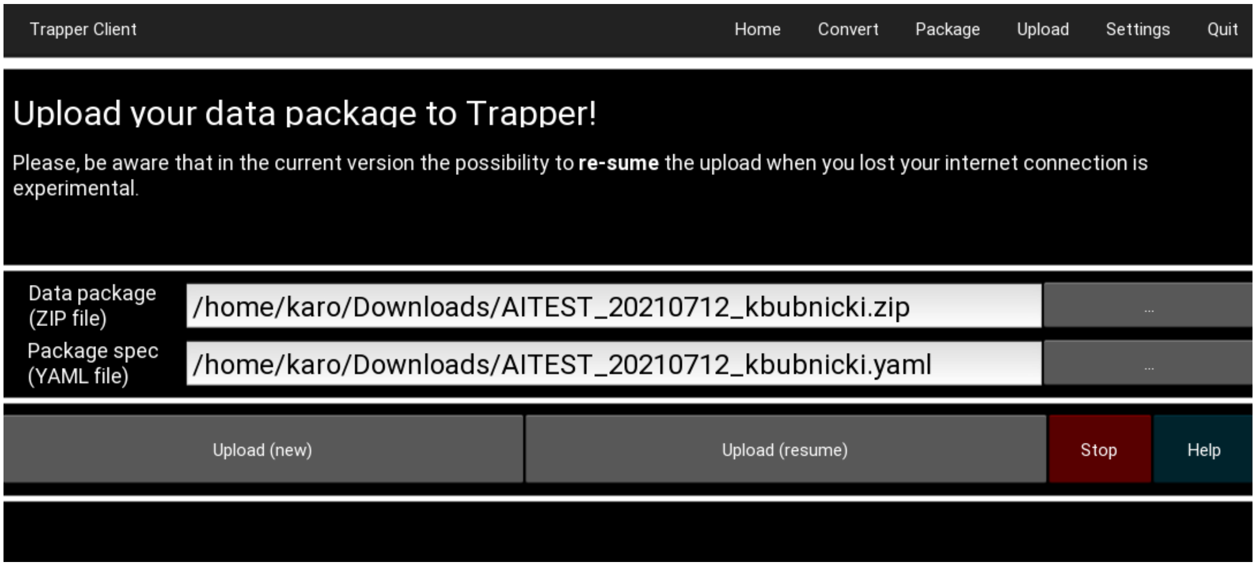
Note: These steps are the same as for creating your Deployment table.

Now click ‘Validate’. This will check on the Trapper server if the deployments you want to add have been added as deployments in the previous step. If this is successful, you can create the data package by clicking ‘Run’. This will create two files, a .zip file and a .yaml file. The .zip file contains all your data, while the .yaml file contains the meta-data (it’s a package definition file).

Upload the data package via FTP

Once you have generated the files, you can upload them to the Trapper server. Make sure your settings are correct (see above). Then go to the ‘Upload’ tab. Here you have to select both the .zip and the .yaml file that you just created. You upload them by clicking on ‘Upload (new)’.

Note: You can also upload your files to FTP server using other tools, for example via command line with *ftp* command, using your trapper account credentials to login.



Upload your data package to Trapper!

Please, be aware that in the current version the possibility to **re-sume** the upload when you lost your internet connection is experimental.

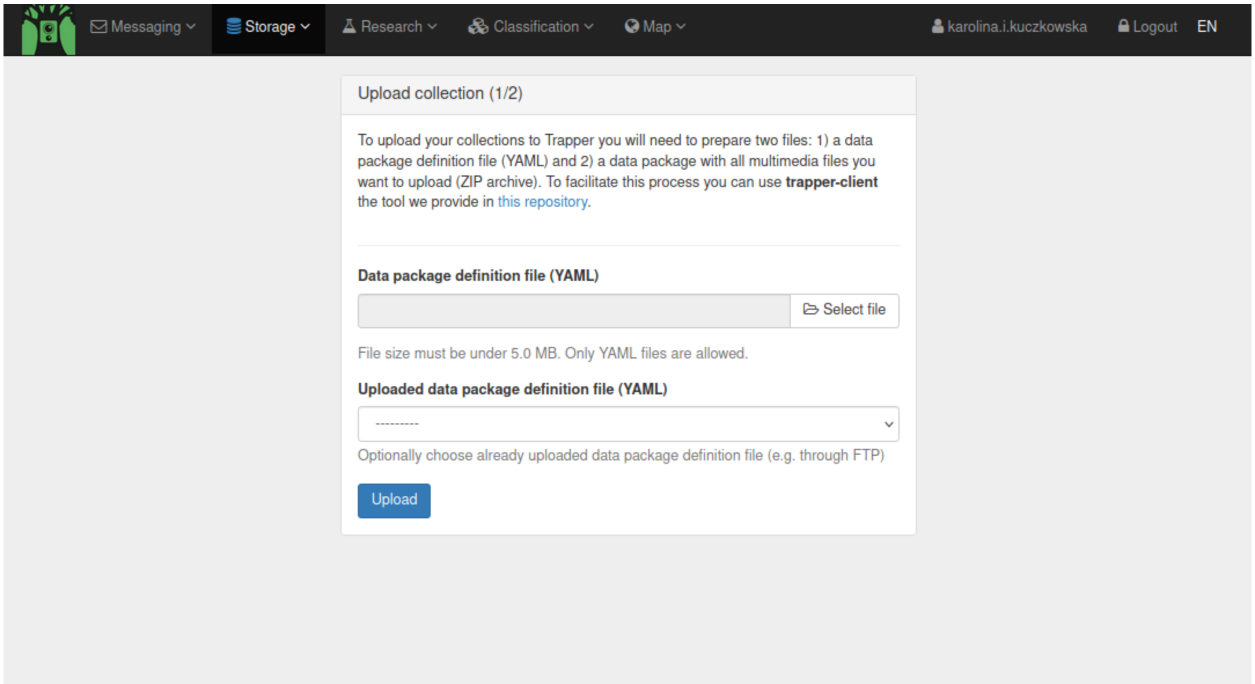
Data package (ZIP file): /home/karo/Downloads/AITEST_20210712_kbubnicki.zip

Package spec (YAML file): /home/karo/Downloads/AITEST_20210712_kbubnicki.yaml

Load the data into Trapper

Once you have uploaded the files go to *Storage* > *Upload collection* on your Trapper server. Here you can select the .yaml file that you just uploaded and click 'Upload'.

Note: You can also upload the file through this interface, but it is recommended to do it through FTP.



Upload collection (1/2)

To upload your collections to Trapper you will need to prepare two files: 1) a data package definition file (YAML) and 2) a data package with all multimedia files you want to upload (ZIP archive). To facilitate this process you can use **trapper-client** the tool we provide in this repository.

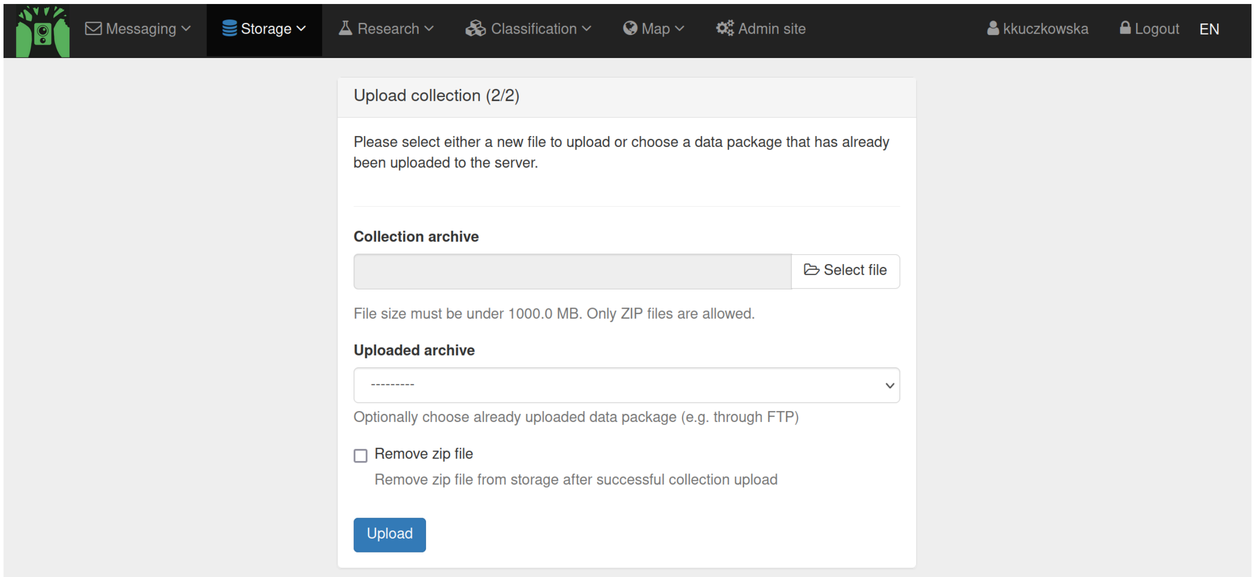
Data package definition file (YAML)

File size must be under 5.0 MB. Only YAML files are allowed.

Uploaded data package definition file (YAML)

Optionally choose already uploaded data package definition file (e.g. through FTP)

You then need to select the .zip file you just uploaded in the second window. Alternatively, you can upload the .zip file here, but again it is recommended to do that via FTP and Trapper Client. Click 'Upload' to load you data package into Trapper.



Upload collection (2/2)

Please select either a new file to upload or choose a data package that has already been uploaded to the server.

Collection archive

Select file

File size must be under 1000.0 MB. Only ZIP files are allowed.

Uploaded archive

Optionally choose already uploaded data package (e.g. through FTP)

□ Remove zip file

Remove zip file from storage after successful collection upload

Upload

You have now created a *Collection* containing the *Resources* of all the *Deployments* that you used while creating the *Collection*. This *Collection* is added to the correct *Research project* automatically. You can browse the *Resources* in the *Collection* by going to *Storage* > *Resources (available)*.

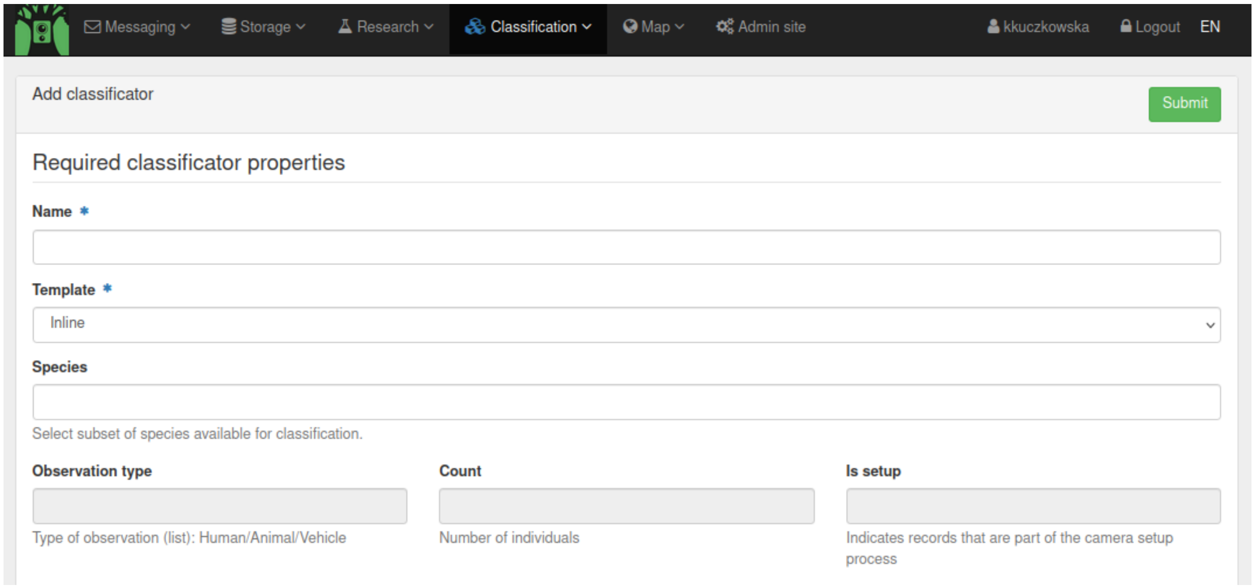
5.3 How to create a Classificator and Classification project in TRAPPER

5.3.1 1. Create a Classificator

In TRAPPER, images are classified in *Classification projects* using a *Classificator*. One of the great flexibilities of TRAPPER is that researchers or students can build their own *Classificator* and *Classification project* depending on the research question. A *Classificator* basically describes a set of attributes that is used while annotating an image or sequence of images.

In the top menu, select *Classification* > *Add classificator*. First there are some required classificator properties that you have to fill in, such as name and template. Furthermore there are several attributes that have become mandatory in the latest version of Trapper, which are species, observation type, count and 'is_setup'. The 'is_setup' and observation type were added to make it easier to include a first screening of images by Microsofts MegaDetector, a Machine Learning Algorithm that automatically identifies empty images from images containing a human, an animal or a vehicle. Also, you can add species you want to classify based on entries in the species list.

Note: Admins can add species to the overall species list in the Admin panel under *Species* either a single species at the time or using a .csv file, or download taxa for entire classes from Catalogue of Life.



Add classificator

Required classificator properties

Name *

Template *

Inline

Species

Select subset of species available for classification.

Observation type

Type of observation (list): Human/Animal/Vehicle

Count

Number of individuals

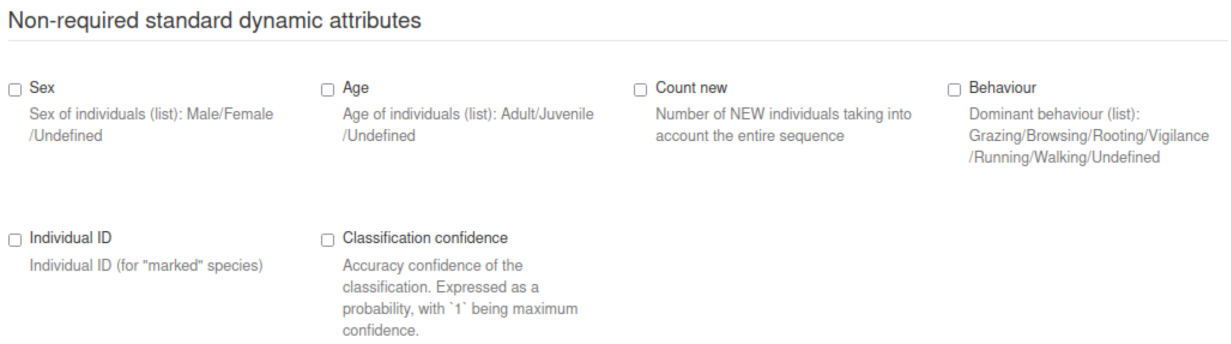
Is setup

Indicates records that are part of the camera setup process

Further down on the page you see a lot of different options, which I will explain step by step.

There are two types of attributes: static and dynamic attributes. Static attributes are attributes that are only given once for each resource, e.g., ‘is setup’. Dynamic attributes are attributes that can be given multiple times for one resource, e.g., the species or sex of an animal recorded by the camera trap.

There are several pre-defined attributes available to quickly add to your *Classificator* by checking the boxes.



Non-required standard dynamic attributes

□ Sex
Sex of individuals (list): Male/Female /Undefined

□ Age
Age of individuals (list): Adult/Juvenile /Undefined

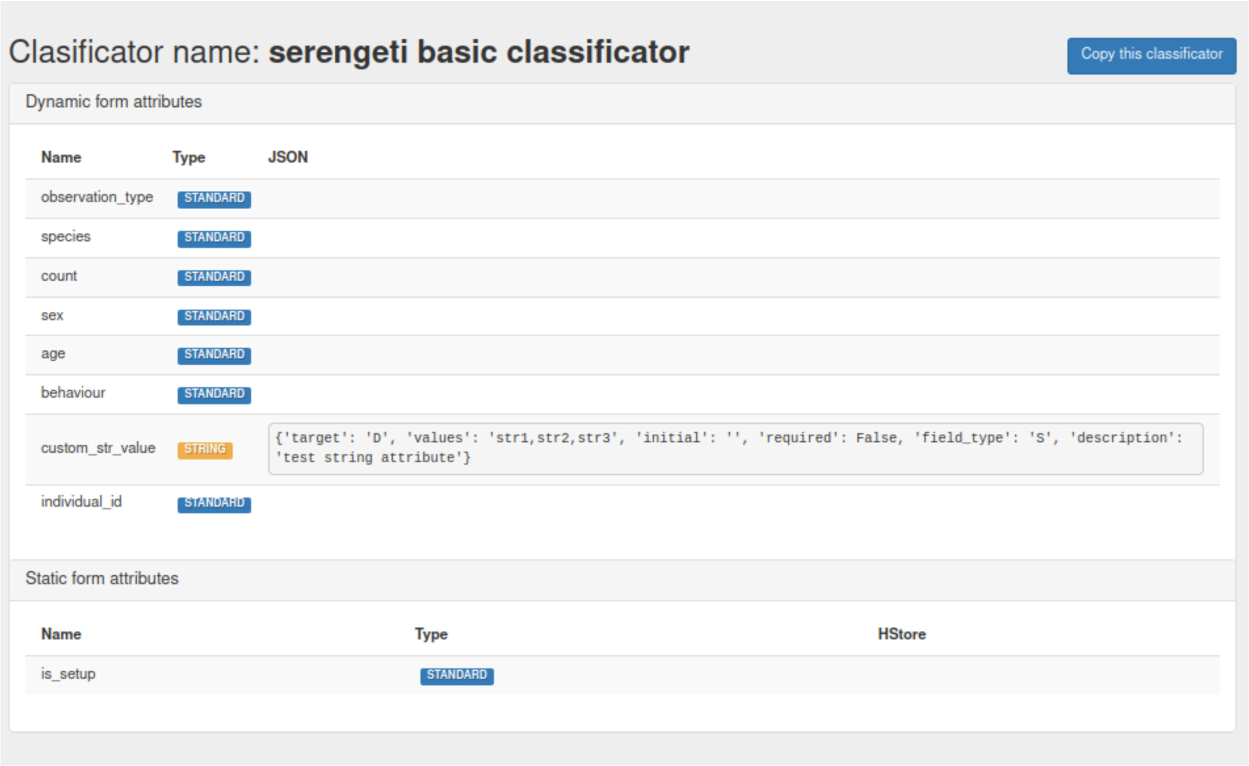
□ Count new
Number of NEW individuals taking into account the entire sequence

□ Behaviour
Dominant behaviour (list): Grazing/Browsing/Rooting/Vigilance /Running/Walking/Undefined

□ Individual ID
Individual ID (for "marked" species)

□ Classification confidence
Accuracy confidence of the classification. Expressed as a probability, with `1` being maximum confidence.

Then you can also add custom attributes in the custom attribute form. This form gives a lot of freedom to determine your own attributes. Below you find the attributes of an example *Classificator*.



Clasificator name: **serengeti basic classificator**

Dynamic form attributes

Name	Type	JSON
observation_type	STANDARD	
species	STANDARD	
count	STANDARD	
sex	STANDARD	
age	STANDARD	
behaviour	STANDARD	
custom_str_value	STRING	{'target': 'D', 'values': 'str1,str2,str3', 'initial': '', 'required': False, 'field_type': 'S', 'description': 'test string attribute'}
individual_id	STANDARD	

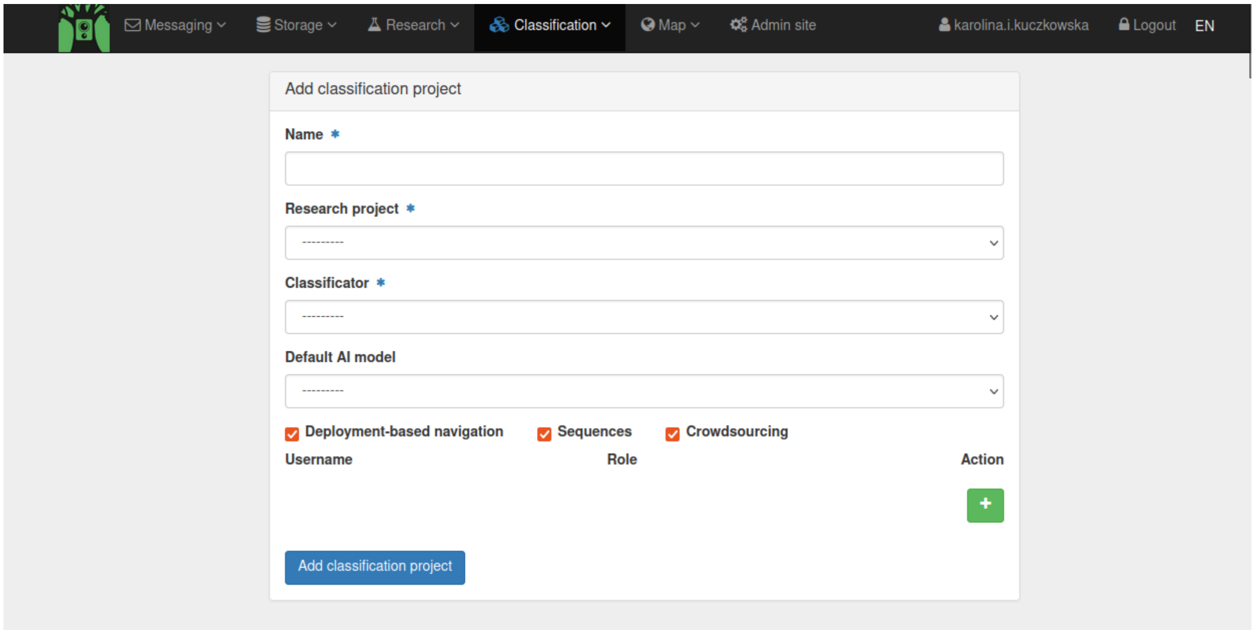
Static form attributes

Name	Type	HStore
is_setup	STANDARD	

You can check all the *Classificators* on your TRAPPER server by going to *Classification* > *Classificators*.

5.3.2 2. Create a Classification project

Once you have created your *Classificator*, you can make a *Classification project* to start classifying your resources. You can make a new *Classification project* by going to *Classification* > *Add classification* project.



Add classification project

Name *

Research project *

Classificator *

Default AI model

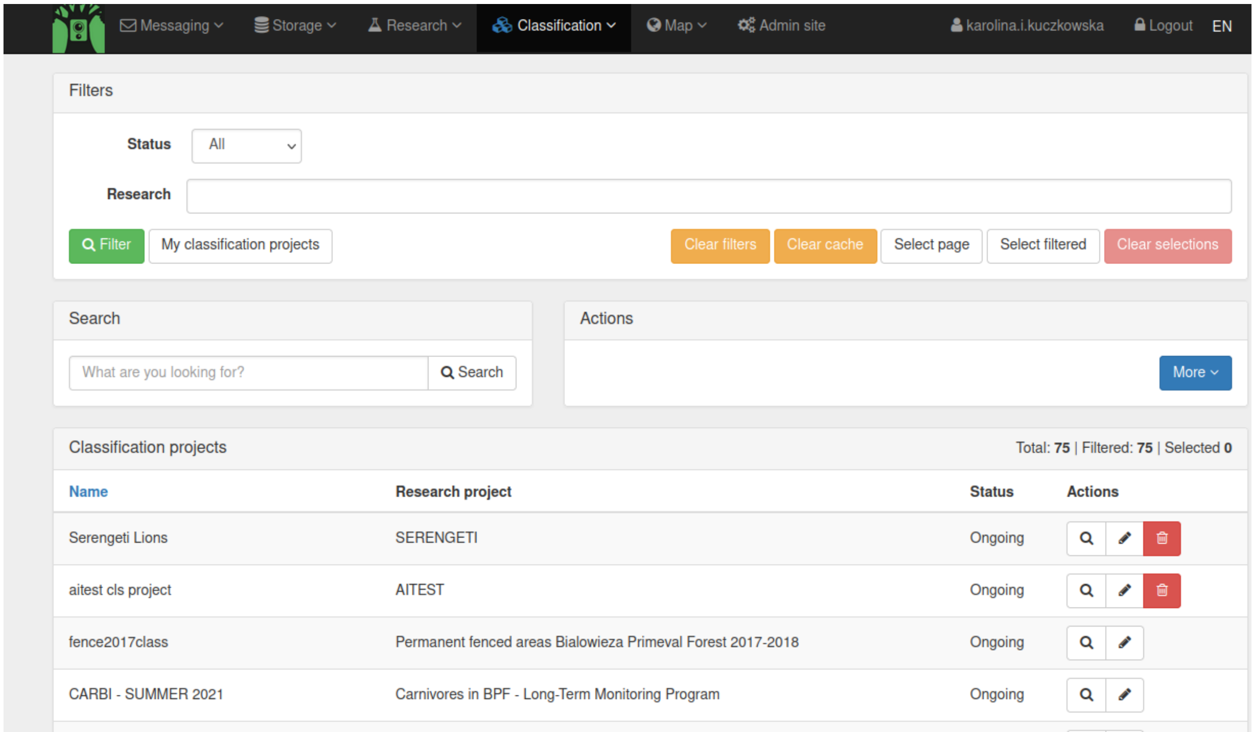
☑ Deployment-based navigation ☑ Sequences ☑ Crowdsourcing

Username	Role	Action
		+

Here you can specify the name of the *Classification project*, to which *Research project* the *Classification project* belongs, which *Classificator* and default *AIProvider* to use and which users are involved in the *Classification project*.

When you are finished filling in all details, click *Add classification project*. You have now created a *Classification project*.

You can check all the Classification projects to which you have access by going to *Classification* > *Classification projects*.



Filters

Status: All

Research:

Search

What are you looking for?

Actions

Classification projects

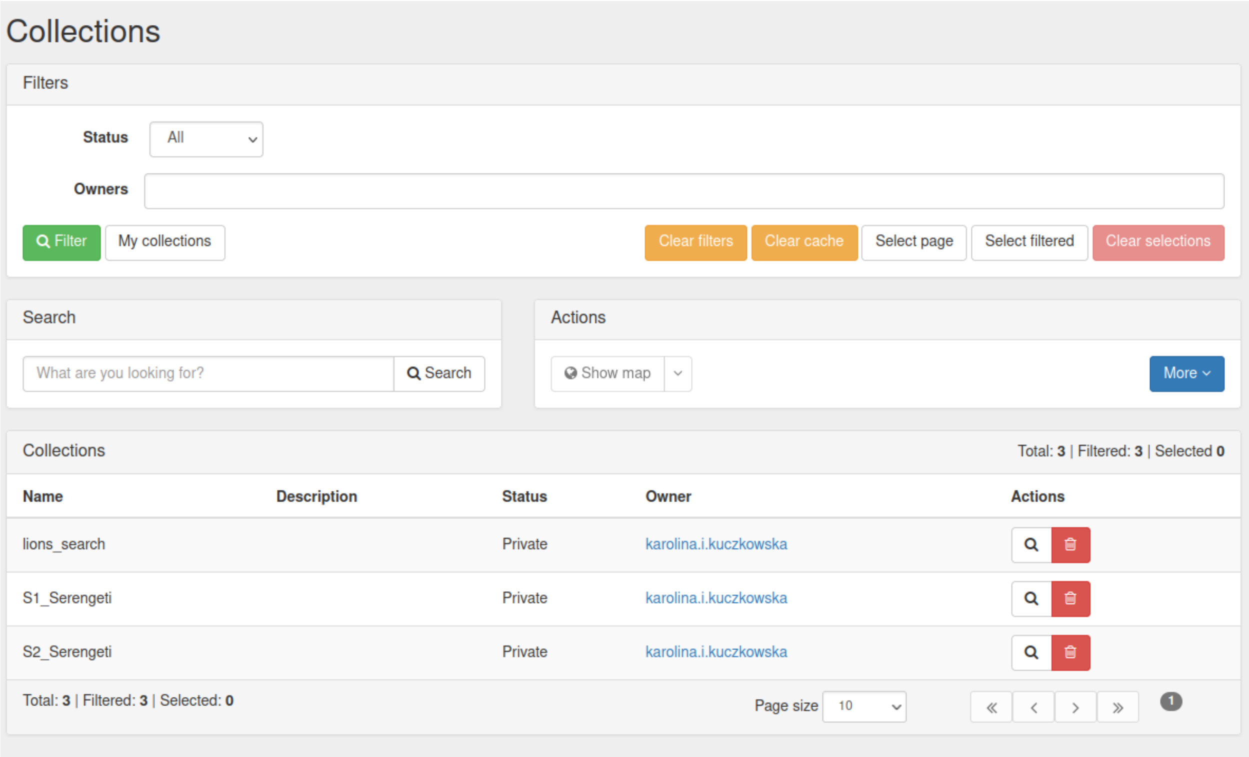
Total: **75** | Filtered: **75** | Selected **0**

Name	Research project	Status	Actions
Serengeti Lions	SERENGETI	Ongoing	
aitest cls project	AITEST	Ongoing	
fence2017class	Permanent fenced areas Bialowieza Primeval Forest 2017-2018	Ongoing	
CARBI - SUMMER 2021	Carnivores in BPF - Long-Term Monitoring Program	Ongoing	

5.3.3 3. Add Resources to a Classification project

In order to be able to classify *Resources*, you need to add them to your *Classification project*. To do this, first go to your *Research project* by going to *Research* > *Research project* and clicking on the magnifying glass next to your research project.

Now you see a selection of all *Collections* that are linked to the *Research project*.



Status: All

Owners:

Total: 3 | Filtered: 3 | Selected 0

Name	Description	Status	Owner	Actions
lions_search		Private	karolina.i.kuczkowska	
S1_Serengeti		Private	karolina.i.kuczkowska	
S2_Serengeti		Private	karolina.i.kuczkowska	

Total: 3 | Filtered: 3 | Selected: 0

Page size 10

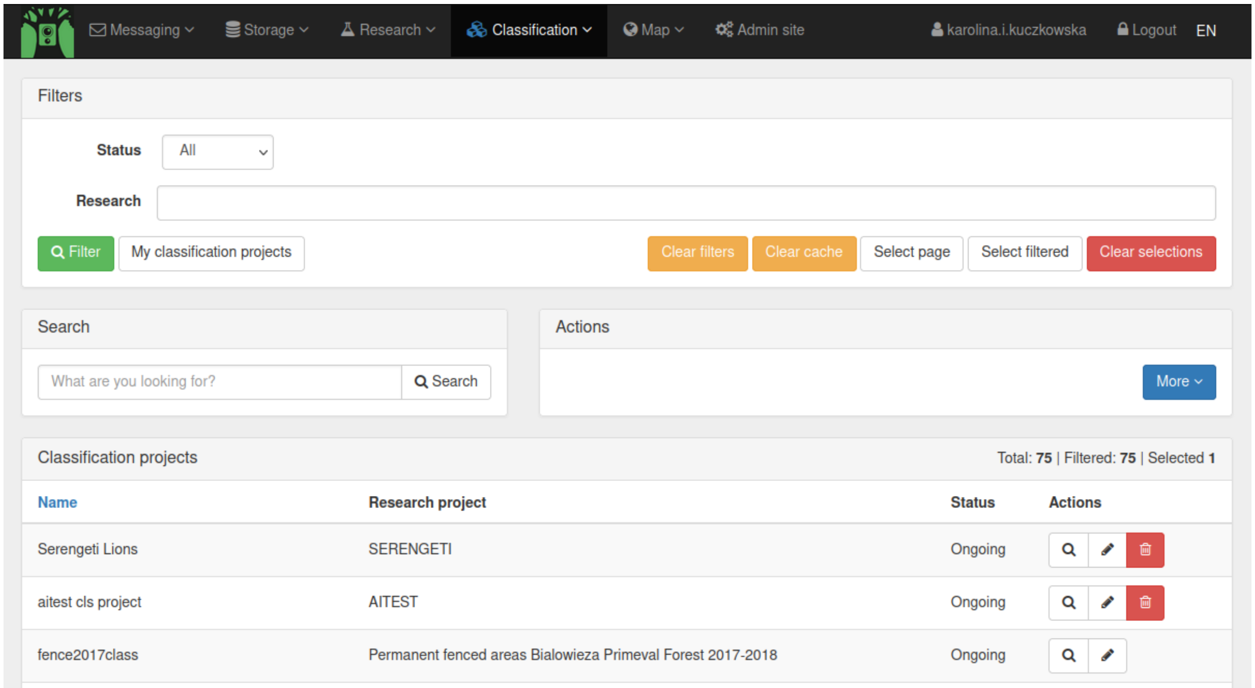
Select a *Collection* by clicking on it and under *Actions* > *More* select ‘Add to Classification project’.

You now have *Resources* in your *Classification project* that can be classified.

5.4 How to classify images in TRAPPER

5.4.1 1. Go to the classification page

In the top menu, select *Classification* > *Classification projects*. You are now at the start screen of all classification projects that you have access to. You will see information on each *Classification project*, including its name, the *Research project* it belongs to, its status and *Actions*.



Status: All

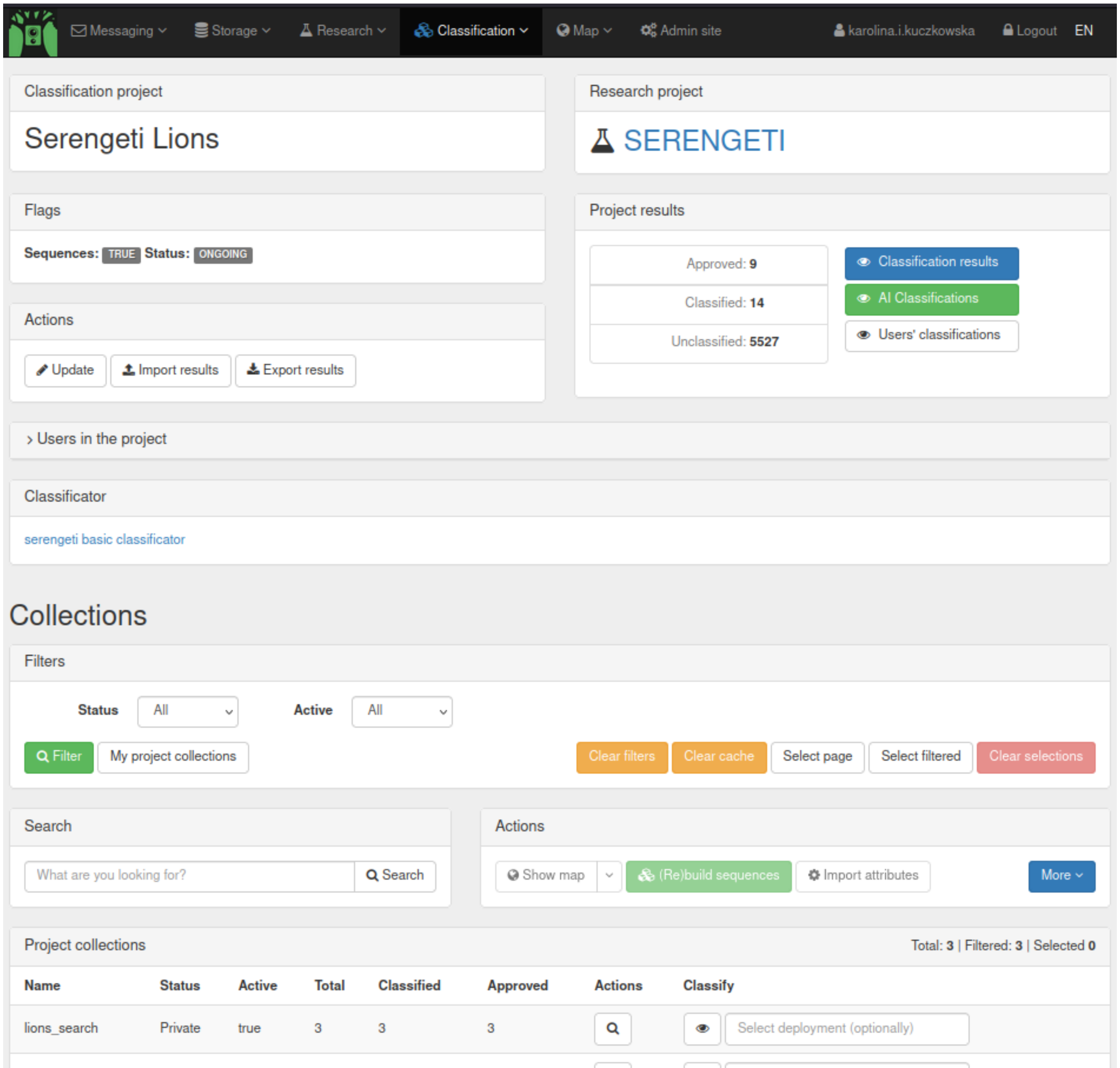
Research:

Classification projects

Total: **75** | Filtered: **75** | Selected **1**

Name	Research project	Status	Actions
Serengeti Lions	SERENGETI	Ongoing	🔍 ✎ 🗑
aitest cls project	AITEST	Ongoing	🔍 ✎ 🗑
fence2017class	Permanent fenced areas Bialowieza Primeval Forest 2017-2018	Ongoing	🔍 ✎

To select the *Classification project* you want to work in, select the magnifying glass underneath *Actions*.



Classification project

Serengeti Lions

Research project

SERENGETI

Flags

Sequences: TRUE **Status:** ONGOING

Project results

Approved: **9**
Classified: **14**
Unclassified: **5527**

Classificator

serengeti basic classificator

Collections

Filters

Status All **Active** All

Project collections — Total: **3** | Filtered: **3** | Selected **0**

Name	Status	Active	Total	Classified	Approved	Actions	Classify
lions_search	Private	true	3	3	3		

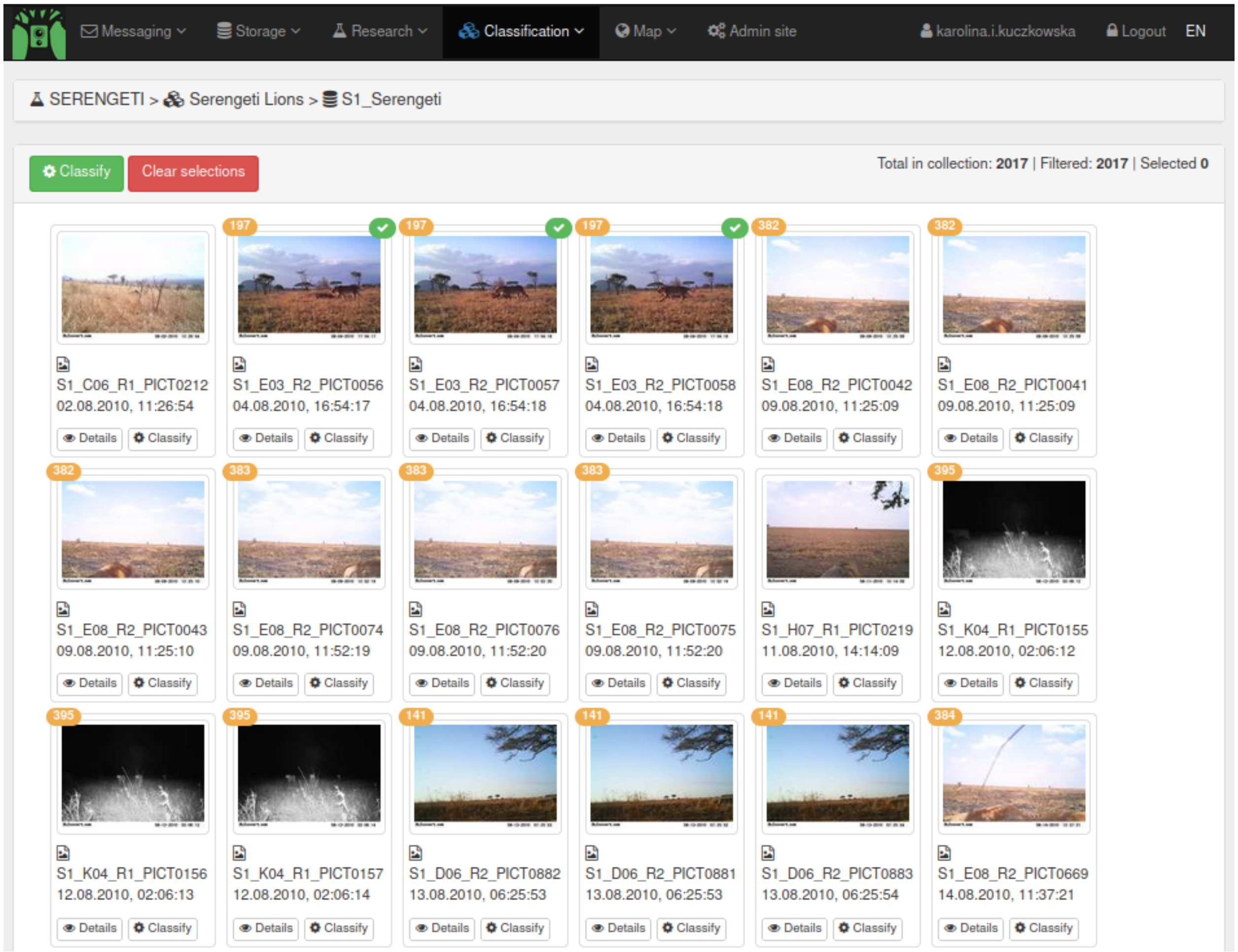
You are now in the start screen of the selected *Classification project*. Here you see more information on the *Classification project*, including which *Classificator* is used, which *Collections* are in the project, and to which *Research project* the *Classification project* belongs. Furthermore, you can see how many of the *Resources* have been Classified and Approved.

Go to the bottom of the page, where you can find the different *Collections* in the *Classification project* (for example the different rounds during which cameras where placed out in the field). Type the name of the *Deployment* within a *Collection* that you would like to classify and select it. This makes sure that you are classifying a specific *Deployment*, which makes it easier to keep an overview during the classification. Once you have selected the *Deployment*, select the eye symbol to start the classification.

Note: Make sure you generate sequences for your collection by selecting the collection and clicking '(Re)build sequences'. You can specify a time interval between sequences. The default is 5, regular use is 15 minutes.

5.4.2 2. Classify a sequence using the initial classification forms

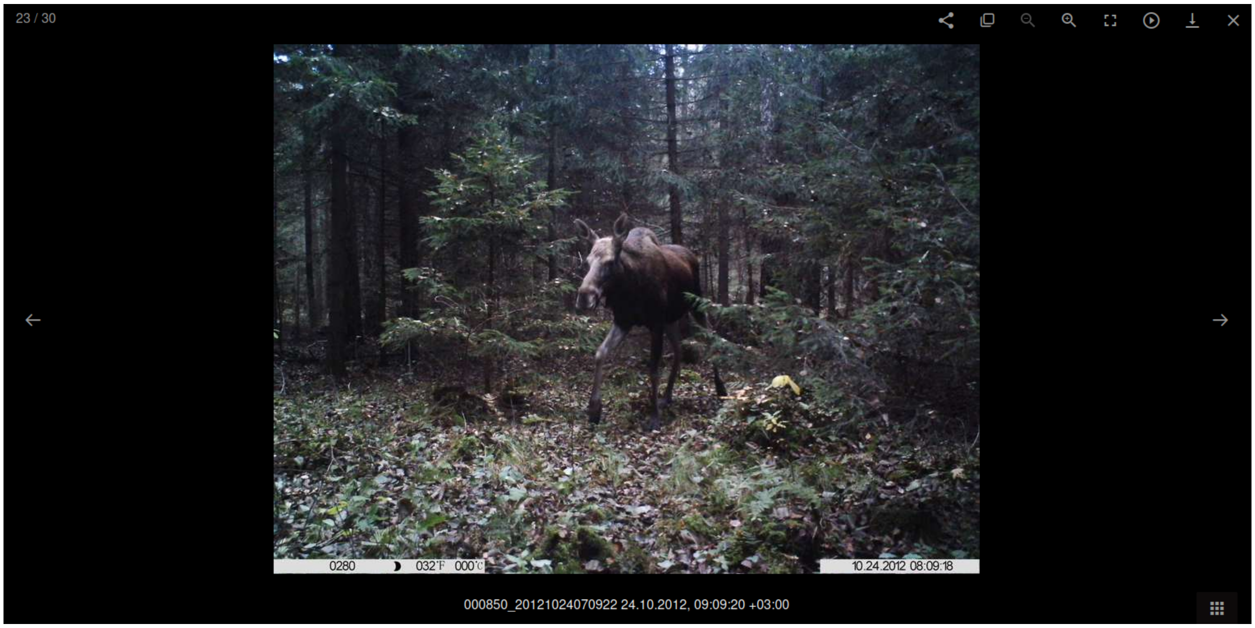
You will now see a screen such as the one below. You see the different *Resources* (images) with an orange number. This is the *Sequence* number.



S1_C06_R1_PICT0212 02.08.2010, 11:26:54
S1_E03_R2_PICT0056 04.08.2010, 16:54:17
S1_E03_R2_PICT0057 04.08.2010, 16:54:18
S1_E03_R2_PICT0058 04.08.2010, 16:54:18
S1_E08_R2_PICT0042 09.08.2010, 11:25:09
S1_E08_R2_PICT0041 09.08.2010, 11:25:09
S1_E08_R2_PICT0043 09.08.2010, 11:25:10
S1_E08_R2_PICT0074 09.08.2010, 11:52:19
S1_E08_R2_PICT0076 09.08.2010, 11:52:20
S1_E08_R2_PICT0075 09.08.2010, 11:52:20
S1_H07_R1_PICT0219 11.08.2010, 14:14:09
S1_K04_R1_PICT0155 12.08.2010, 02:06:12
S1_K04_R1_PICT0156 12.08.2010, 02:06:13
S1_K04_R1_PICT0157 12.08.2010, 02:06:14
S1_D06_R2_PICT0882 13.08.2010, 06:25:53
S1_D06_R2_PICT0881 13.08.2010, 06:25:53
S1_D06_R2_PICT0883 13.08.2010, 06:25:54
S1_E08_R2_PICT0669 14.08.2010, 11:37:21

Note: To circumvent problems of two people classifying the same Deployment, only start classifying a Deployment if none of the Sequences have been classified yet. You can see if Resource has been classified by a green mark (note 3 Resources - one Sequence - classified and others not classified in the screen cap above). At the same time, make sure that you classify all Sequences in a Deployment (keep track of which Deployment you are classifying). If you cannot finish a deployment, contact your admin.

You can now enlarge pictures by clicking on the thumbnail. You can scroll through the enlarged pictures with you arrow keys. It's always good to check the whole sequence before classifying it to check you didn't miss any individuals in the thumbnails.



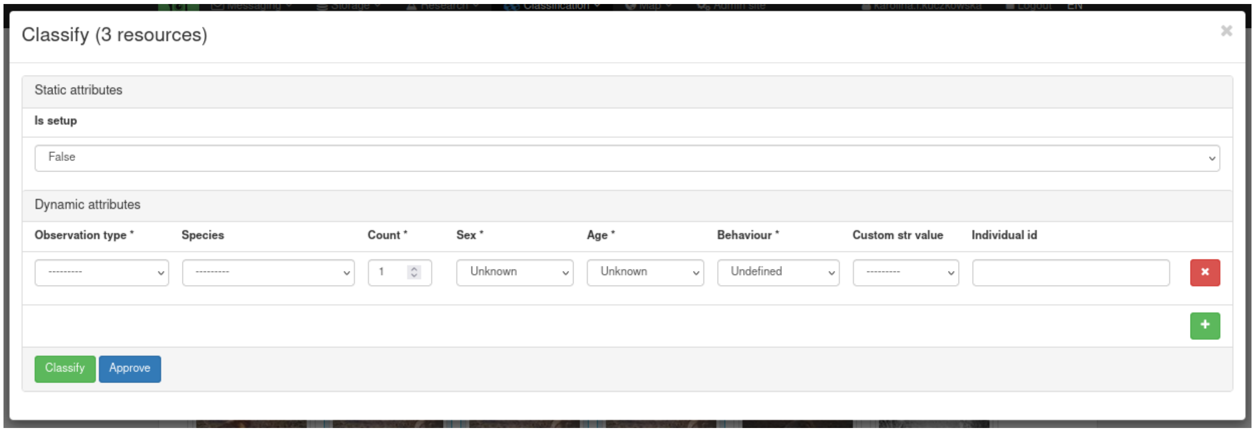
000850_20121024070922 24.10.2012, 09:09:20 +03:00

When you have decided how many individuals and which species is visible in the *Sequence*, it is time to classify the *Sequence*. To do this, click on the sequence number. This will select all images in the sequence. You can now click 'Classify' in the top of the screen to classify the whole sequence. **Make sure to check for empty images at the end of the sequence as it is good to classify those separately as 'observation_type = blank'.**

In the classification view, fill in the right information for the sequence. You classify 'is_setup' as *True* for setup and test images, you can then delete all dynamic attributes by clicking on the red cross.

Note: Classify **empty resources** as *Observation type = Blank*.

Otherwise, keep the default *False* and add the dynamic attribute information.



Classify (3 resources)

Static attributes

Is setup

False

Dynamic attributes

Observation type *	Species	Count *	Sex *	Age *	Behaviour *	Custom str value	Individual id
---------	---------	1	Unknown	Unknown	Undefined	---------	

Dynamic attributes are attributes that can be given multiple times for one sequence. In this case, it gives information about the type of observation, the species of animal, if the observation contains one or more animals, sex, age and behaviour of the animals, and optional custom str value (as specified in the *Classifier*) and individual id.

In the standard classification form there are three required attributes:

1. Observation type: the type of observation (Human/Vehicle/Animal/Blank/Unknown/Unclassified)
2. Species: The animal species shown in the image (to be selected from *Species* from *Classificator*)

3. Count: The number of animals/humans/vehicles, default is 1

There is also several attributes that can be assigned, as described in *Classificator* adding manual.

If there are multiple animals in the field with different attributes, you can add a row by clicking the green button with a plus sign to specify the attributes of all animals in the sequence.

Click 'Classify' to finish the classification.

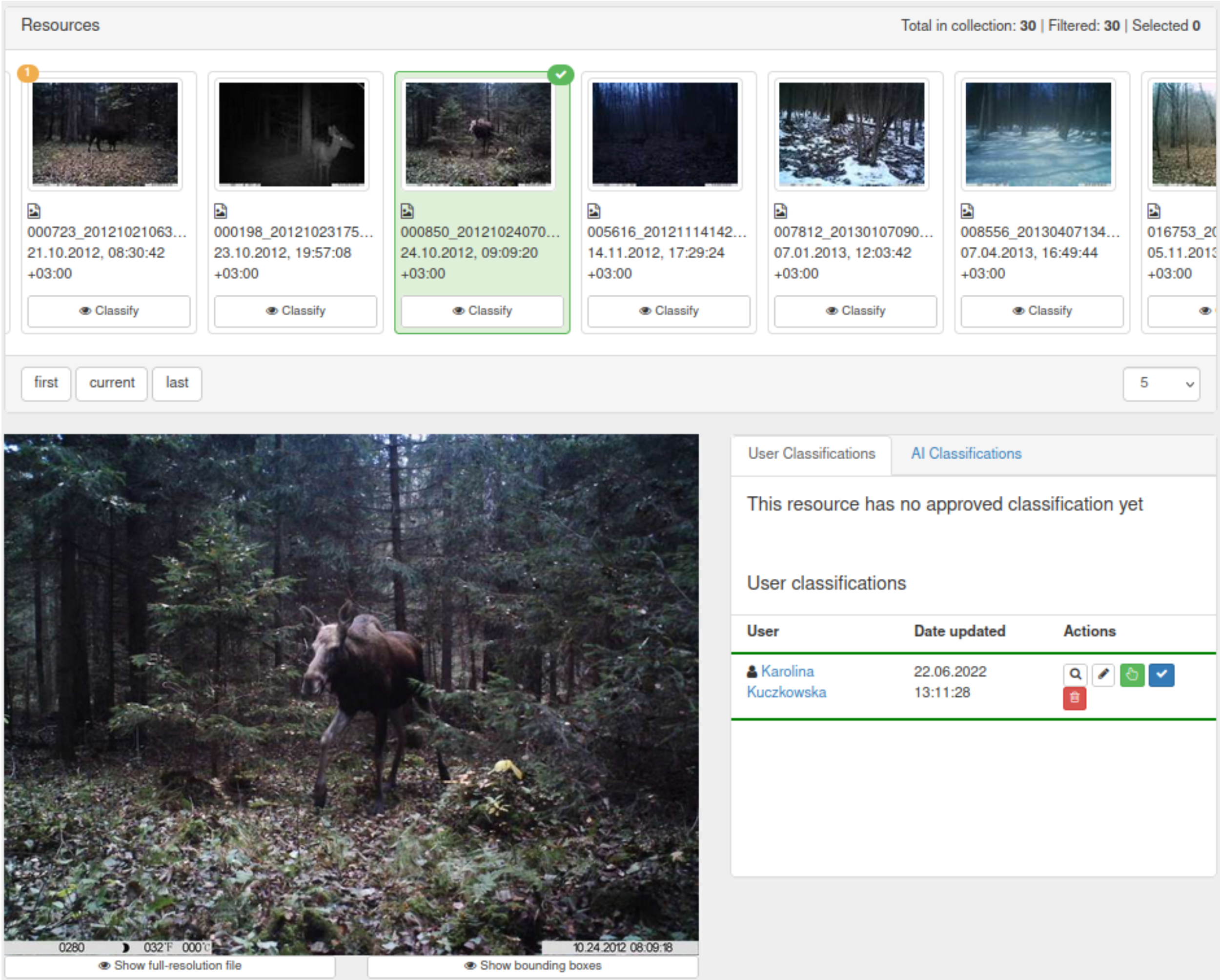
The *Sequence* is now classified and you can go to the next *Sequence(by clicking on the orange sequence number. Follow the steps above until you have classified all resources in a *Deployment*.

Note: FOR GROUPS OF ANIMALS: If there is one image in which all animals in the group are present, please classify the sequence per picture. In this way we can estimate retention time of individuals (especially with groups <5). With big groups, or groups were animals are never all in the picture, classify the whole group in one sequence. This ensures that we have the right number of animals passing in the analysis by just selecting the maximum per sequence. Again, make sure that empty pictures that are part of the sequence are classified as empty.

Note: If a person is in the frame with a dog, add **two observations**: one with observation_type = human and species *Homo sapiens* and the other with observation_type = animal and species *Canis familiaris*.

5.4.3 3. Adding bounding boxes for training dataset for Machine Learning

A new function in Trapper is that we can now add bounding boxes to each classification. These delimit where the object (an animal, human or vehicle) is and will aid in the training of Machine Learning algorithms for automatic species identification. To add bounding boxes to your classification, you need to go to the 'classic' classification view by clicking on 'Details' underneath a thumbnail.



Total in collection: **30** | Filtered: **30** | Selected **0**

000723_20121021063…
21.10.2012, 08:30:42 +03:00

000198_20121023175…
23.10.2012, 19:57:08 +03:00

000850_20121024070…
24.10.2012, 09:09:20 +03:00

005616_20121114142…
14.11.2012, 17:29:24 +03:00

007812_20130107090…
07.01.2013, 12:03:42 +03:00

008556_20130407134…
07.04.2013, 16:49:44 +03:00

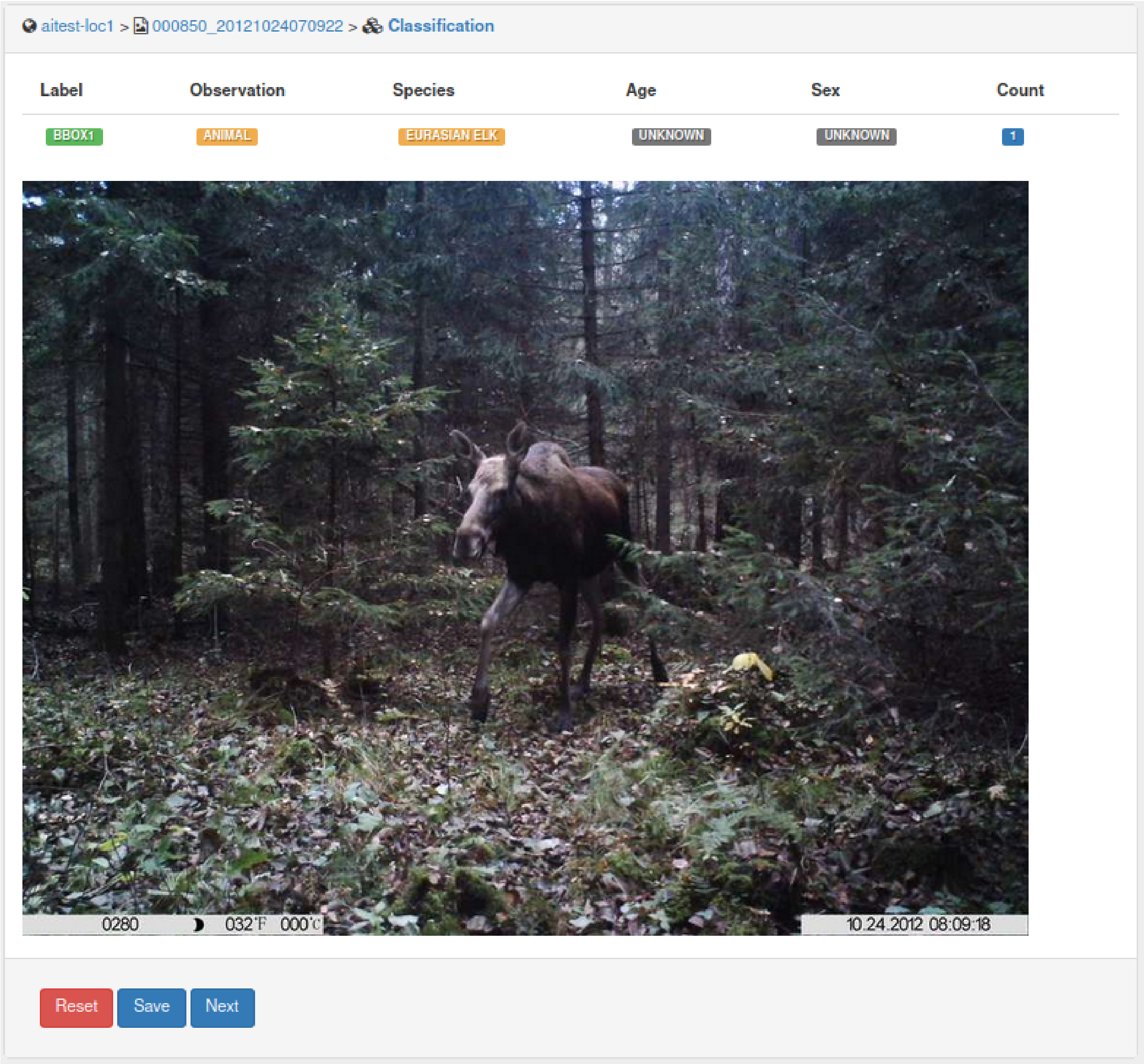
016753_20
05.11.2013
+03:00

This resource has no approved classification yet

User classifications

User	Date updated	Actions
Karolina Kuczkowska	22.06.2022 13:11:28	

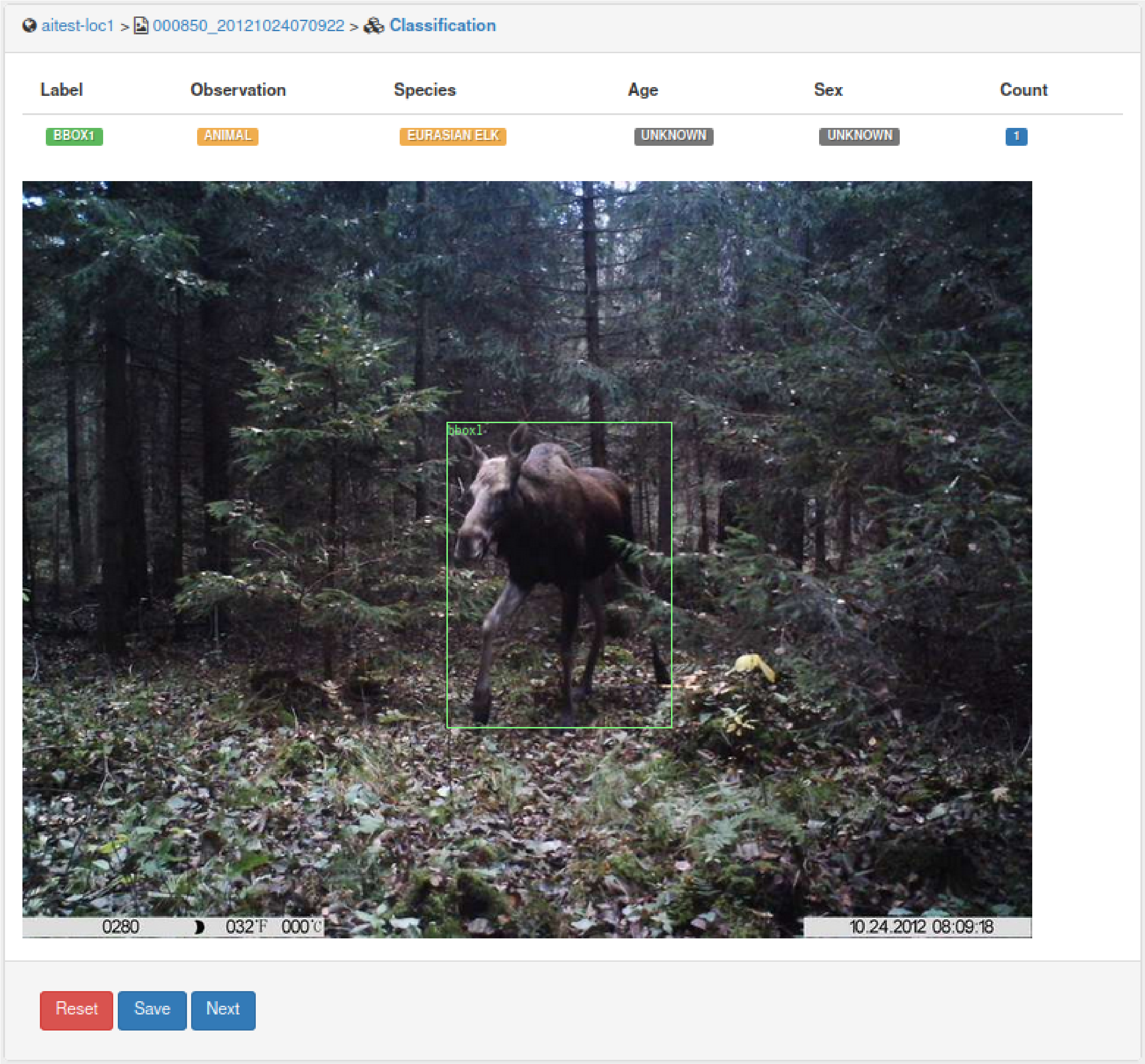
You can now add a bounding box by clicking on the green hand symbol on the user classification. This will lead you to the new bounding box view.



aitest-loc1 > 000850_20121024070922 > Classification

Label	Observation	Species	Age	Sex	Count
BBOX1	ANIMAL	EURASIAN ELK	UNKNOWN	UNKNOWN	1

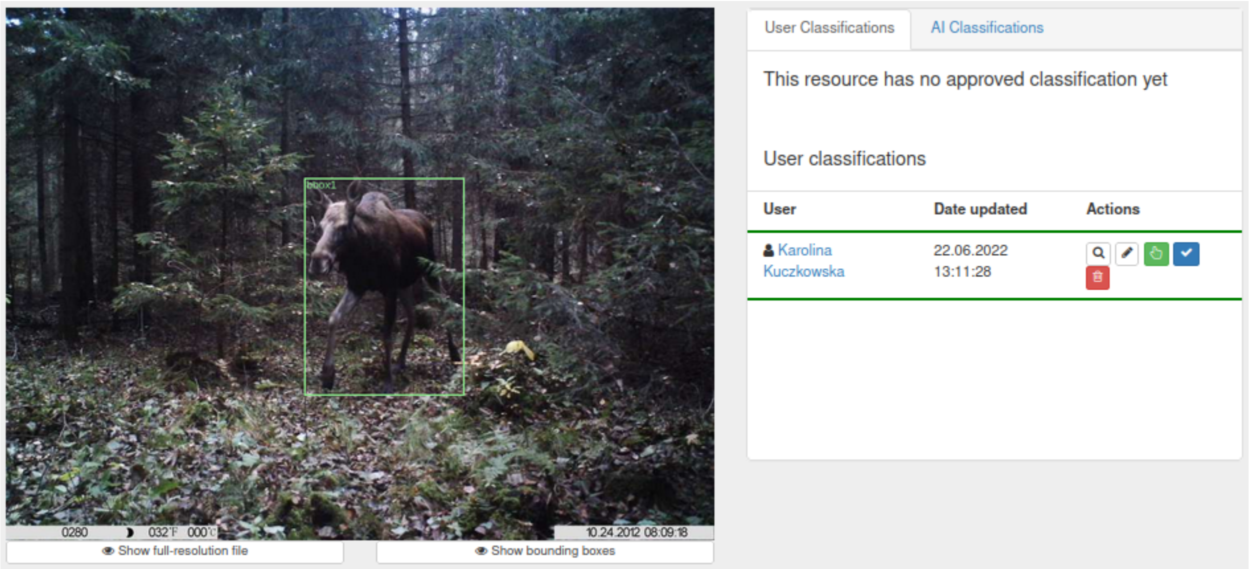
Here you can draw a box around the object. Please do this as tight around the animal as possible. You can then specify which dynamic attribute field should be linked to this bounding box.



aitest-loc1 > 000850_20121024070922 > Classification

Label	Observation	Species	Age	Sex	Count
BBOX1	ANIMAL	EURASIAN ELK	UNKNOWN	UNKNOWN	1

Click ‘Save’ to add the bounding box to the classification. You can now also see it in the ‘classic’ classification screen.



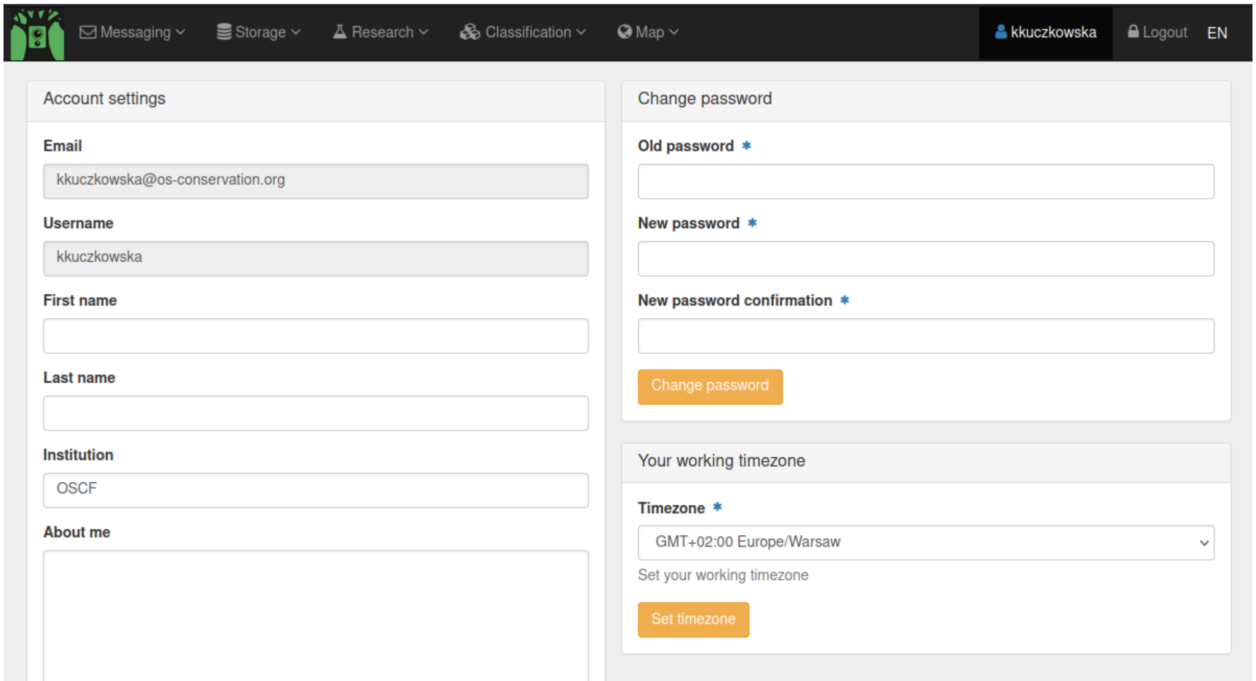
This resource has no approved classification yet

User classifications

User	Date updated	Actions
Karolina Kuczkowska	22.06.2022 13:11:28	

5.5 Timezone behaviour in TRAPPER

The default timezone for TRAPPER app is **Europe/Warsaw**. You can set your working timezone in your User Profile settings tab (click your username in top right corner to access this view).



Account settings

Email: kkuczkowska@os-conservation.org

Username: kkuczkowska

First name:

Last name:

Institution: OSCF

About me:

Change password

Old password *

New password *

New password confirmation *

Your working timezone

Timezone *: GMT+02:00 Europe/Warsaw

Set your working timezone

Your working timezone will be used:

- to correctly display system notifications (e.g. asynchronous task completion times on your dashboard),
- as a default timezone value in forms.

5.5.1 Daylight Savings Time

While not recommended, it is possible to bypass default behaviour regarding DST (Daylight Saving Time).

If your camera trap at a given location *does not use* DST (i.e. does not move its timer one hour forward in the spring and one hour back in the fall), you should mark its `ignore_DST` flag `True`. With that setting, all timestamps in recorded metadata from that location (`Deployment.start_date`, `Deployment.end_date` and `Resource.date_recorded`) will be processed ignoring DST (it would be considered in processing timestamps otherwise).

Note: Whenever this documentation mentions two *Locations* having **the same timezone specification**, it means both the `timezone` field and the `ignore_DST` flag having the same value, as only both values together correctly describe the logic of processing timestamps (i.e. datetime objects) in TRAPPER.

5.5.2 Timezone is set for Location

Location objects have a *timezone* field and an `ignore_DST` flag. You have to specify them in the form when adding location in one of the following ways:

- using the map view with the form for adding a single location,
- importing locations from CSV or GPX file (bulk import),
- creating locations automatically using deployments bulk import.

It is very important to correctly set timezone information for every *Location*, because this data is used to process timestamps in all *Deployments* and *Resources* connected to given *Location*.

5.5.3 Timezone behaviour during CSV import

When importing *Deployments* or *Locations* from CSV file, you have to provide their `timezone` and `ignore_DST` values in the import forms.

This means that you can only import locations from the same `timezone` and with the same `ignore_DST` flag value. Data that do not meet this condition has to be imported using separate CSV files.

When importing *Locations*, values from the form will be set for all imported objects.

When importing *Deployments*, they can be linked to different locations, but all locations need to have the same timezone specification. If locations are being created, their timezone info will be set to data from the form.

If *Deployments* are uploaded to existing *Locations*, the file will be validated to make sure all locations referenced in the data have the same timezone specification as in the form.

This also applies to uploading *Collections* - in trapper-client, *Collections* can only be created from *Resources* linked to *Locations* with the same timezone specification. The timezone specification (as specified in the trapper-client settings screen) is provided in the YAML file describing given *Collection*. When uploading a *Collection*, the validation process is performed to check that provided timezone matches *Locations* in the database as well.

CHAPTER 6

Technology used - software components

6.1 Backend

The following is the list of the most important software components that have been used to design and implement Trapper:

- Ubuntu A base operating system under which Trapper was developed and tested.
- PostgreSQL The open source and industry standard relational database management system (RDBMS) with PostGIS as its spatial extension.
- nginx nginx [engine x] is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP proxy server.
- Gunicorn This pure-python application is used to serve Trapper project.
- Supervisor This application is used to control all project related external applications like celery.
- Celery This python application is used to run various tasks in asnynchronous mode allowing Trapper to work faster and more efficient. The celery is used to generate thumbnails from large files or videos, or to process uploaded collections by users.
- Django The core component of Trapper, a high-level Python web framework maintained by the Django Software Foundation. Django framework simplifies and significantly speeds up the creation of complex, database-driven websites and emphasizes reusability and pluggability of their components. Additionally, a variety of 3rd party, open source django applications have been used to develop Trapper. For a complete list of these applications see **‘this file <>‘_**
- uMAP uMap lets you create maps with OpenStreetMap layers in a minute and embed them in your site. It uses django-leaflet-storage and Leaflet.Storage, built on top of Django and Leaflet.

6.2 Frontend

Trapper’s front-end is developed based on three independent solutions:

1) HTML (version 5) templates and their CSS styles (powered by SASS)
2) set of scripts written in pure JavaScript (ECMAscript 5).
3) all the external libraries and frameworks included in the project:

- Twitter Bootstrap Bootstrap Sass Official which is official SASS version of Twitter Bootstrap. This library provides a set of HTML components and CSS styles used for Trapper scaffold creation.
- Font Awesome Font Awesome - webfont of vector icons used in the project.
- Angular JS Angular JS all the grids/tables including their filters has been build on top of this Google's framework
- Angular Cookies Official angular module for cookies management.
- Angular Sanitize Official angular module which improves angular templates data binding.
- Moment Extremaly powerful library for date parsing & manipulation.
- Select2 Complete solution that extends default HTML select controls.
- Select2 Bootstrap CSS CSS styles for Select2 so it fits Twitter Bootstrap feel & look.
- Bootstrap WYSIHTML5 Javascript Plugin which brings WYSIWYG text editor to the table.
- Bootstrap Datepicker http://eternicode.github.io/bootstrap-datepicker/ Javascript Widget - simple datepicker.
- Jquery Timepicker Jquery Widget which is just a timepicker.
- Bootstrap Datetimepicker Javascript Widget that combines both time and date picker.
- Video JS This library extends standard HTML5 video players.
- Video JS Rangeslider Video JS plugin that allows to set and get video sequences.

CHAPTER 7

License

CHAPTER 8

Indices and tables

- genindex
- modindex
- search