GSi Documentation

Release 1.0

GSi

January 11, 2017

CHAPTER 1

Contents:

1.1 Used technologies

List of libraries are buildout.cfg file located in the root of the project:

```
$ /home/gsi/sites/gsi_website
```

1.1.1 Backend

Python / Django 1.8

Database – PostgreSQL

1.1.2 Frontend

HTML, CSS, JavaScript, jQuery, Bootstrap 3.

1.2 Designation of graphical icons

Fig. 1.1: Edit

Fig. 1.2: Delete

Fig. 1.3: Copy

Fig. 1.4: Details

1.3 Frontend

Gsi frontend project consists of site header and left sidebar. In the left corner of site header it is logo of *"Gsi"* project with link to the home page. On the right is the custom settings menu.

1.3.1 Authorization

Page for authorization when entering the GSi system. To enter two types of users used GSi system: *"Customer"* and *"Admin"*. For users is *"Admin"* access to all the site pages. For the user *"Customer"* only page *"Customer Section"*.

Login (indy4.epcc.ed.ac.uk)

To log in GSi system need a login and password that the user specify at registration.

Registration (indy4.epcc.ed.ac.uk/register/)

If the user enters the first time hi is necessary to register (*"Sign Up"*).

Restore password (indy4.epcc.ed.ac.uk/reset/password_reset/)

If a user forgets his password, he can restore it (*"Forgot you password"*):

1.3.2 The Left Sidebar

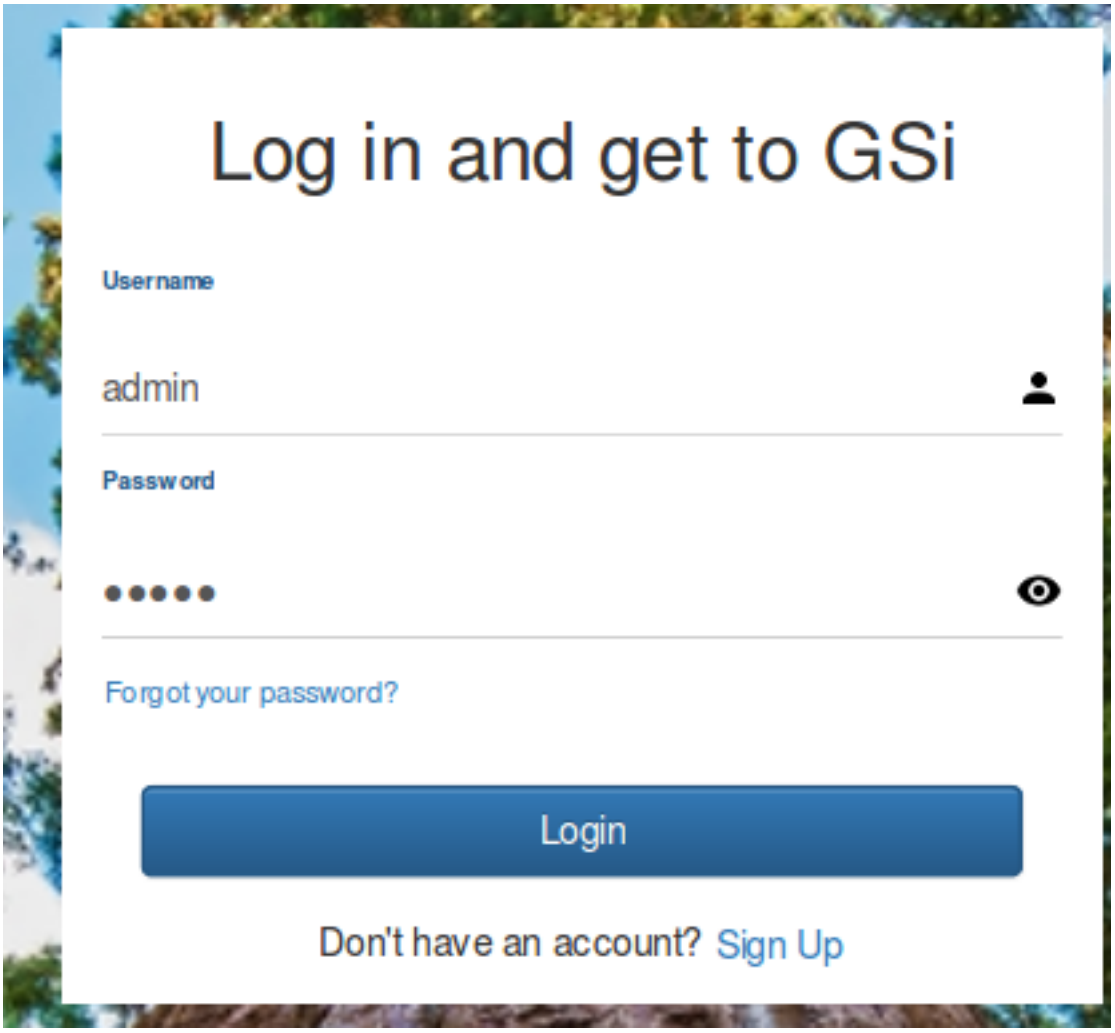
On the left sidebar are menu items for creating and managing data.

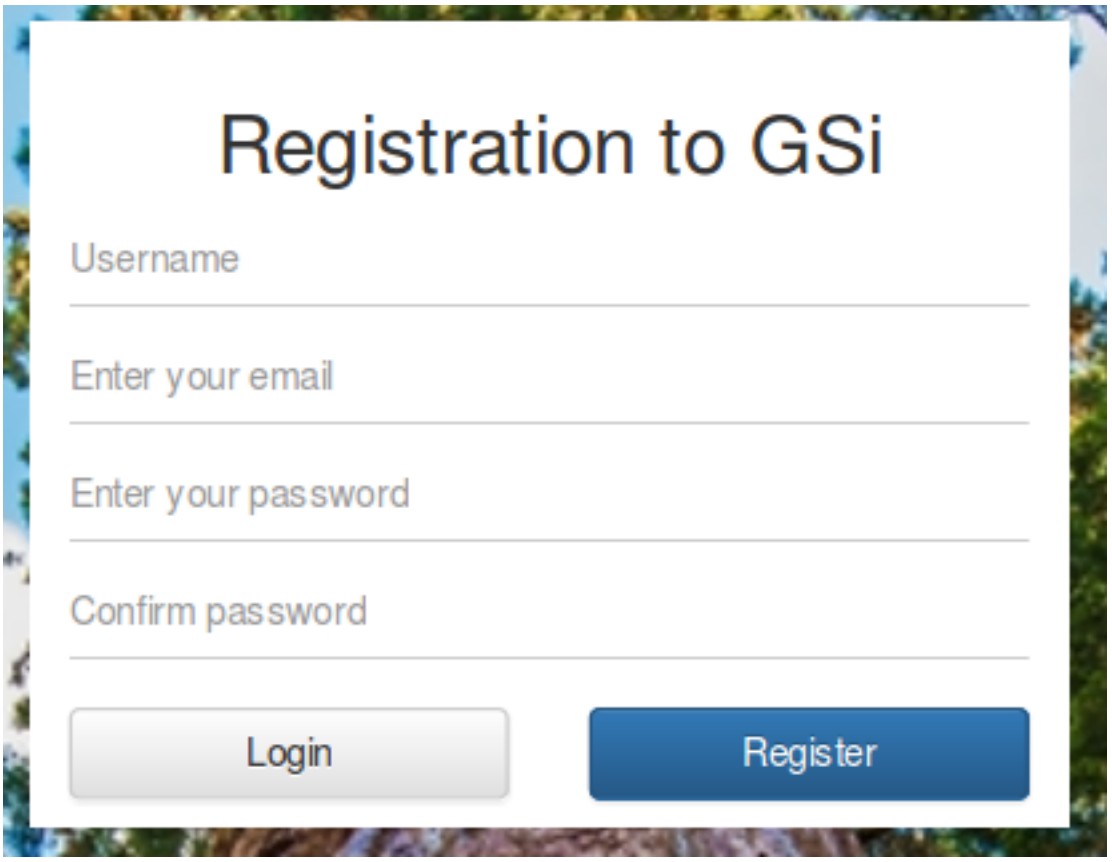
Page "Overview" (indy4.epcc.ed.ac.uk)

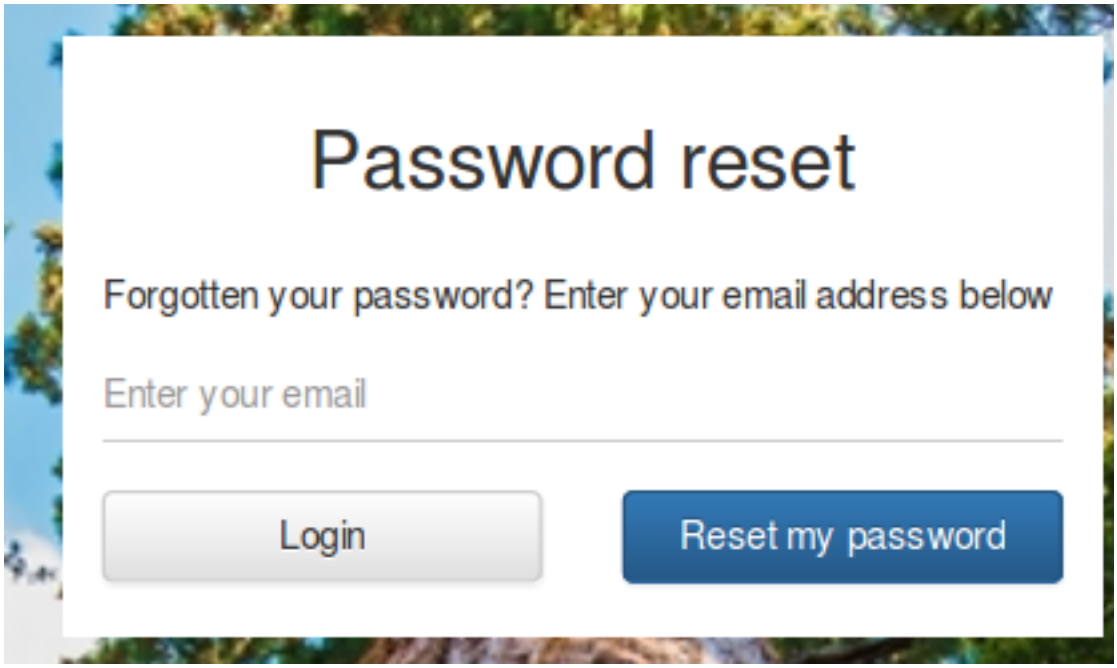
Home page. After login the user lands on the home page.

Page "Setup New Run" (indy4.epcc.ed.ac.uk/run/setup/)

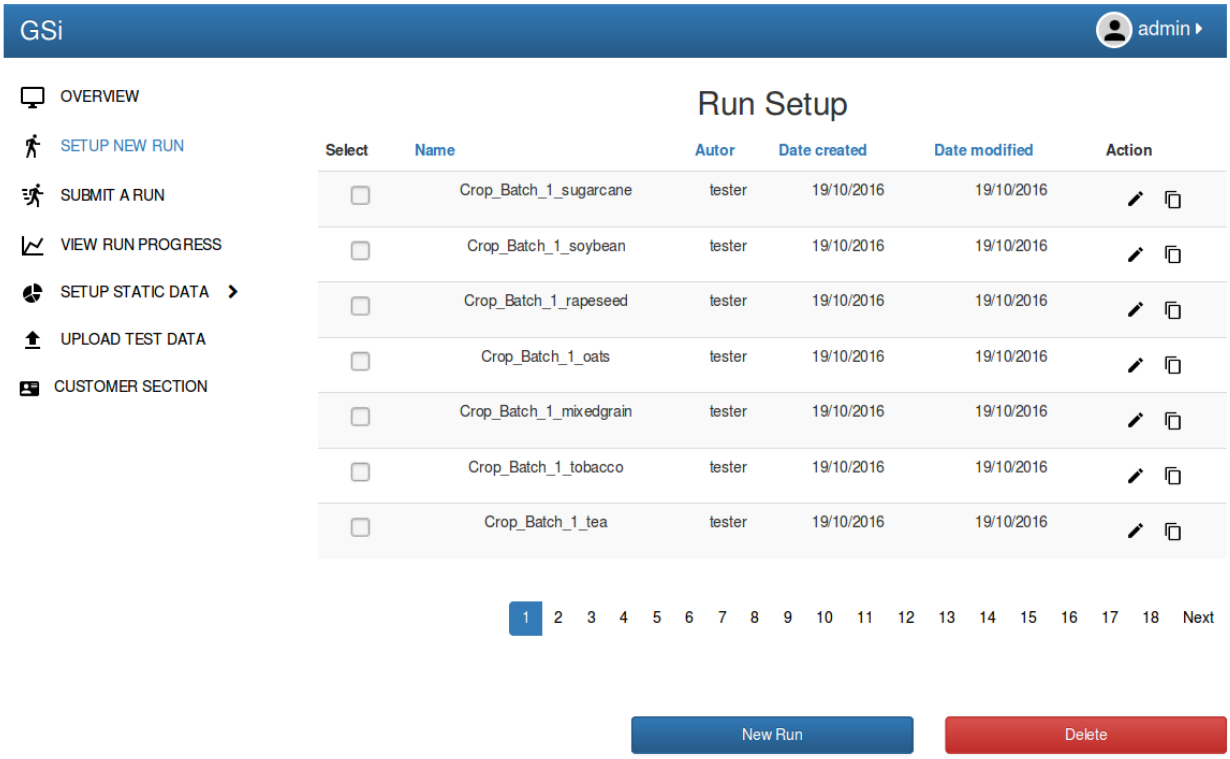
It is intended to create, delete and edit new Runs. Data can be sorted by *"Name"*, *"Author"*, *"Date"* of creation and Date modified. In the *"Action"* field you can change the selected Run or create a copy of it.


Log in and get to GSi
Username
admin
Password
Forgot your password?
Login
Don't have an account? Sign Up


Registration to GSi
Username
Enter your email
Enter your password
Confirm password
Login
Register

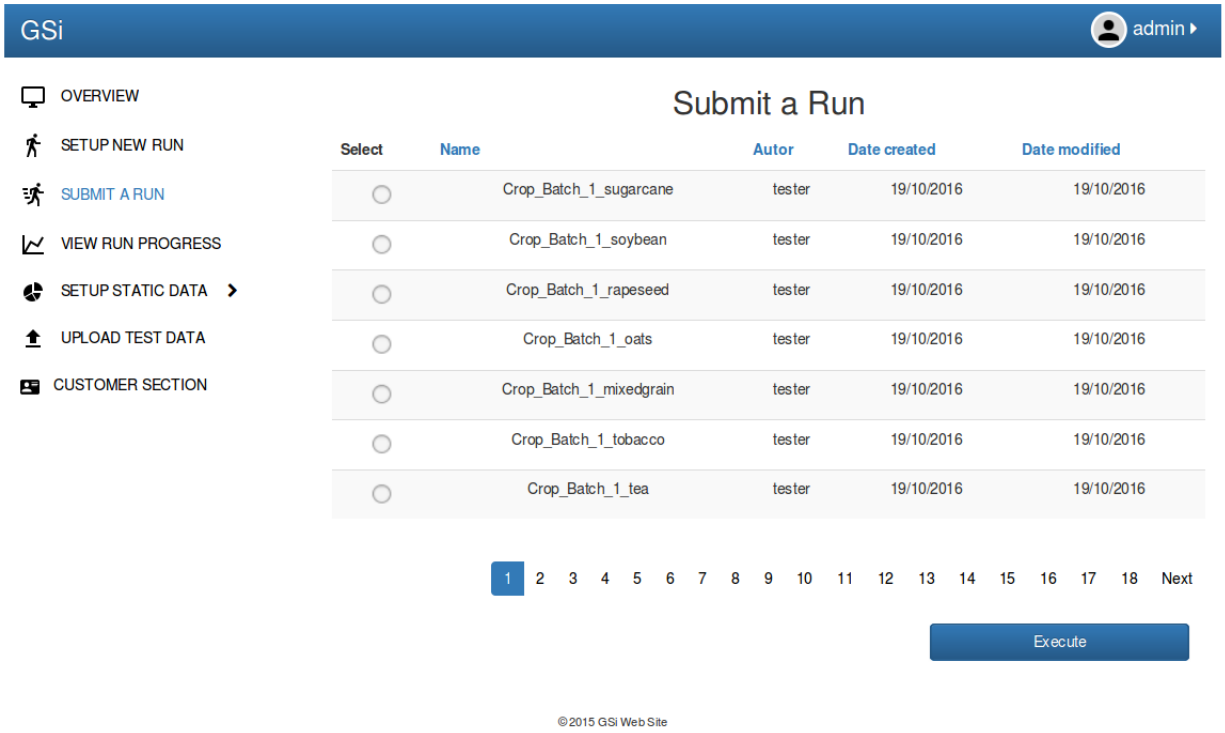

Password reset
Forgotten your password? Enter your email address below
Enter your email
Login
Reset my password


GSi
admin
OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
UPLOAD TEST DATA
CUSTOMER SECTION
©2015 GSi Web Site


GSi
admin
OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
UPLOAD TEST DATA
CUSTOMER SECTION
Run Setup
Select
Name
Autor
Date created
Date modified
Action
Crop_Batch_1_sugarcane tester 19/10/2016 19/10/2016
Crop_Batch_1_soybean tester 19/10/2016 19/10/2016
Crop_Batch_1_rapeseed tester 19/10/2016 19/10/2016
Crop_Batch_1_oats tester 19/10/2016 19/10/2016
Crop_Batch_1_mixedgrain tester 19/10/2016 19/10/2016
Crop_Batch_1_tobacco tester 19/10/2016 19/10/2016
Crop_Batch_1_tea tester 19/10/2016 19/10/2016
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 Next
New Run
Delete

Page “Submit a Run” (indy4.epcc.ed.ac.uk/run/submit/)

It is intended to execute the created runs.



Page “Run Progress” (indy4.epcc.ed.ac.uk/run/progress/)

It is intended to show a process execute Runs. It is possible to remove.

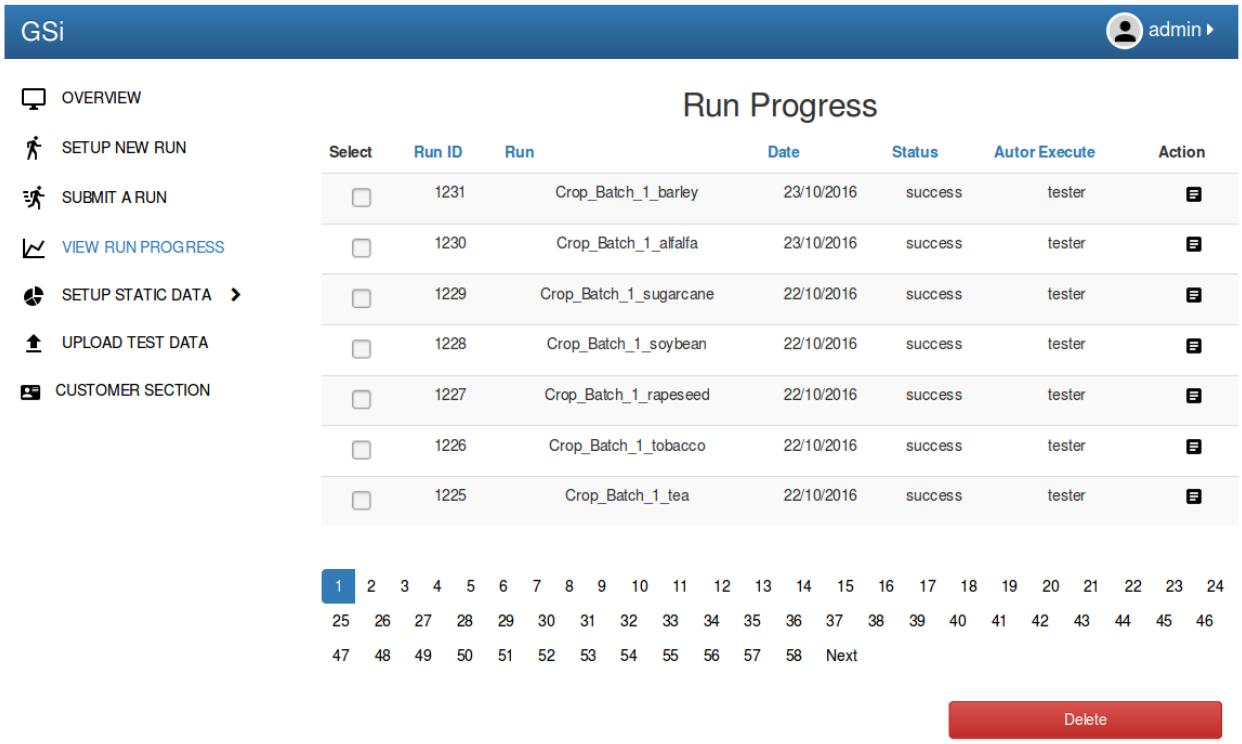
In the *“Action”* has a *“Details”* button to navigate to a new page view logs of work/execution.
You can see the process of the work and a detailed view logs:

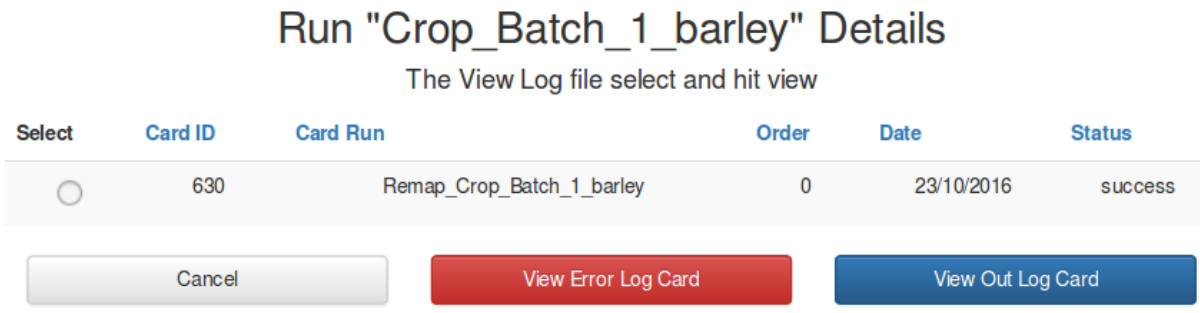
Page-Section “Setup Static Data”

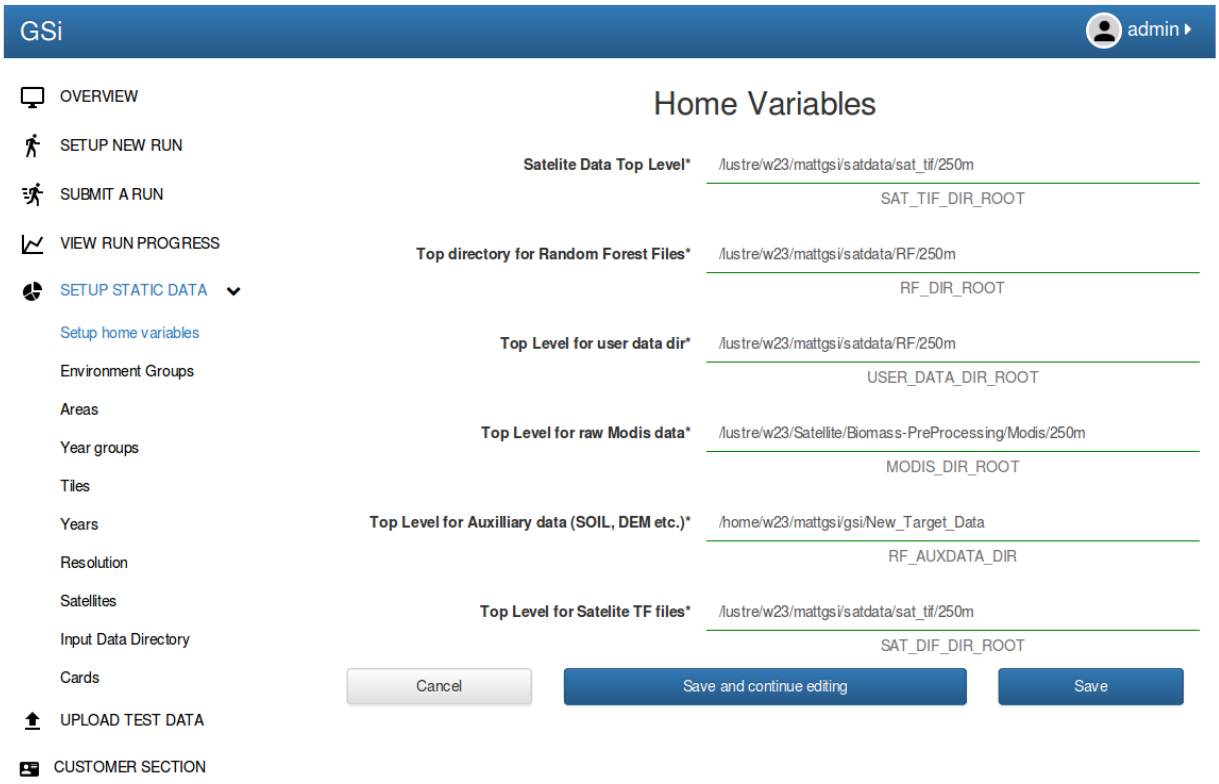
It is intended to set the parameters required for work.

Page “Home Variables” (indy4.epcc.ed.ac.uk/run/home-variable/setup/)

Setting paths for home variables.


GSi
admin
OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
UPLOAD TEST DATA
CUSTOMER SECTION
Run Progress
Select
Run ID
Run
Date
Status
Autor Execute
Action
1231 Crop_Batch_1_barley 23/10/2016 success tester
1230 Crop_Batch_1_alfalfa 23/10/2016 success tester
1229 Crop_Batch_1_sugarcane 22/10/2016 success tester
1228 Crop_Batch_1_soybean 22/10/2016 success tester
1227 Crop_Batch_1_rapeseed 22/10/2016 success tester
1226 Crop_Batch_1_tobacco 22/10/2016 success tester
1225 Crop_Batch_1_tea 22/10/2016 success tester
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 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 Next
Delete


Run "Crop_Batch_1_barley" Details
The View Log file select and hit view
Select
Card ID
Card Run
Order
Date
Status
630 Remap_Crop_Batch_1_barley 0 23/10/2016 success
Cancel
View Error Log Card
View Out Log Card



Page "Environment Groups" (indy4.epcc.ed.ac.uk/run/environment-groups/)

Creating, editing and deleting data of the *"Environment Groups"*. Data from this menu are used to create new Runs.

Editing the *"Environment Groups"*:

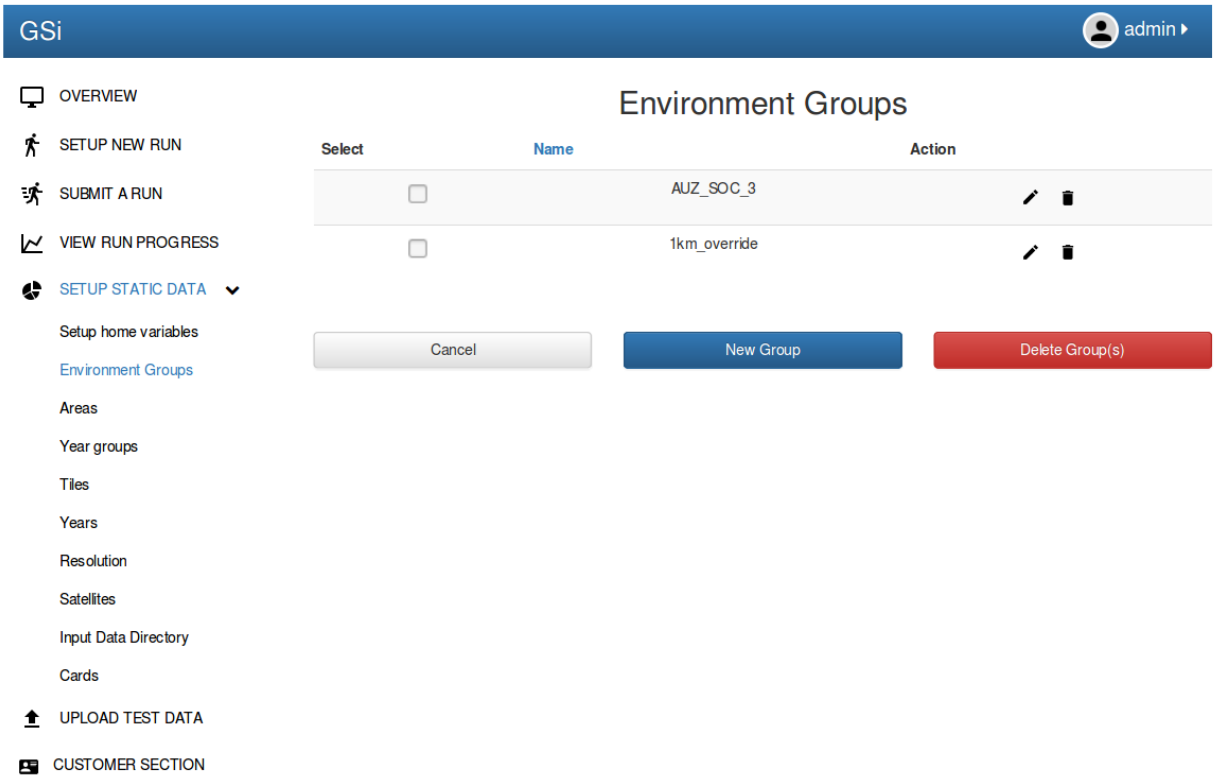
Page "Areas" (indy4.epcc.ed.ac.uk/areas/list/)

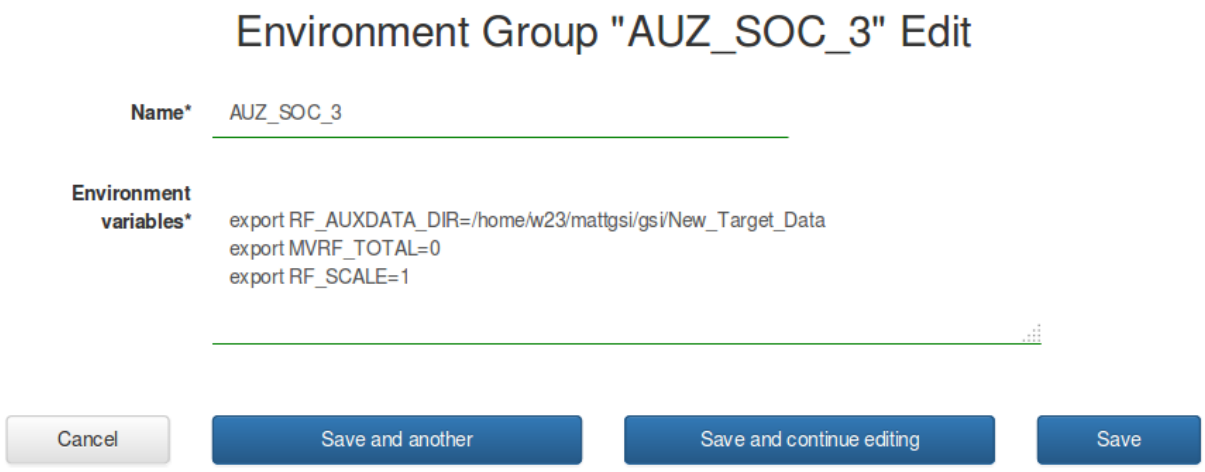
Creating, editing and deleting the areas.

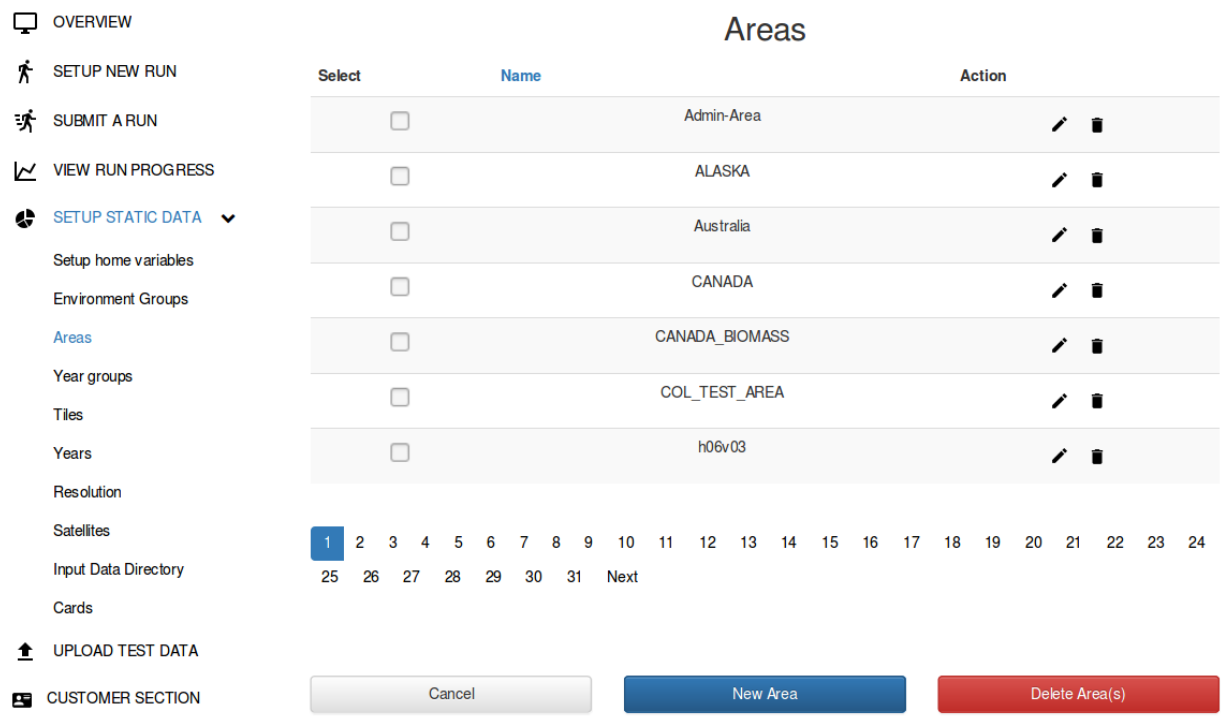
When editing, you can select one or more tiles and clicking *"Add"* button to add the selected area. When you delete a select one or more tiles tiles *"Delete"* to remove from the selected area

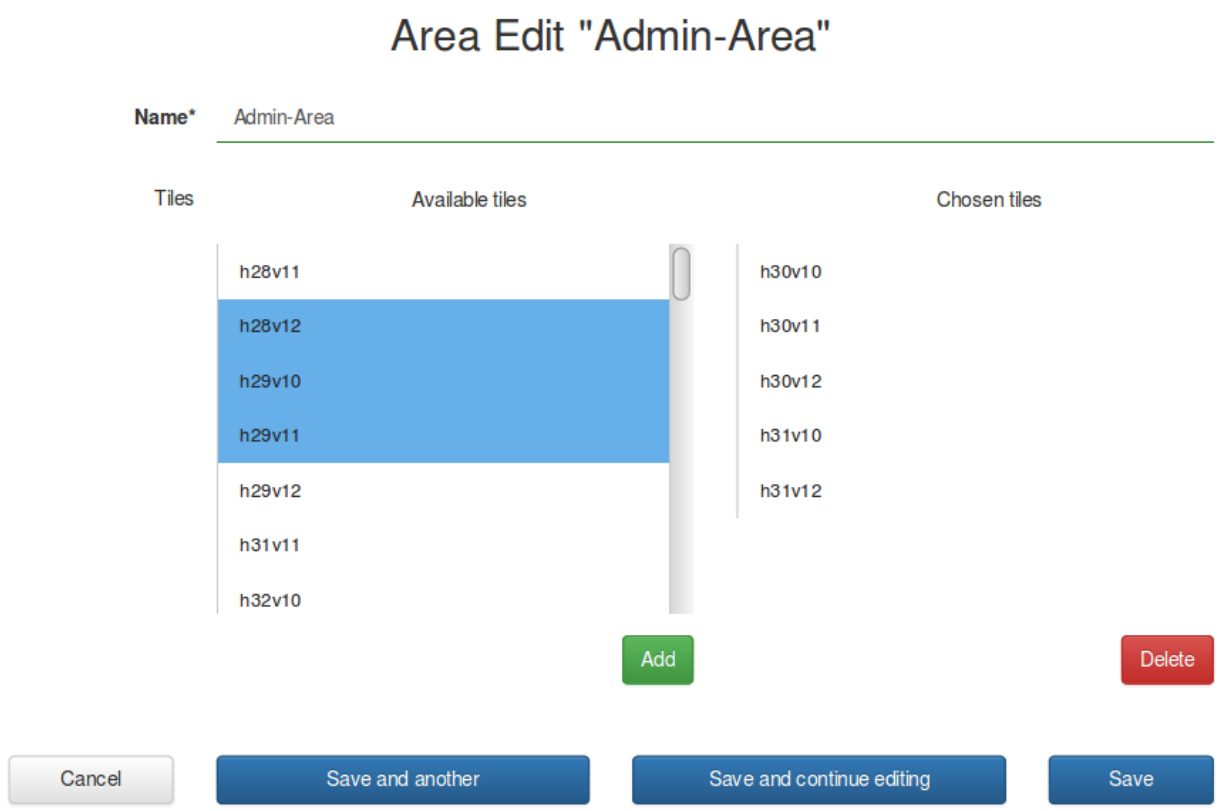
Page "Years Groups" (indy4.epcc.ed.ac.uk/years-group/list/)

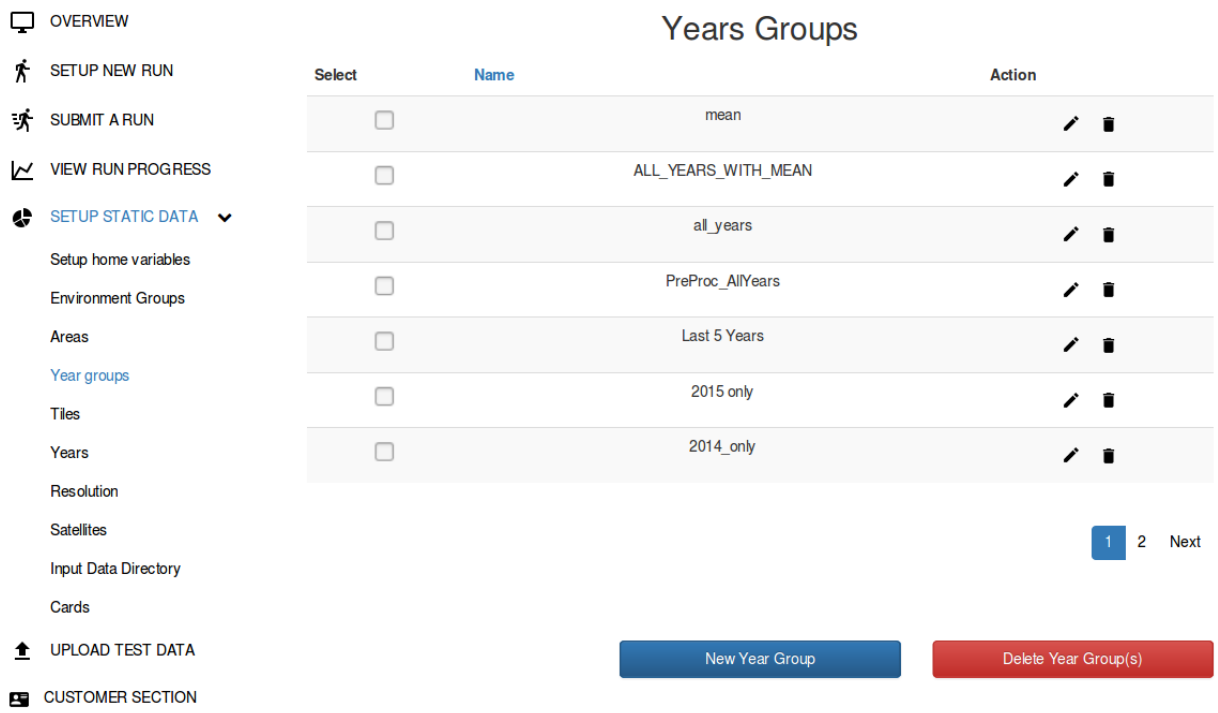
Creating, editing and deleting items "Years Groups".


GSi
admin
OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Environment Groups
Select
Name
Action
AUZ_SOC_3
1km_override
Cancel
New Group
Delete Group(s)


Environment Group "AUZ_SOC_3" Edit
Name*
AUZ_SOC_3
Environment variables*
export RF_AUXDATA_DIR=/home/w23/mattgsi/gsi/New_Target_Data
export MVRF_TOTAL=0
export RF_SCALE=1
Cancel
Save and another
Save and continue editing
Save


OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Areas
Select
Name
Action
Admin-Area
ALASKA
Australia
CANADA
CANADA_BIOMASS
COL_TEST_AREA
h06v03
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 Next
Cancel
New Area
Delete Area(s)


Area Edit "Admin-Area"
Name*
Admin-Area
Tiles
Available tiles
Chosen tiles
h28v11
h28v12
h29v10
h29v11
h29v12
h31v11
h32v10
h30v10
h30v11
h30v12
h31v10
h31v12
Add
Delete
Cancel
Save and another
Save and continue editing
Save



When editing, you can select one or more years, and then clicking *"Add"* button to add the selected group. When you delete a select one or more years, and clicking *"Delete"* to remove from the selected group

Page "Tiles" (indy4.epcc.ed.ac.uk/tiles/list/)

Creation and delete of tiles. In this menu, you can only create or delete tiles. These tiles are used to create of areas.

Page "Years" (indy4.epcc.ed.ac.uk/years/list/)

Create, edit and delete years. Data from this menu is used when creating the group of years.

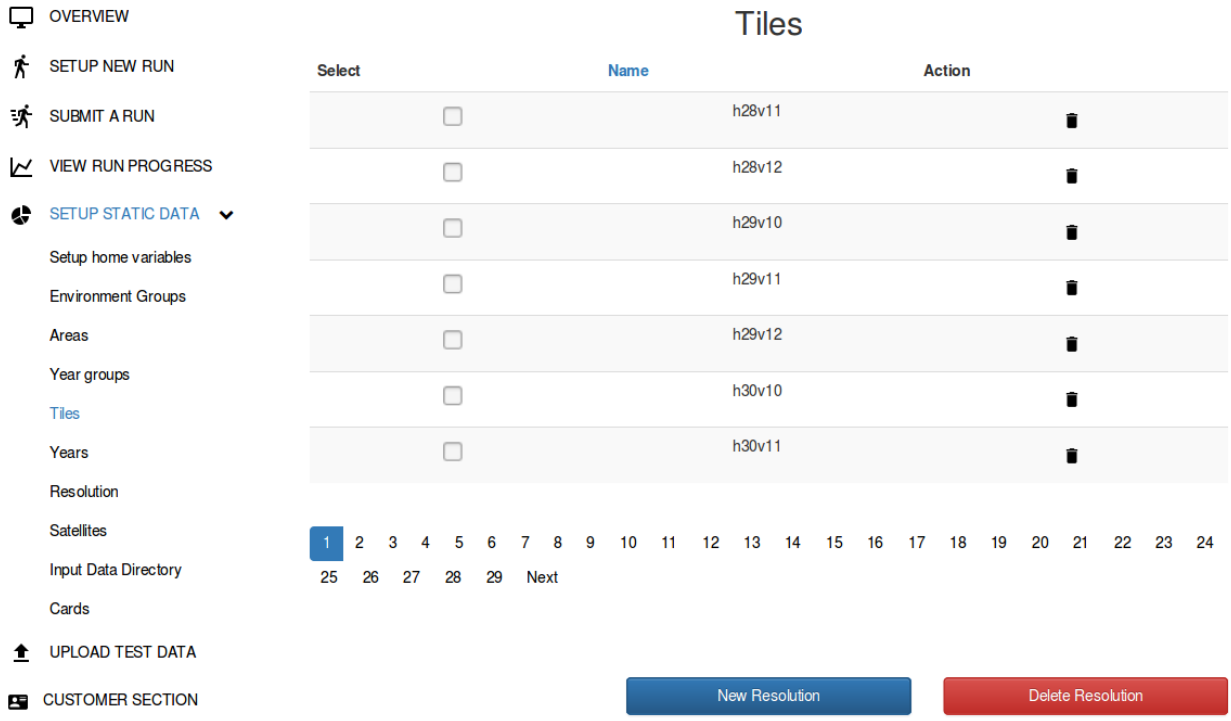
Page "Resolutions" (indy4.epcc.ed.ac.uk/resolution/list/)

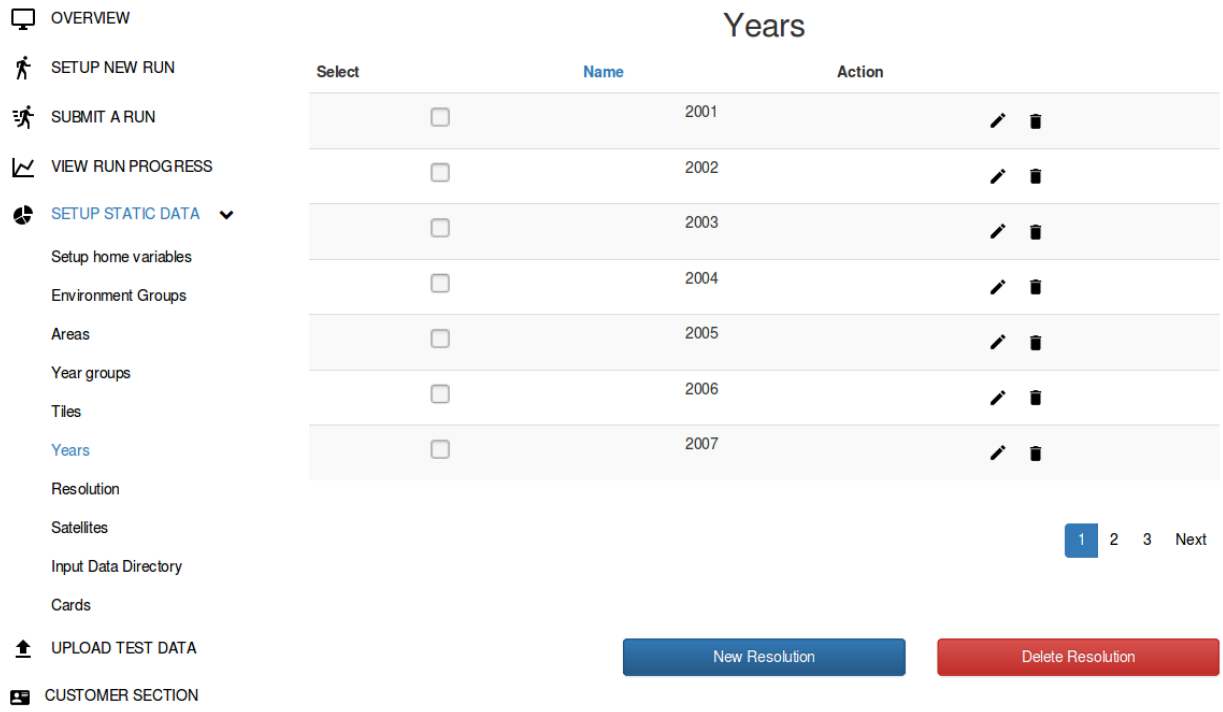
Creating, editing and deleting of resolutions. Data from this menu is used when creating a new runs.

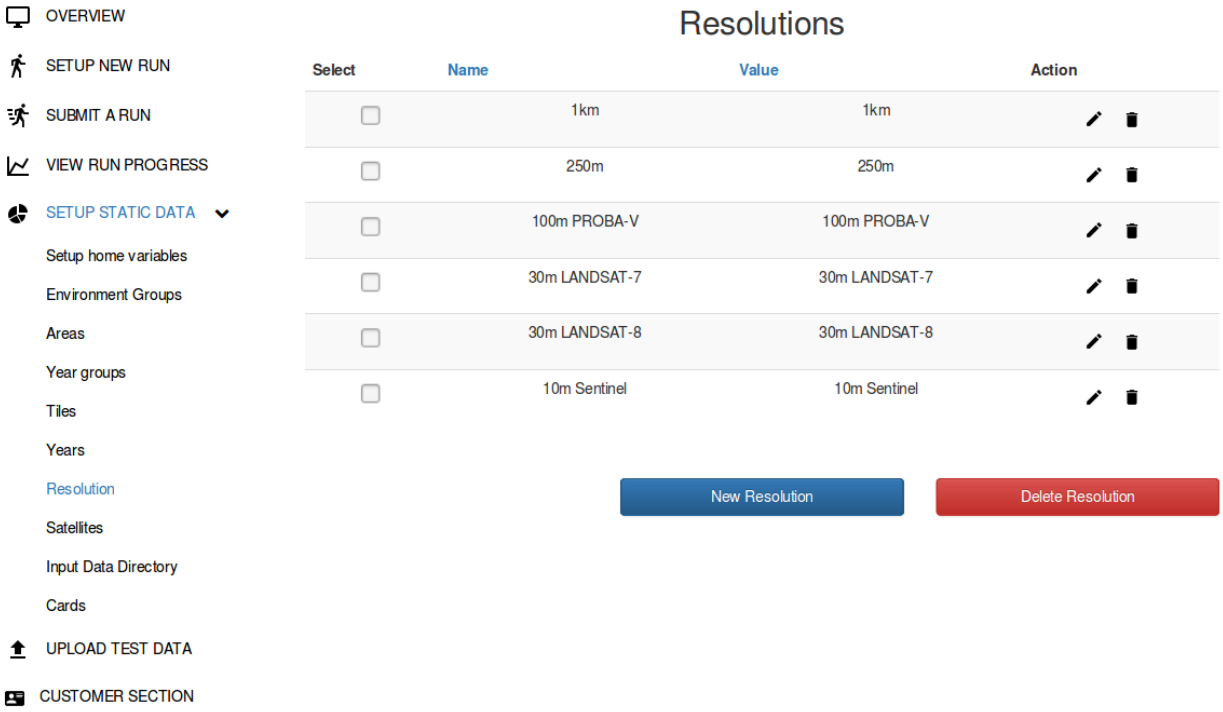
Page "Satellites" (indy4.epcc.ed.ac.uk/satellites/list/)

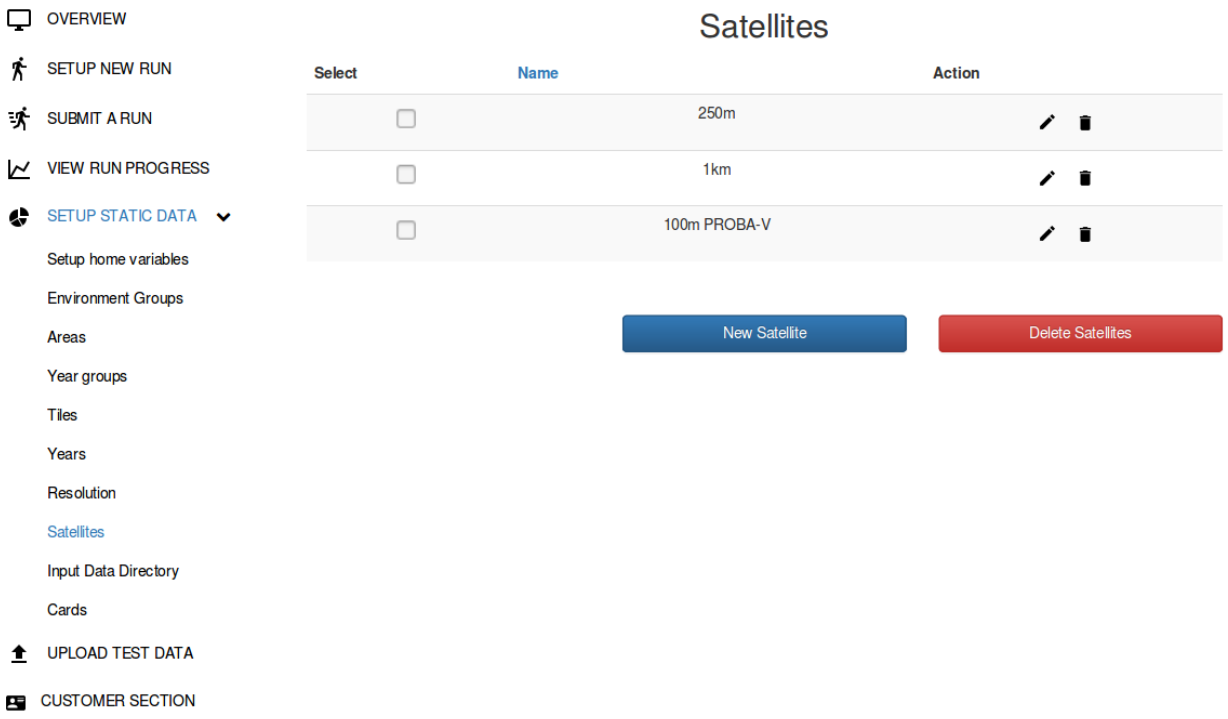
Creating, editing and deleting of satellites.


YearGroup Edit "ALL_YEARS_WITH_MEAN"
Name*
ALL_YEARS_WITH_MEAN
Years
Available years
Chosen years
2016
2002
2003
2004
2005
2006
2007
2008
Add
Delete
Cancel
Save and another
Save and continue editing
Save


OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Tiles
Select
Name
Action
h28v11
h28v12
h29v10
h29v11
h29v12
h30v10
h30v11
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 Next
New Resolution
Delete Resolution


OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Years
Select
Name
Action
2001
2002
2003
2004
2005
2006
2007
1
2
3
Next
New Resolution
Delete Resolution


OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Resolutions
Select
Name
Value
Action
1km
1km
250m
250m
100m PROBA-V
100m PROBA-V
30m LANDSAT-7
30m LANDSAT-7
30m LANDSAT-8
30m LANDSAT-8
10m Sentinel
10m Sentinel
New Resolution
Delete Resolution



Page "Input Data Directory" (indy4.epcc.ed.ac.uk/input-data-dirs/list/)

Creating paths to Input Data Directory. In this menu, you can only create or delete the data.

Page "Cards" (indy4.epcc.ed.ac.uk/cards/list/)

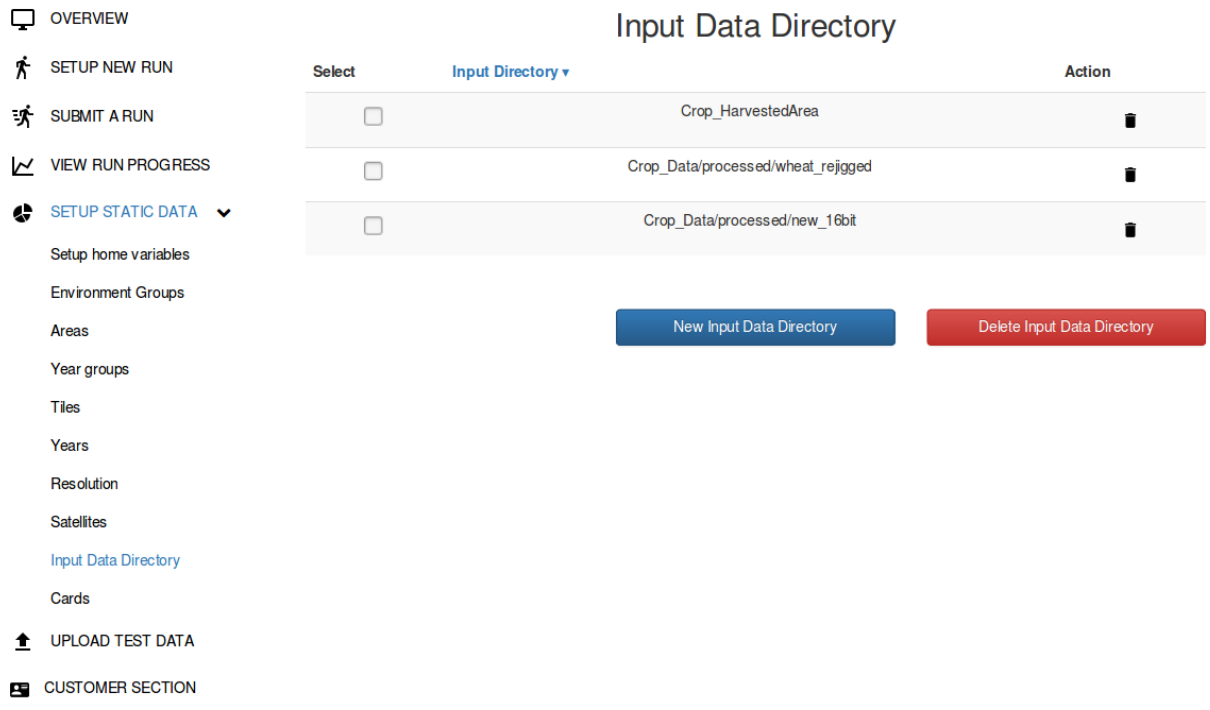
List all of the cards used in the creation of new runs. In this menu, only the removal of cards possible.

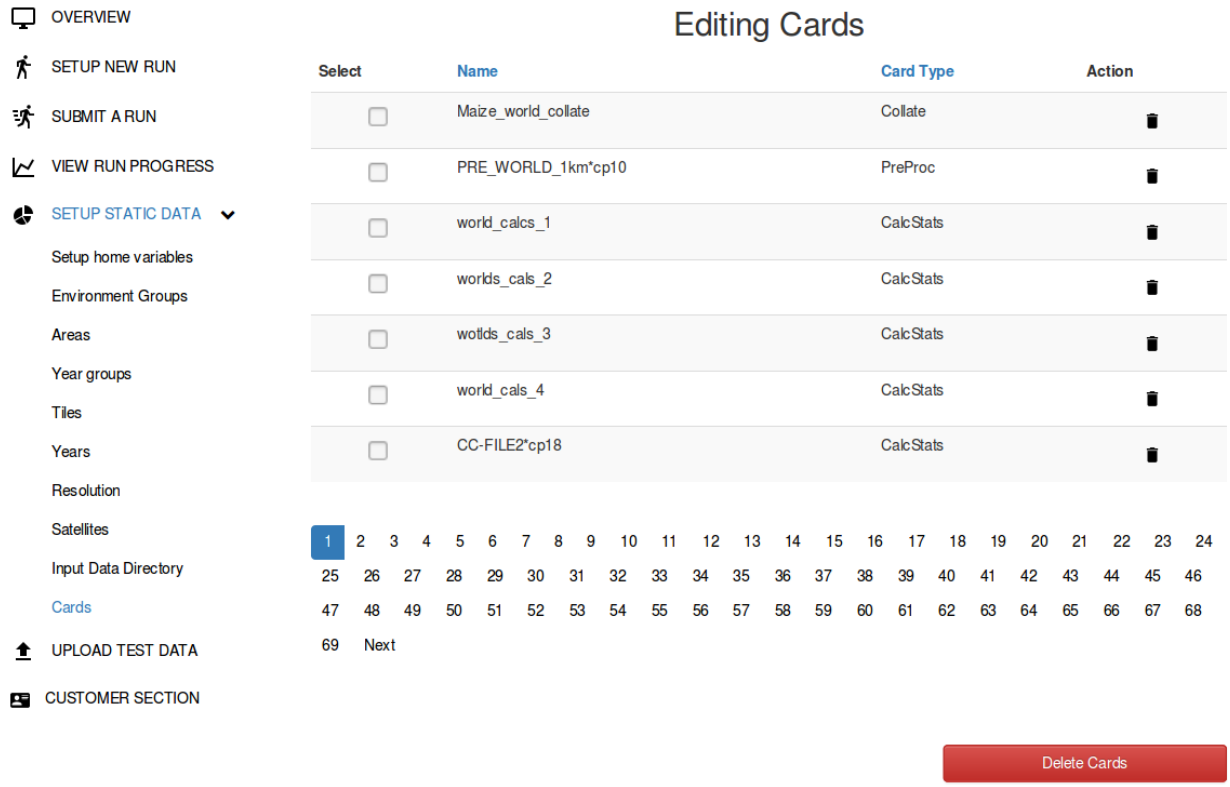
Page "Upload Test Data" (indy4.epcc.ed.ac.uk/upload-file/)

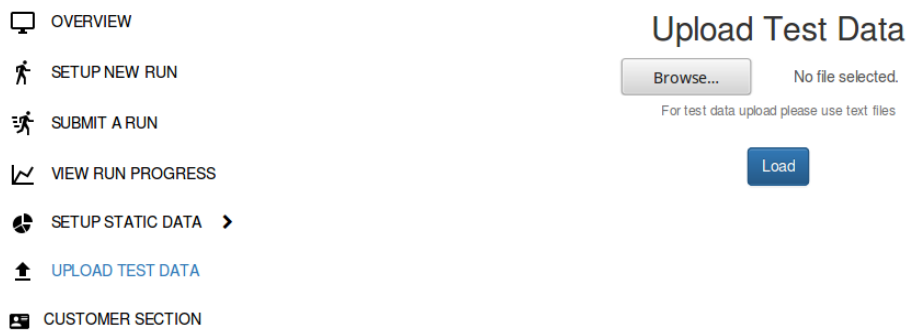
This menu for to download testing data to the server.

Page "Customer Section" (indy4.epcc.ed.ac.uk/customer/)

This section is created for the customer. In this menu the customer can get a map image with ready data. In the field *"Enter Lat"* and *"Enter Log"* to set the desired coordinates and get a map image for these coordinates. Right-click to set markers. By clicking on the button *"Delete the Markers"* to remove the creation of markers. To draw a new polygon. To do this, set the marker and click *"Draw the Polygon"*. Use the *"Delete the Polygon"* remove it or the button *"Show the Polygon on/off"* to show or hide from


OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Input Data Directory
Select
Input Directory
Action
Crop_HarvestedArea
Crop_Data/processed/wheat_rejigged
Crop_Data/processed/new_16bit
New Input Data Directory
Delete Input Data Directory


OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
Setup home variables
Environment Groups
Areas
Year groups
Tiles
Years
Resolution
Satellites
Input Data Directory
Cards
UPLOAD TEST DATA
CUSTOMER SECTION
Editing Cards
Select
Name
Card Type
Action
Maize_world_collate
Collate
PRE_WORLD_1km*cp10
PreProc
world_calcs_1
CalcStats
worlds_cals_2
CalcStats
wotlds_cals_3
CalcStats
world_cals_4
CalcStats
CC-FILE2*cp18
CalcStats
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 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 Next
Delete Cards



the map. In the fields *"Lat"* and *"Lon"* displaying the mouse position within the polygon. Use the *"Transparency"*, you can change the transparency of a polygon with ready data.

1.3.3 Creating of Cards

The Cards for the Runs creating in the menu *"SETUP NEW RUN"* when you creating a new Run or edit an existing one.

Add to created Run

Choose in the field *"Action"* the *"Edit"*:

Go to the form of the Run editing.

Push the button *"Edit Run Details"* and go to the editing form of the *"Card sequence"* the chosen Run.

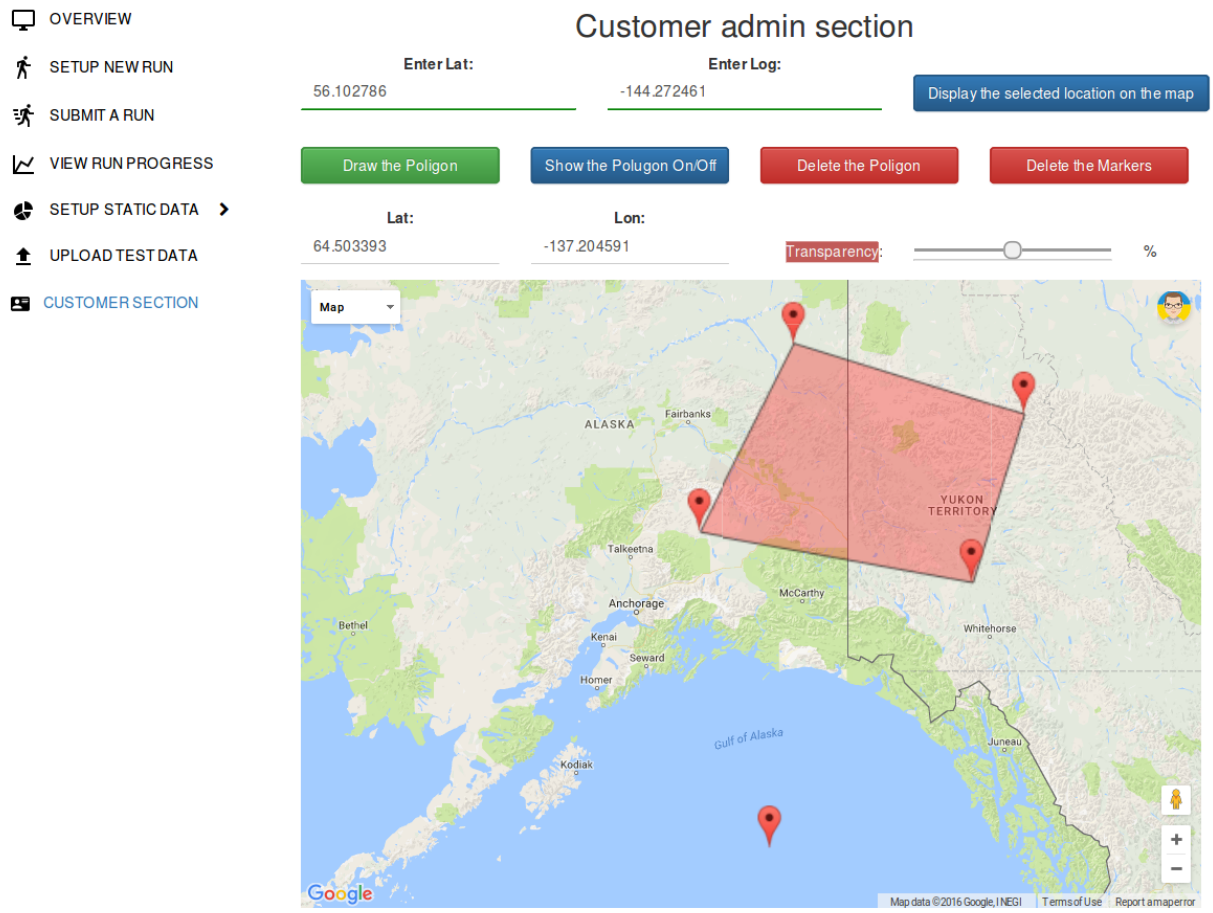
To create a new Card you need to push the *"New Processing Card"* button. In the menu select the needed type of card and fill it.

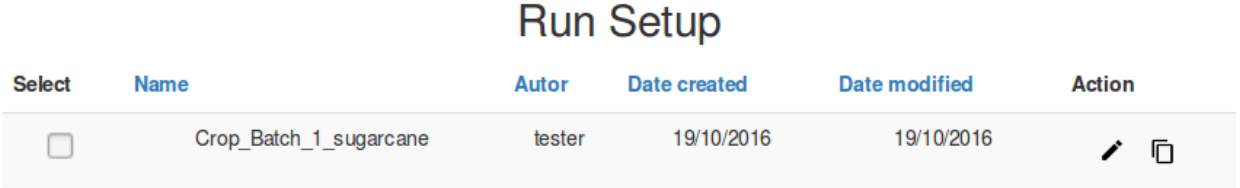
Create a new Card in a new Run

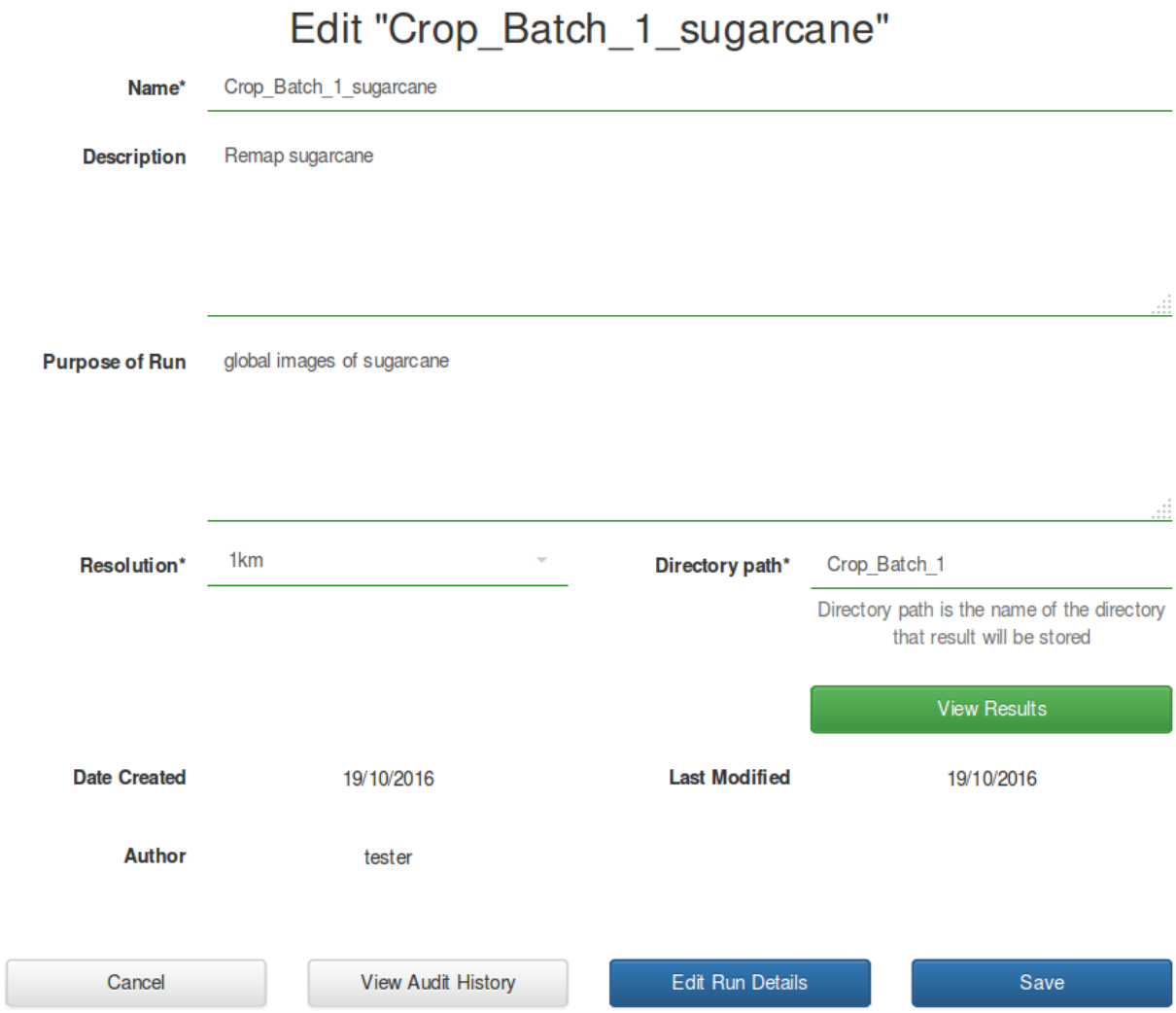
In the menu *"Setup New Run"* select *"New Run"*.

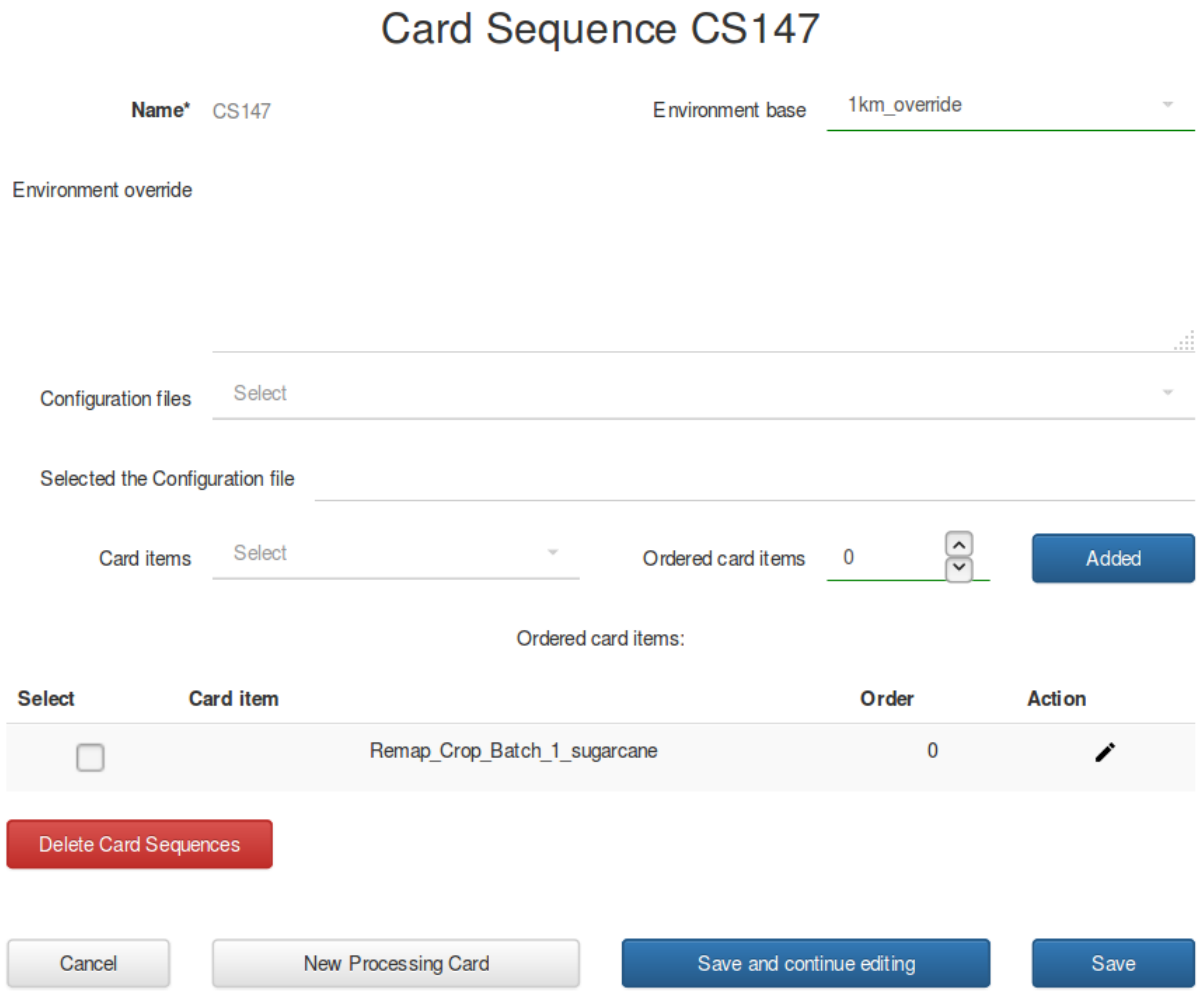
You can select an existing card in the *"Select Card item"*

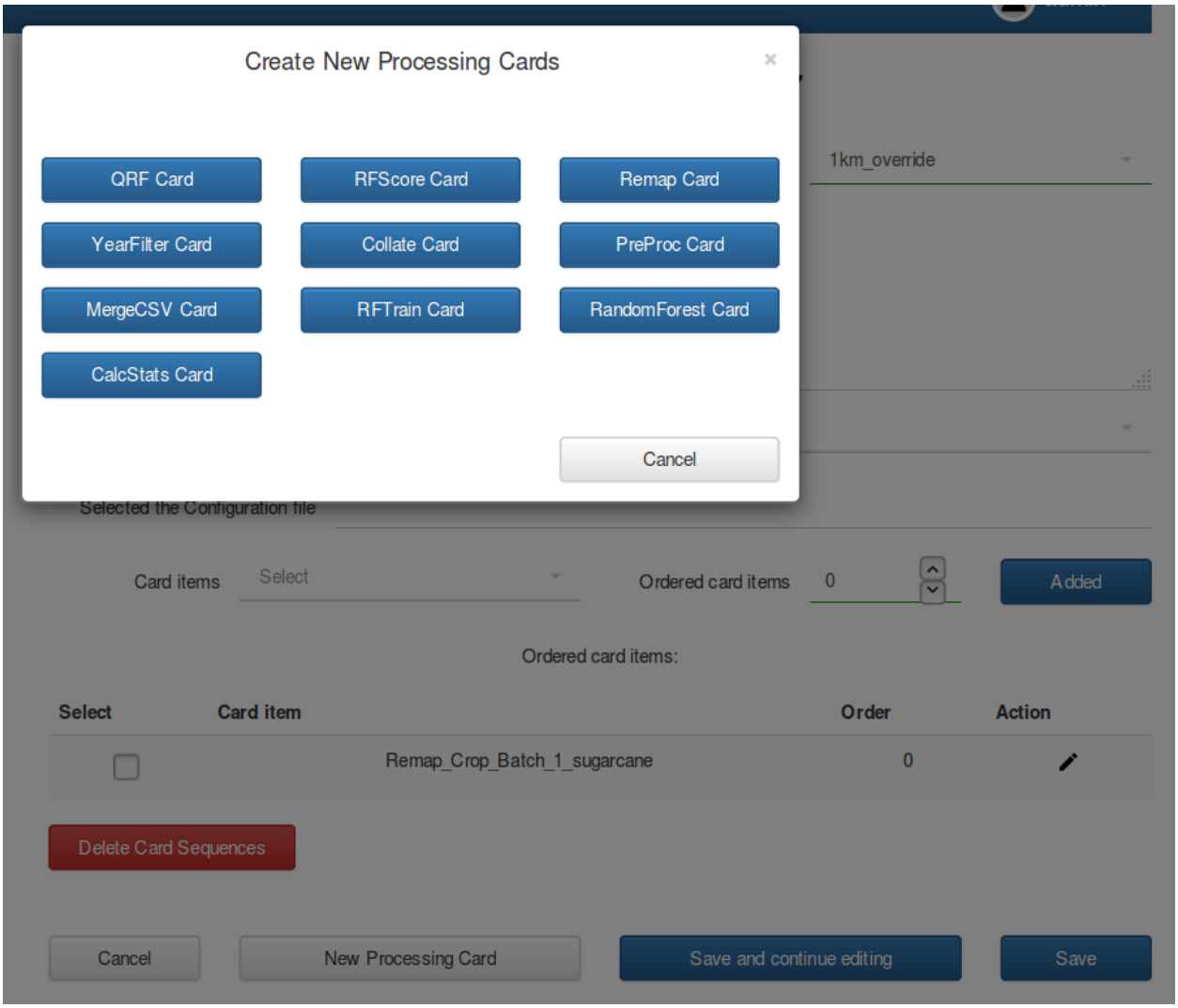
To create a new card, you need to save the Run and continue editing by pressing the *"Save and continue editing"*. After creating a new Run, to create a new card sequence. You need to click *"Edit Run Details"* button. In the edit form *"Card sequence"* creating Run click *"New Processing Card"*. In the menu select the needed type of card and fill it.

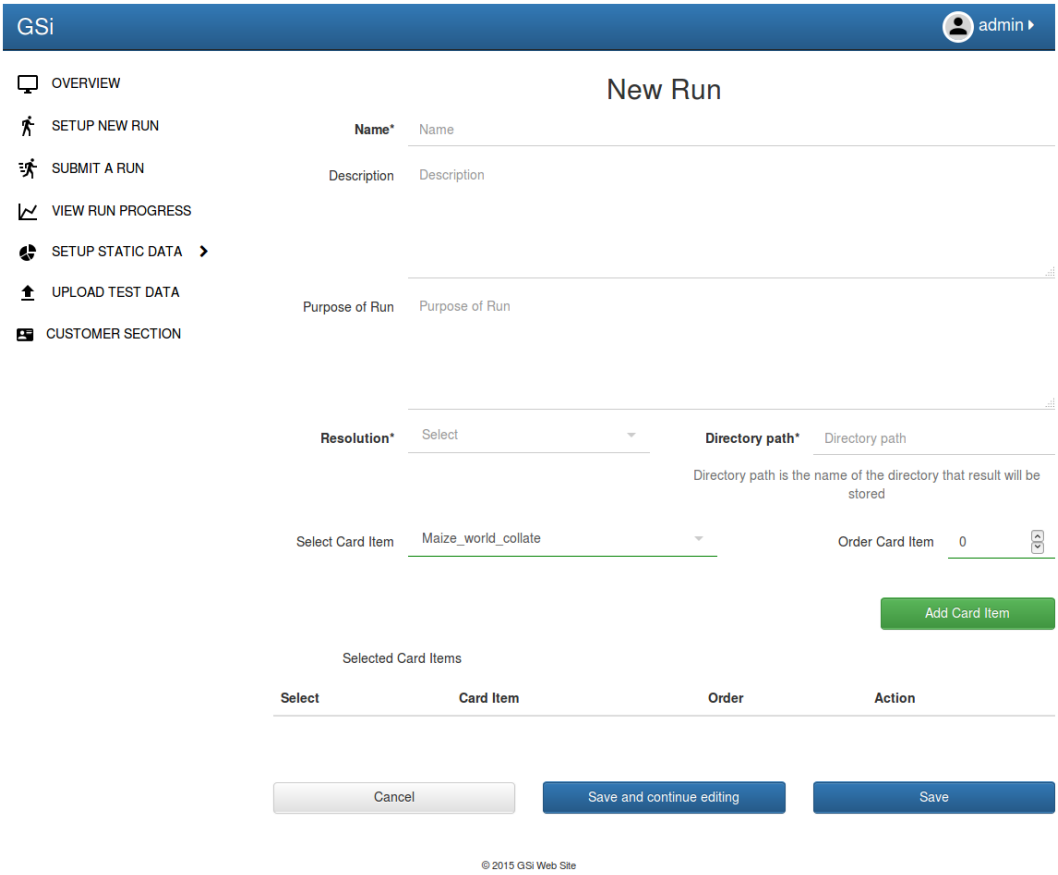

OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
UPLOAD TEST DATA
CUSTOMER SECTION
Customer admin section
Enter Lat:
56.102786
Enter Log:
-144.272461
Display the selected location on the map
Draw the Poligon
Show the Polugon On/Off
Delete the Poligon
Delete the Markers
Lat:
64.503393
Lon:
-137.204591
Transparency:
%
Map
ALASKA
Fairbanks
YUKON TERRITORY
Talkeetna
Anchorage
McCarthy
Whitehorse
Bethel
Kenai
Seward
Homer
Gulf of Alaska
Juneau
Kodiak
Google
Map data ©2016 Google, INEGI
Terms of Use
Report a map error

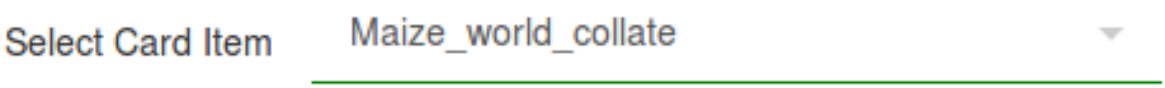

Run Setup
Select
Name
Autor
Date created
Date modified
Action
Crop_Batch_1_sugarcane
tester
19/10/2016
19/10/2016

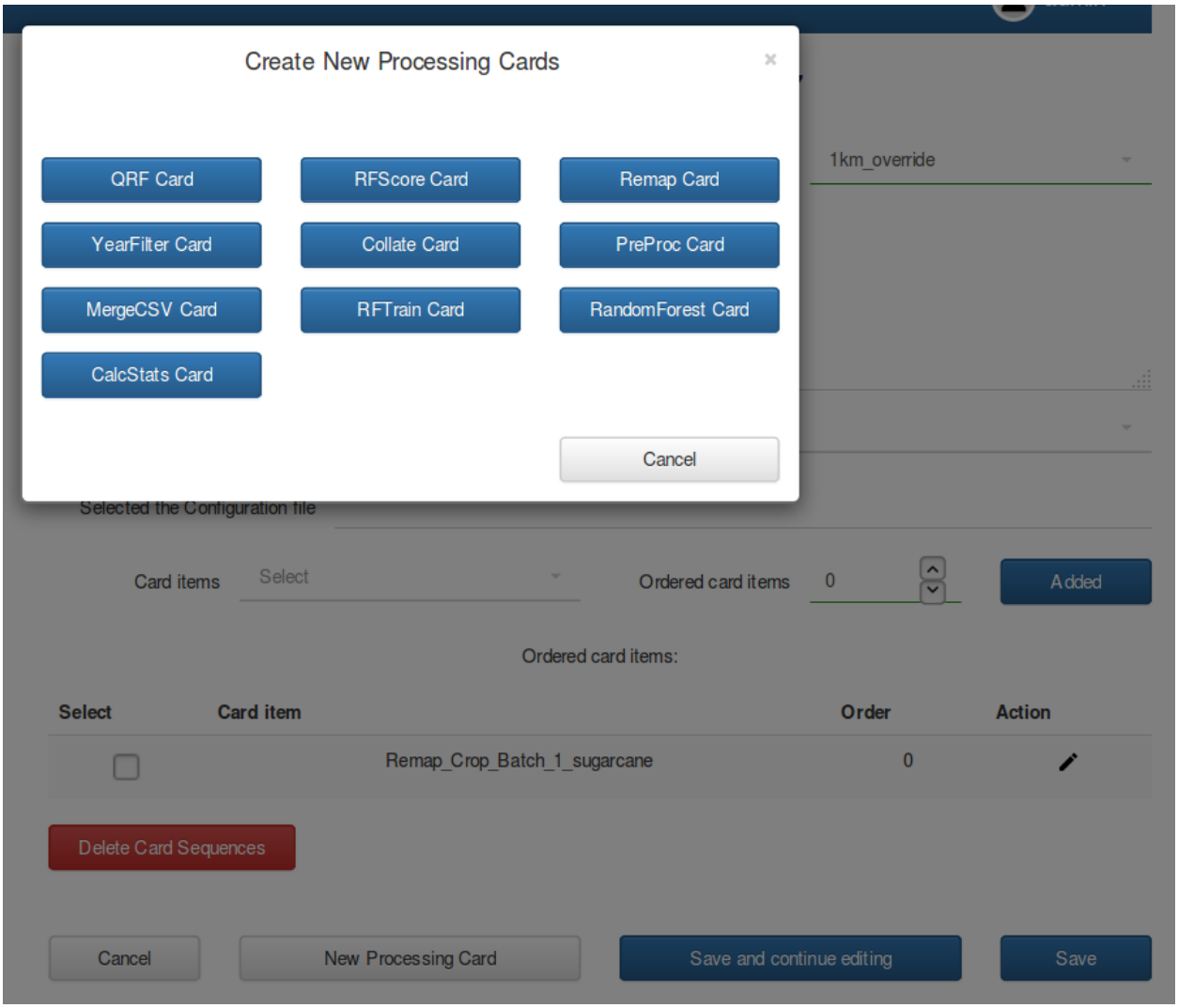

Edit "Crop_Batch_1_sugarcane"
Name*
Crop_Batch_1_sugarcane
Description
Remap sugarcane
Purpose of Run
global images of sugarcane
Resolution*
1km
Directory path*
Crop_Batch_1
Directory path is the name of the directory that result will be stored
View Results
Date Created
19/10/2016
Last Modified
19/10/2016
Author
tester
Cancel
View Audit History
Edit Run Details
Save


Card Sequence CS147
Name*
CS147
Environment base
1km_override
Environment override
Configuration files
Select
Selected the Configuration file
Card items
Select
Ordered card items
0
Added
Ordered card items:
Select
Card item
Order
Action
Remap_Crop_Batch_1_sugarcane
0
Delete Card Sequences
Cancel
New Processing Card
Save and continue editing
Save


Create New Processing Cards
QRF Card
RFScore Card
Remap Card
YearFilter Card
Collate Card
PreProc Card
MergeCSV Card
RFTrain Card
RandomForest Card
CalcStats Card
Cancel
1km_override
Selected the Configuration file
Card items
Select
Ordered card items
0
Added
Ordered card items:
Select
Card item
Order
Action
Remap_Crop_Batch_1_sugarcane
0
Delete Card Sequences
Cancel
New Processing Card
Save and continue editing
Save


GSi
admin
OVERVIEW
SETUP NEW RUN
SUBMIT A RUN
VIEW RUN PROGRESS
SETUP STATIC DATA
UPLOAD TEST DATA
CUSTOMER SECTION
New Run
Name*
Name
Description
Description
Purpose of Run
Purpose of Run
Resolution*
Select
Directory path*
Directory path
Directory path is the name of the directory that result will be stored
Select Card Item
Maize_world_collate
Order Card Item
0
Add Card Item
Selected Card Items
Select
Card Item
Order
Action
Cancel
Save and continue editing
Save
© 2015 GSi Web Site


Select Card Item
Maize_world_collate


Create New Processing Cards
QRF Card
RFScore Card
Remap Card
YearFilter Card
Collate Card
PreProc Card
MergeCSV Card
RFTrain Card
RandomForest Card
CalcStats Card
Cancel
1km_override
Selected the Configuration file
Card items
Select
Ordered card items
0
Added
Ordered card items:
Select
Card item
Order
Action
Remap_Crop_Batch_1_sugarcane
0
Delete Card Sequences
Cancel
New Processing Card
Save and continue editing
Save

1.4 Backend

1.4.1 Project on the server

The path to the project on the server.

```
$ /home/gsi/sites/gsi_website
```

1.4.2 Restart the server from the local computer

Source code the GSi project is in the repository. For the project management need to clone it to your local computer in the home folder:

```
$ cd /home/[user]
$ git clone https://github.com/gtsarik/GSI.git
```

After cloning the project you want to add a new file fab_local.py:

```
$ cd GSI
$ touch fab_local.py
```

Open the fab_local.py file:

```
$ vim fab_local.py
```

and add 3 variables in it:

```
$ GSI_APP_SERVER = 'gsi@indy4.epcc.ed.ac.uk'
$ REMOTE_CODE_DIR = [remote root folder of the project]
$ ENV_PASS = 'password to login on ssh'
```

The example for the REMOTE_CODE_DIR variable. The GSi project is along this path: /home/gsi/sites/gsi_website, so REMOTE_CODE_DIR = 'sites/gsi_website'

Save and close the file fab_local.py. Reboot the server:

```
$ make restart
```

1.4.3 Restart the server remotely

```
$ ssh gsi@indy4.epcc.ed.ac.uk
$ [enter password]
$ sudo service supervisor restart
$ [enter password]
```

1.5 Project structure

1.5.1 The project includes

- package for work with API: **api**;
- three applications: **cards**, **gsi**, **log**;

- package with auxiliary functions: **core**;
- folder with statics (styles, java script files, the image to the project): **static**;
- folder with templates: **template**;
- package with custom tags: **tags**;
- folder with technical documentations: **docs**;

– *"__init__.py"*: the file for to initialize of the packages.

– *"urls.py"*: the file for the URLs of web pages.

– *"admin.py"*: the file to display models in the admin part of the project.

– *"migrations"*: the package migration of models. At each change in the tables creates a new migration.

– *"template"*: the folder for templates of web pages.

– *"static/css"*: the folder for styles of css.

– *"static/js"*: the folder for JavaScript files which used in the project.

– *"static/img"*: the folder for images which used in the project.

1.5.2 Structure of folders

api - source code to work with api:

```
|-- api
|   |-- __init__.py
|   -- views.py
```

cards - source code to work with cards application:

```
|-- cards
|   |-- admin.py
|   |-- cards_forms.py
|   |-- card_update_create.py
|   |-- get_card_post.py
|   |-- __init__.py
|   |-- management
|   |   |-- commands
|   |   |   |-- fill_name_for_carditem_card.py
|   |   |   -- __init__.py
|   |   -- __init__.py
|   |-- migrations
|   |   |-- 0001_initial.py
|   |   |-- ...
|   |   |-- 0042_carditem_name.py
|   |   |-- __init__.py
|   |-- models.py
|   |-- urls.py
|   |-- views_card_runid_csid.py
|   |-- views.py
```

gsi - source code to work with gsi application:

```
|-- gsi
|   |-- admin.py
|   |-- gsi_forms.py
```

```
|   |-- gsi_update_create.py
|   |-- __init__.py
|   |-- migrations
|   |   |-- 0001_initial.py
|   |   |-- ...
|   |   |-- 0026_auto_20160909_0806.py
|   |   -- __init__.py
|   |-- models.py
|   |-- settings_local.py
|   |-- settings.py
|   |-- signals.py
|   |-- tests
|   |   |-- __init__.py
|   |   -- test_signals.py
|   |-- urls.py
|   |-- views_cs_card_runid_csid.py
|   |-- views.py
|   -- wsgi.py
```

log - source code to work with log application:

```
|-- gsi
|   |-- admin.py
|   |-- __init__.py
|   |-- logger.py
|   |-- migrations
|   |   |-- 0001_initial.py
|   |   |-- 0002_logdebug.py
|   |   |-- 0003_remove_logdebug_user.py
|   |   -- __init__.py
|   |-- models.py
|   |-- tests.py
|   -- views.py
```

core - folder with auxiliary functions:

```
|-- core
|   |-- copy_card.py
|   |-- get_post.py
|   |-- __init__.py
|   |-- multithreaded.py
|   |-- paginations.py
|   |-- utils.py
|   -- validator_gsi.py
```

static - folder with statics:

```
|-- static
|   |-- css
|   |   |-- form_style.css
|   |   |-- jquery.fs.selecter.css
|   |   |-- main_reg.css
|   |   -- styles.css
|   |-- img
|   |   |-- account-circle.png
|   |   |-- anonim.png
|   |   |-- back-18.png
|   |   |-- background_log.jpg
|   |   |-- back-to-up-18.png
```

```
|   |   |-- chevron-double-right-18.png
|   |   |-- copy-18.png
|   |   |-- customer_section.png
|   |   |-- delete-18.png
|   |   |-- details_18.png
|   |   |-- download-18.png
|   |   |-- edit-18.png
|   |   |-- editor
|   |   |   |-- anchor.png
|   |   |   |-- bold.png
|   |   |   |-- border-image.png
|   |   |   |-- code.png
|   |   |   |-- header-3.png
|   |   |   |-- header-4.png
|   |   |   |-- image.png
|   |   |   |-- italic.png
|   |   |   |-- link.png
|   |   |   |-- ordered.png
|   |   |   |-- paragraph.png
|   |   |   |-- quote.png
|   |   |   |-- top.png
|   |   |   |-- underline.png
|   |   |   -- unordered.png
|   |   |-- email_18.png
|   |   |-- eye_18.png
|   |   |-- file-18.png
|   |   |-- file-archive-18.png
|   |   |-- file-bin-18.png
|   |   |-- file-document-18.png
|   |   |-- file-image-18.png
|   |   |-- file-pdf-18.png
|   |   |-- file-word-18.png
|   |   |-- folder-18.png
|   |   |-- go-folder-18.png
|   |   |-- image-18.png
|   |   |-- logout.png
|   |   |-- menu-down-18.png
|   |   |-- order-18.png
|   |   |-- overview.png
|   |   |-- parallel-1-18.png
|   |   |-- parallel-2-18.png
|   |   |-- profile-18.png
|   |   |-- settings.png
|   |   |-- setup_new_run.png
|   |   |-- setup_static_data.png
|   |   |-- submit_a_run.png
|   |   |-- upload.png
|   |   -- view_run_progress.png
|   |-- js
|   |   |-- jquery.fs.selecter.min.js
|   |   |-- modernizr.js
|   |   |-- registrations.js
|   |   |-- scripts_ajax.js
|   |   |-- scripts_google_maps.js
|   |   -- scripts.js
```

template - folder with templates:

```
    |-- template
    |   |-- base.html
    |   |-- base_registrations.html
    |   |-- base_wiki.html
    |   |-- cards
    |   |   |-- _calcstats_form.html
    |   |   |-- _collate_form.html
    |   |   |-- _create_processing_card_form.html
    |   |   |-- _mergecsv_form.html
    |   |   |-- _preproc_form.html
    |   |   |-- processing_card_new_run.html
    |   |   |-- _qrf_form.html
    |   |   |-- _randomforest_form.html
    |   |   |-- _remap_form.html
    |   |   |-- _rfscore_form.html
    |   |   |-- _rftrain_form.html
    |   |   |-- runid_csid_card.html
    |   |   -- _year_filter_form.html
    |   |-- gsi
    |   |   |-- _area_form.html
    |   |   |-- areas_list.html
    |   |   |-- audit_history.html
    |   |   |-- card_editions.html
    |   |   |-- card_item_update.html
    |   |   |-- _card_sequence_form.html
    |   |   |-- card_sequence.html
    |   |   |-- card_sequence_update.html
    |   |   |-- cards_list.html
    |   |   |-- _cs_calcstats_form.html
    |   |   |-- _cs_collate_form.html
    |   |   |-- _cs_mergecsv_form.html
    |   |   |-- _cs_preproc_form.html
    |   |   |-- _cs_qrf_form.html
    |   |   |-- _cs_randomforest_form.html
    |   |   |-- _cs_remap_form.html
    |   |   |-- _cs_rfscore_form.html
    |   |   |-- _cs_rftrain_form.html
    |   |   |-- _cs_year_filter_form.html
    |   |   |-- customer_section.html
    |   |   |-- _env_group_form.html
    |   |   |-- environment_groups_list.html
    |   |   |-- execute_run.html
    |   |   |-- gsi_map.html
    |   |   |-- home_variable_setup.html
    |   |   |-- index.html
    |   |   |-- _input_data_dir_form.html
    |   |   |-- input_data_dir_list.html
    |   |   |-- _modal_new_card.html
    |   |   |-- new_card_sequence.html
    |   |   |-- new_run.html
    |   |   |-- _ordered_card_items.html
    |   |   |-- _resolution_form.html
    |   |   |-- resolution_list.html
    |   |   |-- run_details.html
    |   |   |-- run_progress.html
    |   |   |-- run_setup.html
    |   |   |-- run_update.html
    |   |   |-- _satellite_form.html
```

```
|   |   |-- satellite_list.html
|   |   |-- static_data_item_edit.html
|   |   |-- sub_card_details.html
|   |   |-- submit_run.html
|   |   |-- _tile_form.html
|   |   |-- tiles_list.html
|   |   |-- upload_file.html
|   |   |-- view_log_file.html
|   |   |-- view_log_file_sub_card.html
|   |   |-- view_results_folder.html
|   |   |-- view_results.html
|   |   |-- _year_form.html
|   |   |-- _years_group_form.html
|   |   |-- years_group_list.html
|   |   -- years_list.html
|   |-- _modal_check_delete_items.html
|   |-- _modal_preload.html
|   |-- paginations.html
|   |-- registration
|   |   |-- login.html
|   |   |-- password_change_done.html
|   |   |-- password_change_form.html
|   |   |-- password_reset_complete.html
|   |   |-- password_reset_confirm.html
|   |   |-- password_reset_done.html
|   |   |-- password_reset_email.html
|   |   |-- password_reset_form.html
|   |   -- registration_form.html
```

tags - folder with custom tags:

```
|-- tags
|  |-- __init__.py
|  |-- __init__.pyc
|  -- templatetags
|      |-- gsi_tags.py
|      -- __init__.py
```

docs - folder with technical documentations:

```
|-- docs
|   |-- build
|   |   |-- doctrees
|   |   |   |-- environment.pickle
|   |   |   |-- _general
|   |   |   |   |-- _1_technologies
|   |   |   |   |   |-- 1_1_backend.doctree
|   |   |   |   |   -- 1_2_frontend.doctree
|   |   |   |   |-- 1_used_technologies.doctree
|   |   |   |   |-- 2_designation_of_graphical_icons.doctree
|   |   |   |   |-- _3_frontend
|   |   |   |   |   |-- 3_1_authorization.doctree
|   |   |   |   |   |-- 3_2_sidebar.doctree
|   |   |   |   |   -- 3_3_creating_of_cards.doctree
|   |   |   |   |-- 3_frontend.doctree
|   |   |   |   -- 4_backend.doctree
|   |   |   |-- index.doctree
|   |   |   -- _modules
```

```
    |   |   |       |-- api.doctree
    |   |   |       |-- cards.doctree
    |   |   |       |-- core.doctree
    |   |   |       |-- gsi.doctree
    |   |   |       |-- js.doctree
    |   |   |       |-- log.doctree
    |   |   |       -- structure.doctree
    |   |   |-- html
    |   |   |   |-- _general
    |   |   |   |   |-- _1_technologies
    |   |   |   |   |   |-- 1_1_backend.html
    |   |   |   |   |   -- 1_2_frontend.html
    |   |   |   |   |-- 1_used_technologies.html
    |   |   |   |   |-- 2_designation_of_graphical_icons.html
    |   |   |   |   |-- _3_frontend
    |   |   |   |   |   |-- 3_1_authorization.html
    |   |   |   |   |   |-- 3_2_sidebar.html
    |   |   |   |   |   -- 3_3_creating_of_cards.html
    |   |   |   |   |-- 3_frontend.html
    |   |   |   |   -- 4_backend.html
    |   |   |   |-- genindex.html
    |   |   |   |-- _images
    |   |   |   |   |-- card_01.png
    |   |   |   |   |-- card_02.png
    |   |   |   |   |-- card_03.png
    |   |   |   |   |-- card_04.png
    |   |   |   |   |-- copy.png
    |   |   |   |   |-- delete.png
    |   |   |   |   |-- details.png
    |   |   |   |   |-- edit_area.png
    |   |   |   |   |-- edit_eg.png
    |   |   |   |   |-- edit.png
    |   |   |   |   |-- edit_yg.png
    |   |   |   |   |-- env_group.png
    |   |   |   |   |-- fill_sr.png
    |   |   |   |   |-- full_areas.png
    |   |   |   |   |-- full_cards.png
    |   |   |   |   |-- full_cs_polugon_markers.png
    |   |   |   |   |-- full_idd.png
    |   |   |   |   |-- full_resolutions.png
    |   |   |   |   |-- full_satellites.png
    |   |   |   |   |-- full_tiles.png
    |   |   |   |   |-- full_utd.png
    |   |   |   |   |-- full_years.png
    |   |   |   |   |-- full_yg.png
    |   |   |   |   |-- home_var.png
    |   |   |   |   |-- login_full.png
    |   |   |   |   |-- logs_current_progress.png
    |   |   |   |   |-- new_run_1.png
    |   |   |   |   |-- new_run.png
    |   |   |   |   |-- overview.png
    |   |   |   |   |-- pass_reset.png
    |   |   |   |   |-- registration.png
    |   |   |   |   |-- run_progress.png
    |   |   |   |   -- select_card_item.png
    |   |   |   |-- index.html
    |   |   |   |-- _modules
    |   |   |   |   |-- api
```

```
|   |   |   |   |   -- views.html
|   |   |   |   |-- api.html
|   |   |   |   |-- cards
|   |   |   |   |   |-- cards_forms.html
|   |   |   |   |   |-- card_update_create.html
|   |   |   |   |   |-- models.html
|   |   |   |   |   |-- views_card_runid_csid.html
|   |   |   |   |   -- views.html
|   |   |   |   |-- cards.html
|   |   |   |   |-- core
|   |   |   |   |   |-- copy_card.html
|   |   |   |   |   |-- get_post.html
|   |   |   |   |   |-- multithreaded.html
|   |   |   |   |   |-- paginations.html
|   |   |   |   |   |-- utils.html
|   |   |   |   |   -- validator_gsi.html
|   |   |   |   |-- core.html
|   |   |   |   |-- gsi
|   |   |   |   |   |-- gsi_forms.html
|   |   |   |   |   |-- gsi_update_create.html
|   |   |   |   |   |-- models.html
|   |   |   |   |   |-- signals.html
|   |   |   |   |   |-- views_card_runid_csid.html
|   |   |   |   |   -- views.html
|   |   |   |   |-- gsi.html
|   |   |   |   |-- index.html
|   |   |   |   |-- js.html
|   |   |   |   |-- log
|   |   |   |   |   |-- logger.html
|   |   |   |   |   -- models.html
|   |   |   |   |-- log.html
|   |   |   |   -- structure.html
|   |   |   |-- objects.inv
|   |   |   |-- py-modindex.html
|   |   |   |-- search.html
|   |   |   |-- searchindex.js
|   |   |   |-- _sources
|   |   |   |   |-- _general
|   |   |   |   |   |-- _1_technologies
|   |   |   |   |   |   |-- 1_1_backend.txt
|   |   |   |   |   |   -- 1_2_frontend.txt
|   |   |   |   |   |-- 1_used_technologies.txt
|   |   |   |   |   |-- 2_designation_of_graphical_icons.txt
|   |   |   |   |   |-- _3_frontend
|   |   |   |   |   |   |-- 3_1_authorization.txt
|   |   |   |   |   |   |-- 3_2_sidebar.txt
|   |   |   |   |   |   -- 3_3_creating_of_cards.txt
|   |   |   |   |   |-- 3_frontend.txt
|   |   |   |   |   -- 4_backend.txt
|   |   |   |   |-- index.txt
|   |   |   |   -- _modules
|   |   |   |       |-- api.txt
|   |   |   |       |-- cards.txt
|   |   |   |       |-- core.txt
|   |   |   |       |-- gsi.txt
|   |   |   |       |-- js.txt
|   |   |   |       |-- log.txt
|   |   |   |       -- structure.txt
```

```
    |   |   |   -- _static
    |   |   |       |-- ajax-loader.gif
    |   |   |       |-- anonim.png
    |   |   |       |-- areas
    |   |   |       |   |-- edit_area.png
    |   |   |       |   -- full_areas.png
    |   |   |       |-- basic.css
    |   |   |       |-- card_create
    |   |   |       |   |-- card_01.png
    |   |   |       |   |-- card_02.png
    |   |   |       |   |-- card_03.png
    |   |   |       |   |-- card_04.png
    |   |   |       |   -- new_run
    |   |   |       |       |-- new_run_1.png
    |   |   |       |       -- select_card_item.png
    |   |   |       |-- cards
    |   |   |       |   -- full_cards.png
    |   |   |       |-- comment-bright.png
    |   |   |       |-- comment-close.png
    |   |   |       |-- comment.png
    |   |   |       |-- copy.png
    |   |   |       |-- customer_section
    |   |   |       |   |-- full_cs.png
    |   |   |       |   -- full_cs_polugon_markers.png
    |   |   |       |-- delete.png
    |   |   |       |-- details.png
    |   |   |       |-- doctools.js
    |   |   |       |-- down.png
    |   |   |       |-- down-pressed.png
    |   |   |       |-- edit.png
    |   |   |       |-- env_group
    |   |   |       |   |-- edit_eg_pic.png
    |   |   |       |   |-- edit_eg.png
    |   |   |       |   -- env_group.png
    |   |   |       |-- file.png
    |   |   |       |-- home_var
    |   |   |       |   -- home_var.png
    |   |   |       |-- input_data_directory
    |   |   |       |   -- full_idd.png
    |   |   |       |-- jquery-1.11.1.js
    |   |   |       |-- jquery.js
    |   |   |       |-- login
    |   |   |       |   |-- login_empty.png
    |   |   |       |   |-- login_full.png
    |   |   |       |   |-- pass_reset.png
    |   |   |       |   -- registration.png
    |   |   |       |-- minus.png
    |   |   |       |-- nature.css
    |   |   |       |-- overview
    |   |   |       |   -- overview.png
    |   |   |       |-- plus.png
    |   |   |       |-- pygments.css
    |   |   |       |-- resolutions
    |   |   |       |   -- full_resolutions.png
    |   |   |       |-- run_progress
    |   |   |       |   |-- logs_current_progress.png
    |   |   |       |   |-- pic_view_current_progress.png
    |   |   |       |   |-- run_progress.png
```

```
|   |   |       |   -- view_current_progress.png
|   |   |       |-- satellites
|   |   |       |   -- full_satellites.png
|   |   |       |-- searchtools.js
|   |   |       |-- setup_new_run
|   |   |       |   |-- new_run.png
|   |   |       |   |-- n_run_copy.png
|   |   |       |   -- n_run_edit.png
|   |   |       |-- setup_static_data
|   |   |       |   -- home_var
|   |   |       |       -- full.png
|   |   |       |-- submit_a_run
|   |   |       |   -- fill_sr.png
|   |   |       |-- tiles
|   |   |       |   -- full_tiles.png
|   |   |       |-- underscore-1.3.1.js
|   |   |       |-- underscore.js
|   |   |       |-- upload_test_data
|   |   |       |   -- full_utd.png
|   |   |       |-- up.png
|   |   |       |-- up-pressed.png
|   |   |       |-- websupport.js
|   |   |       |-- years
|   |   |       |   -- full_years.png
|   |   |       -- years_group
|   |   |           |-- edit_yg.png
|   |   |           -- full_yg.png
|   |   -- singlehtml
|   |       |-- _images
|   |       |   |-- card_01.png
|   |       |   |-- card_02.png
|   |       |   |-- card_03.png
|   |       |   |-- card_04.png
|   |       |   |-- copy.png
|   |       |   |-- delete.png
|   |       |   |-- details.png
|   |       |   |-- edit_area.png
|   |       |   |-- edit_eg.png
|   |       |   |-- edit.png
|   |       |   |-- edit_yg.png
|   |       |   |-- env_group.png
|   |       |   |-- fill_sr.png
|   |       |   |-- full_areas.png
|   |       |   |-- full_cards.png
|   |       |   |-- full_cs_polugon_markers.png
|   |       |   |-- full_idd.png
|   |       |   |-- full_resolutions.png
|   |       |   |-- full_satellites.png
|   |       |   |-- full_tiles.png
|   |       |   |-- full_utd.png
|   |       |   |-- full_years.png
|   |       |   |-- full_yg.png
|   |       |   |-- home_var.png
|   |       |   |-- login_full.png
|   |       |   |-- logs_current_progress.png
|   |       |   |-- new_run_1.png
|   |       |   |-- new_run.png
|   |       |   |-- overview.png
```

```
|   |       |   |-- pass_reset.png
|   |       |   |-- registration.png
|   |       |   |-- run_progress.png
|   |       |   -- select_card_item.png
|   |       |-- index.html
|   |       |-- objects.inv
|   |       -- _static
|   |           |-- ajax-loader.gif
|   |           |-- anonim.png
|   |           |-- areas
|   |           |   |-- edit_area.png
|   |           |   -- full_areas.png
|   |           |-- basic.css
|   |           |-- card_create
|   |           |   |-- card_01.png
|   |           |   |-- card_02.png
|   |           |   |-- card_03.png
|   |           |   |-- card_04.png
|   |           |   -- new_run
|   |           |       |-- new_run_1.png
|   |           |       -- select_card_item.png
|   |           |-- cards
|   |           |   -- full_cards.png
|   |           |-- comment-bright.png
|   |           |-- comment-close.png
|   |           |-- comment.png
|   |           |-- copy.png
|   |           |-- customer_section
|   |           |   |-- full_cs.png
|   |           |   -- full_cs_polugon_markers.png
|   |           |-- delete.png
|   |           |-- details.png
|   |           |-- doctools.js
|   |           |-- down.png
|   |           |-- down-pressed.png
|   |           |-- edit.png
|   |           |-- env_group
|   |           |   |-- edit_eg_pic.png
|   |           |   |-- edit_eg.png
|   |           |   -- env_group.png
|   |           |-- file.png
|   |           |-- home_var
|   |           |   -- home_var.png
|   |           |-- input_data_directory
|   |           |   -- full_idd.png
|   |           |-- jquery-1.11.1.js
|   |           |-- jquery.js
|   |           |-- login
|   |           |   |-- login_empty.png
|   |           |   |-- login_full.png
|   |           |   |-- pass_reset.png
|   |           |   -- registration.png
|   |           |-- minus.png
|   |           |-- nature.css
|   |           |-- overview
|   |           |   -- overview.png
|   |           |-- plus.png
|   |           |-- pygments.css
```

```
|   |           |-- resolutions
|   |           |   -- full_resolutions.png
|   |           |-- run_progress
|   |           |   |-- logs_current_progress.png
|   |           |   |-- pic_view_current_progress.png
|   |           |   |-- run_progress.png
|   |           |   -- view_current_progress.png
|   |           |-- satellites
|   |           |   -- full_satellites.png
|   |           |-- searchtools.js
|   |           |-- setup_new_run
|   |           |   |-- new_run.png
|   |           |   |-- n_run_copy.png
|   |           |   -- n_run_edit.png
|   |           |-- setup_static_data
|   |           |   -- home_var
|   |           |       -- full.png
|   |           |-- submit_a_run
|   |           |   -- fill_sr.png
|   |           |-- tiles
|   |           |   -- full_tiles.png
|   |           |-- underscore-1.3.1.js
|   |           |-- underscore.js
|   |           |-- upload_test_data
|   |           |   -- full_utd.png
|   |           |-- up.png
|   |           |-- up-pressed.png
|   |           |-- websupport.js
|   |           |-- years
|   |           |   -- full_years.png
|   |           -- years_group
|   |               |-- edit_yg.png
|   |               -- full_yg.png
|   |-- Makefile
|   |-- reload.sh
|   |-- source
|   |   |-- conf.py
|   |   |-- _general
|   |   |   |-- _1_technologies
|   |   |   |   |-- 1_1_backend.rst
|   |   |   |   -- 1_2_frontend.rst
|   |   |   |-- 1_used_technologies.rst
|   |   |   |-- 2_designation_of_graphical_icons.rst
|   |   |   |-- _3_frontend
|   |   |   |   |-- 3_1_authorization.rst
|   |   |   |   |-- 3_2_sidebar.rst
|   |   |   |   -- 3_3_creating_of_cards.rst
|   |   |   |-- 3_frontend.rst
|   |   |   -- 4_backend.rst
|   |   |-- index.rst
|   |   |-- _modules
|   |   |   |-- api.rst
|   |   |   |-- cards.rst
|   |   |   |-- core.rst
|   |   |   |-- gsi.rst
|   |   |   |-- log.rst
|   |   |   -- structure.rst
|   |   |-- _static
```

```
|   |   |   |-- anonim.png
|   |   |   |-- areas
|   |   |   |   |-- edit_area.png
|   |   |   |   -- full_areas.png
|   |   |   |-- card_create
|   |   |   |   |-- card_01.png
|   |   |   |   |-- card_02.png
|   |   |   |   |-- card_03.png
|   |   |   |   |-- card_04.png
|   |   |   |   -- new_run
|   |   |   |       |-- new_run_1.png
|   |   |   |       -- select_card_item.png
|   |   |   |-- cards
|   |   |   |   -- full_cards.png
|   |   |   |-- copy.png
|   |   |   |-- customer_section
|   |   |   |   |-- full_cs.png
|   |   |   |   -- full_cs_polugon_markers.png
|   |   |   |-- delete.png
|   |   |   |-- details.png
|   |   |   |-- edit.png
|   |   |   |-- env_group
|   |   |   |   |-- edit_eg_pic.png
|   |   |   |   |-- edit_eg.png
|   |   |   |   -- env_group.png
|   |   |   |-- home_var
|   |   |   |   -- home_var.png
|   |   |   |-- input_data_directory
|   |   |   |   -- full_idd.png
|   |   |   |-- login
|   |   |   |   |-- login_empty.png
|   |   |   |   |-- login_full.png
|   |   |   |   |-- pass_reset.png
|   |   |   |   -- registration.png
|   |   |   |-- overview
|   |   |   |   -- overview.png
|   |   |   |-- resolutions
|   |   |   |   -- full_resolutions.png
|   |   |   |-- run_progress
|   |   |   |   |-- logs_current_progress.png
|   |   |   |   |-- pic_view_current_progress.png
|   |   |   |   |-- run_progress.png
|   |   |   |   -- view_current_progress.png
|   |   |   |-- satellites
|   |   |   |   -- full_satellites.png
|   |   |   |-- setup_new_run
|   |   |   |   |-- new_run.png
|   |   |   |   |-- n_run_copy.png
|   |   |   |   -- n_run_edit.png
|   |   |   |-- setup_static_data
|   |   |   |   -- home_var
|   |   |   |       -- full.png
|   |   |   |-- submit_a_run
|   |   |   |   -- fill_sr.png
|   |   |   |-- tiles
|   |   |   |   -- full_tiles.png
|   |   |   |-- upload_test_data
|   |   |   |   -- full_utd.png
```

```
    |   |   |   |-- years
    |   |   |   |   -- full_years.png
    |   |   |   -- years_group
    |   |   |       |-- edit_yg.png
    |   |   |       -- full_yg.png
    |   |   -- _templates
```

1.6 Api package

Package for work with API.

1.6.1 Api.views

api.views.**api_gsi_maps**(**args*, ***kwargs*)
API to get ready card images.

api.views.**is_finished**(*run_id*, *card_id*, *cur_counter*, *last*, *run_parallel*)
Function to determine the last card in a running list of cards.

Arguments

- *run_id*: run id
- *card_id*: card id
- *cur_counter*: the current cards position in the running list of cards
- *last*: the last position in the running list of cards
- *run_parallel*: boolean value that parallel card opledelyaet running or sequentially

api.views.**set_state_fail**(*obj*, *state*)
Set a card execution status if it does not 'fail'.

Arguments

- *obj*: card object
- *state*: the current status of the card

api.views.**update_run**(**args*, ***kwargs*)
Update the status of the card.

The function receives the request and the card data. If the launched the card is last, the process stops.

Arguments

- *request*: request
- *run_id*: card details. Presented as a string: <run_id>.<card_sequence_id>.<order_card_item_id>.<current_position>

1.7 Cards aplication

1.7.1 Cards.models

class cards.models.**CalcStats**(**args*, ***kwargs*)
Model for the cards CalcStats.

Functions Applies filtering (either Fourier filter of specified order, or Kalman filter with specified gain scale factor) to all geotiff input images for the specified time period. Also performs statistics on the resulting imagery, giving mean, min, max, median values per pixel.

Fields **output_tile_subdir**: Output directory

year_group: Relation with the YearGroup model from the applications GSI

area: Relation with the Area model from the applications GSI

period: Time period over which statistics are calculated

doy_variable: Character field

filter: Order of Fourier series to fit to time-series (default=0)

filter_out: Filtered outputs

input_fourier: Character field

out_dir: Output directory (if non-MODIS tile specified as first argument)

path_spec_location: The path to the location of a special file

class cards.models.**CardItem**(**args*, ***kwargs*)

Model for the cards CardItem.

Functions It writes to the database all models Cards application.

CONTENT_LIMIT The list of the all models Cards application.

Fields **name**: Model name

content_type: Relation with the ContentType model

object_id: ID of the element from the model

content_object: Associates model element and its ID

order: The order of an element

class cards.models.**Collate**(**args*, ***kwargs*)

Model for the cards Collate.

Functions Processing GeoTIFF files to specify the tiles.

Fields **area**: Relation with the Area model from the applications GSI

mode: Data are MODIS, DEM and soil

output_tile_subdir: Output tile subdirectory/filename root

input_scale_factor: Input scale factor

input_data_directory: Relation with the InputDataDirectory model from the applications GSI

input_files: Input ad-hoc geotiff filename, assumed to be in $RF_AUXDATA_DIR directory. Relation with the InputDataDirectory model from the applications GSI

class cards.models.**MergeCSV**(**args*, ***kwargs*)

Model for the cards MergeCSV.

Functions Data on csv files.

Fields **csv1**: Filename

csv2: Filename

class cards.models.**NamedModel**(**args*, ***kwargs*)
Abstract model.

Inherits the *"name"* field

class cards.models.**OrderedCardItem**(**args*, ***kwargs*)
Model for the cards OrderedCardItem.

Functions It writes to the database all models Cards application.

CONTENT_LIMIT The list of the all models Cards application.

Fields **name**: Model name

content_type: Relation with the ContentType model

object_id: ID of the element from the model

content_object: Associates model element and its ID

order: The order of an element the CardItem model

class cards.models.**ParallelModel**(**args*, ***kwargs*)
Abstract model.

Inherits the *"run_parallel"* field

class cards.models.**PreProc**(**args*, ***kwargs*)
Model for the cards PreProc.

Functions The normal use for PreProc is to extract selected bands from MODIS hdf files (in the per-tile directory structure under $MODIS_DIR), saving the extracted images as GeoTiffs in the appropriate per-tile location in the directory structure under $SAT_TIF_DIR.

Fields **area**: Relation with the Area model from the applications GSI

mode: Extracts GeoTiffs from hdf and performs stats for selected year(s)

year_group: Relation with the YearGroup model from the applications GSI

path_spec_location: The full path to a specific hdf file

class cards.models.**QRF**(**args*, ***kwargs*)
Model for the cards QRF.

Functions The Quantile Random Forest (QRF) process analyses all of the decision trees generated by RFtrain, looking at the distributions of all Y-parameter values across all trees, to determine the confidence that can be placed on each predicted value; this information can be used subsequently in scoring, to provide a range of statiscs for each scored pixel, including: Median, lower quartile, upper quartile, min (eg <5%) quantile, max (eg >95%) quantile, standard error, standard deviation, variance, and per-pixel quantile value.

Fields **interval**: Confidence interval (overrides $QRF_INTERVAL)

number_of_trees: Number of trees to use for QRF (overrides actual number of trees created)

number_of_threads: Number of parallel threads to use for QRF

directory: Identifies parent directory for trees, if not in default location (under $RF_DIR)

class cards.models.**RFScore**(**args*, ***kwargs*)
Model for the cards RFScore.

Functions RFscore uses the Random Forest decision trees generated by RFtrain to predict the Y-parameter value for each combination of X-parameter values in the specified input files. The range of possible output files depends on whether QRF has been run, or not. If QRF has been

run for the selected training set, then a number of statistical measures can be output, (based on the setting of the <QRFopts> command line argument). If QRF has not been run, then the only output will be the Conditional Mean of the predicted Y-parameters values.

Fields **area**: Relation with the Area model from the applications GSI

year_group: Relation with the YearGroup model from the applications GSI

bias_corrn:Nnumber of parallel threads to use for QRF

number_of_threads: Number of parallel threads used for QRF, if enabled ($QRF_INTERVAL>0)

QRFopts: Controls outputs: 0=None, 1=Mean, 2=Median, 4=Min, 8=Max, 16=LowerQ, 32=UpperQ, 64=QRFstats (set OR of values), 128=Std Error, 256=Std Dev, 512=Variance (sigma sq), 1024=nVar, 2048=per pixel Quantile

ref_target: Datatype used for training (eg 'Biomass'), which triggers generation of new target <90th quantile

clean_name: Text to uniquely identify filename for new cleaned target (<RefTarget>_<CleanName>)

class cards.models.**RFTrain**(**args*, ***kwargs*)

Model for the cards RFTrain.

Functions Core Random Forest processing, which uses the specified "target" (Y-parameter) imagery as the basis for training using all the specified input "X-parameter" images as the basis. (this will enable subsequent scoring using other images (different times/locations) for X- parameters to derive the target Y-parameter values for those times and locations.

The main output from RFtrain are a set of decision trees, which define how any combination of input X-parameters can be used to predict a corresponding output Y-parameter.

Fields **tile_type**: Relation with the TileType model from the applications GSI

number_of_trees: The number of trees to use in randomForest (default = 50)

value: Value

config_file: The configuration file

output_tile_subdir: The output tile subdirectory

input_scale_factor: The input scale factor

training: Integer field

number_of_variable: Number of iterations used for tree node (or automatically calculated if not defined)

number_of_thread: Number of parallel threads used for tree generation (default is one, if not defined)

class cards.models.**RandomForest**(**args*, ***kwargs*)

Model for the cards RandomForest.

Functions Random Forest processing.

Fields **aoi_name**: Character field

satellite: Relation with the Satellite model from the applications GSI

param_set: Text field

run_set: Character field

model: Character field

mvrf: Character field

class cards.models.**Remap**(**args*, ***kwargs*)

Model for the cards Remap.

Functions Extracts/mosaics and remaps the specified input image(s) to the specified output min/max lat/long limits. If a shape file is specified, then the polygons within the shape file are used to identify which portions of the output will have remapped pixel values (all regions outside the polygons will be set to NO_DATA (=-9999)). This utility can be used for remapping files stored within the MODIS tiles directory structure (for a specified parameter name), with output as WGS84 image. It can also be used for extracting a subset of an existing WGS84 image and rescaling to the same size, with the same spatial resolution as other extracted images. The latter functionality can alternatively be achieved using the GDAL utility – gdalwarp. Remap also provides the option to output images as .png format files, using a user-specified colour look-up table (LUT). The lut may be in any of three formats: ".lut" as used by the OpenSource ImageJ program, .pal format, which are simple ASCII text files, with 2 header lines, then each of 256 lines containing "R G B" values, or .txt files which allow for specifying the LUT in terms of breakpoints, where each line contain 4 values ("breakpoint R G B"), and the software interpolates between each breakpoint to give the final 256 colour range.

Fields **year_group**: Relation with the YearGroup model from the applications GSI

file_spec: <subdir>/<fileroot> below $RF_DIR/Tiles/hxxvyy

roi: <filename> With single line comprising: MinLat MinLon MaxLat MaxLon

model_name: The name of the model

output_root: Pathname and root for filename of all outputs generated (<OutRoot>_<ParamName>.tif)

output_suffix: Suffix for output filenames, eg <OutRoout>/<ParamName>_<OutSuffix>.tif

scale: Spatial resolution of required output (1000=1km, 250=250m etc)

output: 1 for .tif, 2 for .png, 4 for .csv (OR value for multiple outputs)

color_table: Full pathname for required colour LUT for .png

refstats_file: Geotiff <filename>.tif

refstats_scale: Scale factor for RefStatsFile pixel values

conditional_mean: Boolean field

conditional_min: Boolean field

conditional_median: Boolean field

conditional_max: Boolean field

lower_quartile: Boolean field

upper_quartile: Boolean field

class cards.models.**YearFilter**(**args*, ***kwargs*)

Model for the cards YearFilter.

Functions Applies filtering across multiple years (either Fourier filter of specified order, or Kalman filter with specified gain scale factor) to all geotiff input images for the specified time period, covering (per-year) output from Calcstats or output from RFscore.

Fields **area**: Relation with the Area model from the applications GSI

filetype: Values to determine which files to output

filter: Smoothing filter method

filter_output: Selection of output images

extend_start: Number of years to extend the start backwards to attempt better early fit (default=0)

input_fourier: Value for NDVI files, maximum Fourier order used in CalcStats

output_directory: Specific output directory

input_directory: Specific input directory

cards.models.**auto_add_card_item**(*sender*, *instance=None*, *created=False*, ***kwargs*)
To write a new element models of the Cards applications in the CardItem model.

cards.models.**get_card_item**(*self*)
Get an element the CardItem model or create a new.

1.7.2 Cards.views

cards.views.**proces_card_new_run**(*request*, **args*, ***kwargs*)
View at the creation of new cards.

When choosing a new card in "processing_card_menu" function transferred dictionary "rev_url" values {<button_a_new_card>: <url_to_create_a_new_card>}

cards.views.**processing_card_menu**(*request*, *rev_url*, *args=False*)
The function to receive a response depending on the type of card.

Arguments **request:** The request is sent to the server when processing the page

rev_url: Dictionary. It contains values for reverse after processing of form.

rev_url[<name_card>][0]: url for reverse,

If the element to edit, then the variable is its id:

rev_url[<name_card>][1]: id of element,

args: Boolean. If the value is True, then edit the element and contains id

1.7.3 Cards.views_card_runid_csid

cards.views_card_runid_csid.**runid_csid_calcstats_add**(*request*, **args*, ***kwargs*)
View for to create a new CalcStats card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_calcstats_edit**(*request*, **args*, ***kwargs*)
View for to edit the CalcStats card.

Arguments

- *run_id*: ID of the Run

- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_collate_add**(*request*, **args*, ***kwargs*)

View for to create a new Collate card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_collate_edit**(*request*, **args*, ***kwargs*)

View for to edit the Collate card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_mergecsv_add**(*request*, **args*, ***kwargs*)

View for to create a new MergeCSV card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_mergecsv_edit**(*request*, **args*, ***kwargs*)

View for to edit the MergeCSV card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_preproc_add**(*request*, **args*, ***kwargs*)

View for to create a new PreProc card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_preproc_edit**(*request*, **args*, ***kwargs*)

View for to edit the PreProc card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_qrf_add**(*request*, **args*, ***kwargs*)

View for to create a new QRF card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_qrf_edit**(*request*, **args*, ***kwargs*)

View for to edit the QRF card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_randomforest_add**(*request*, **args*, ***kwargs*)

View for to create a new RandomForest card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_randomforest_edit**(*request*, **args*, ***kwargs*)

View for to edit the RandomForest card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_remap_add**(*request*, **args*, ***kwargs*)

View for to create a new Remap card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_remap_edit**(*request*, **args*, ***kwargs*)

View for to edit the Remap card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_rfscore_add**(*request*, **args*, ***kwargs*)

View for to create a new RFScore card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_rfscore_edit**(*request*, **args*, ***kwargs*)

View for to edit the RFScore card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_rftrain_add**(*request*, **args*, ***kwargs*)

View for to create a new RFTrain card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_rftrain_edit**(*request*, **args*, ***kwargs*)
View for to edit the RFTrain card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_year_filter_add**(*request*, **args*, ***kwargs*)
View for to create a new YearFilter card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

cards.views_card_runid_csid.**runid_csid_year_filter_edit**(*request*, **args*, ***kwargs*)
View for to edit the YearFilter card.

Arguments

- *run_id*: ID of the Run
- *cs_id*: ID of the CardSequence

1.7.4 Cards.cards_forms

This module contains methods for creating and editing card forms.

class cards.cards_forms.**CalcStatsForm**(**args*, ***kwargs*)
Form for editing CalcStats Card.

class cards.cards_forms.**CollateForm**(**args*, ***kwargs*)
Form for editing Collate Card.

class cards.cards_forms.**MergeCSVForm**(**args*, ***kwargs*)
Form for editing MergeCSV Card.

class cards.cards_forms.**PreProcForm**(**args*, ***kwargs*)
Form for editing PreProc Card.

class cards.cards_forms.**QRFForm**(**args*, ***kwargs*)
Form for editing QRF Card.

class cards.cards_forms.**RFScoreForm**(**args*, ***kwargs*)
form for editing RFScore Card.

class cards.cards_forms.**RFTrainForm**(**args*, ***kwargs*)
Form for editing RFTrain Card.

class cards.cards_forms.**RandomForestForm**(**args*, ***kwargs*)
Form for editing RandomForest Card.

class cards.cards_forms.**RemapForm**(**args*, ***kwargs*)
Form for editing Remap Card.

class cards.cards_forms.**YearFilterForm**(**args*, ***kwargs*)
Form for editing YearFilter Card.

1.7.5 Cards.card_update_create

This module contains methods to refresh the data in the cards after you create and edit.

cards.card_update_create.**calcstats_update_create**(*form*, *item_id=None*)

Updated card CalcStats.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**collate_update_create**(*form*, *item_id=None*, *multiple=None*, *delete=False*)

Updated card Collate.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**mergecsv_update_create**(*form*, *item_id=None*)

Updated card MergeCSV.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**preproc_update_create**(*form*, *item_id=None*)

Updated card PreProc.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**qrf_update_create**(*form*, *item_id=None*)

Updated card QRF.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**randomforest_update_create**(*form*, *item_id=None*)

Updated card RandomForest.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**remap_update_create**(*form*, *item_id=None*)

Updated card Remap.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**rfscore_update_create**(*form*, *item_id=None*)
Updated card RFScore.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**rftrain_update_create**(*form*, *item_id=None*)
Updated card RFTrain.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

cards.card_update_create.**year_filter_update_create**(*form*, *item_id=None*)
Updated card YearFilter.

Arguments

- *form*: Object of the form
- *item_id*: ID of the card. Set when editing (the default=None when you create a card)

1.8 Gsi aplication

1.8.1 Gsi.models

class gsi.models.**Area**(**args*, ***kwargs*)
Model for the Areas.

Fields **name**: Name of a Area

tiles: Relation with the Tile model from the applications GSI

class gsi.models.**CardSequence**(**args*, ***kwargs*)
Model for the Card Sequences.

Functions The CardSequence model unites the selected cards in a sequence of cards

Fields **name**: Name the Card Sequence

environment_base: Relation with the VariablesGroup model from the applications GSI

environment_override: The Environment override for the object of the Card Sequence model

cards: Relation with the CardItem model from the applications cards

configfile: Relation with the ConfigFile model from the applications GSI

class gsi.models.**ConfigFile**(**args*, ***kwargs*)
Model for the Config Files.

Fields **pathname**: Path to the config file

description: The config file description

configuration_file: Name the configuration file

class gsi.models.**HomeVariables**(**args*, ***kwargs*)
Model for the Home Variables.

Fields **SAT_TIF_DIR_ROOT**: Satelite Data Top Level

RF_DIR_ROOT: Top directory of Random Forest Files

USER_DATA_DIR_ROOT: Top Level for user data directory

MODIS_DIR_ROOT: Top Level for raw Modis data

RF_AUXDATA_DIR: Top Level for Auxilliary data(SOIL, DEM etc.

SAT_DIF_DIR_ROOT: Top Level for Satelite TF files

class gsi.models.**InputDataDirectory**(**args*, ***kwargs*)
Model for the Input Data Directorys.

Fields **name**: Name of a Input Data Directory

full_path: Full path to the Input Data Directory

save(**args*, ***kwargs*)
The method "save" start when create a new object of the model

Create the new directory ('name' variable) at the specified path ('full_path' variable)

class gsi.models.**ListTestFiles**(**args*, ***kwargs*)
Model for the Lists Test Files.

Fields **name**: Name of a List Test Files

input_data_directory: Relation with the InputDataDirectory model from the applications GSI

size: File size

date_modified: Date modified of file

class gsi.models.**Log**(**args*, ***kwargs*)
Model for the Logs.

Fields **name**: Name the Log

log_file_path: The path to the log file

log_file: The log file name

class gsi.models.**OrderedCardItem**(**args*, ***kwargs*)
Model for the Ordered Card Items.

Functions The OrderedCardItem is an intermediary model for designed relational communication the objects of cards models with objects CardSequence model

Fields **card_item**: Relation with the CardItem model from the applications Cards

sequence: Relation with the CardSequence model from the applications GSI

order: Priority object in its CardSequence

run_parallel: Startup type parallel or not

number_sub_cards: The number of sub-cards to the card when run

class gsi.models.**Resolution**(**args*, ***kwargs*)
Model for the Resolutions.

Fields **name**: Name of a Resolution (this will be a short display of the value, i.e. 1KM, 250M)

value: Value in meters, e.g 1000 for 1KM display name

class gsi.models.**Run**(**args*, ***kwargs*)
Model for the Runs.

Functions The Run model stores information about each startup RunBase model

Fields **user**: Relation with the User model from the Django Auth model

run_base: Relation with the RunBase model from the applications GSI

state: Execution the RunBase model status

log: Relation with the Log model from the applications GSI

run_date: Execution the RunBase model date of creating

class gsi.models.**RunBase**(**args*, ***kwargs*)

Model for the Runs Base.

Functions The CardSequence model is designed to run the cards that are in the CardSequence model

Fields **name**: Name the Run Bas

author: Relation with the User model from the Django Auth model

description: RunBase description

purpose: Purpose of Run

directory_path: Directory path

resolution: Resolution

card_sequence: Relation with the CardSequence model from the applications GSI

date_created: The RunBase object date created

date_modified: The RunBase object date modified

save(**args*, ***kwargs*)

The method 'save' records the modified date when the object was changed

class gsi.models.**RunStep**(**args*, ***kwargs*)

Model for the Runs Steps.

Functions RunStep model stores information about the cards after execute RunBase model

Fields **parent_run**: Relation with the Run model from the applications GSI

card_item: Relation with the OrderedCardItem model from the applications GSI

state: Status executed of the card work

start_date: Date of the run of card

get_next_step()

Gets the next card. If it is, then return the card object. If not - False

class gsi.models.**Satellite**(**args*, ***kwargs*)

Model for the Satellites.

Fields **name**: Name of a Satellite

class gsi.models.**SubCardItem**(**args*, ***kwargs*)

Model for the Sub-Card Items.

Functions The SubCardItem model unites the syb-cards with its main the card to the parallel start.

Fields **name**: Name the Sub-Card Item

state: Status executed of the card work

run_id: RUN ID's

card_id: CARD ID's

start_date: Date of the run of syb-card

start_time: Time of the run of syb-card

class gsi.models.**Tile**(**args*, ***kwargs*)
Model for the Tiles.

Fields **name**: Name of a Tile

class gsi.models.**TileType**(**args*, ***kwargs*)
Model for the Tile Types.

Fields **name**: Name of a Tile Type

class gsi.models.**VariablesGroup**(**args*, ***kwargs*)
Model for the Variables Groups.

Fields **name**: Name of a Variables Group

environment_variables: List of the Variables

class gsi.models.**Year**(**args*, ***kwargs*)
Model for the Years.

Fields **name**: Name of a Year

class gsi.models.**YearGroup**(**args*, ***kwargs*)
Model for the Year Groups.

Fields **name**: Name of a Tile

years: Relation with the Year model from the applications GSI

1.8.2 Gsi.views

gsi.views.**area_add**(*request*, **args*, ***kwargs*)
View for the "Area Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**area_edit**(*request*, **args*, ***kwargs*)
View for the "Area "<name>" Edit" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *area_id:* The Area object ID

gsi.views.**areas**(*request*, **args*, ***kwargs*)
View for the "Areas" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**audit_history**(*request*, **args*, ***kwargs*)
View for the "Audit record for '<name>'" page.

Arguments

- *request:* The request is sent to the server when processing the page

- *run_id:* The RunBase object

gsi.views.**card_sequence_update**(*request*, **args*, ***kwargs*)
View the CardSequence object for editing in the creation of the Run.

Arguments

- ***request:*** **The request is sent to the server when processing the page**
  - *run_id:* The RunBase object ID
- *cs_id:* The CardSequence object ID

gsi.views.**cards_list**(*request*, **args*, ***kwargs*)
View for the "Cards List" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**copy_runbase**(*request*, *name*)
Copy the RunBase object.

Functions The method creates a copy of an existing of the RunBase object.

Arguments

- *name*: The RunBase object name
- *request:* The request is sent to the server when processing the page

gsi.views.**customer_section**(*request*, **args*, ***kwargs*)
View for the "Customer '<user>' section" page.

Functions When you load the page is loaded map with Google MAP. Initial coordinates: eLat = 0, eLng = 0. Zoom map is variable GOOGLE_MAP_ZOOM, whose value is in the project settings. Code view allows to change position when you enter values in the fields on the page "Enter Lat" and "Enter Log".

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**environment_group_add**(*request*, **args*, ***kwargs*)
View for the "Environment Group Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**environment_group_edit**(*request*, **args*, ***kwargs*)
View for the "Environment Group "<name>" Edit" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *env_id:* The VariablesGroup object ID

gsi.views.**environment_groups**(*request*, **args*, ***kwargs*)
View for the "Environment Groups" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**get_number_cards**(*rb*, *user*)
Get the number of cards in the object RunBase.

Functions The method creates a copy of an existing of the CardSequence object.

Arguments

- *rb*: The RunBase object
- *user*: Current the user

gsi.views.**handle_uploaded_file**(*f*, *path*)
Upload file on the server.

Arguments

- *f*: File name
- *path*: Path where to save the file

gsi.views.**home_variable_setup**(*request*, **args*, ***kwargs*)
View for the "Home Variables Setup" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**index**(*request*, **args*, ***kwargs*)
View for the Main page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**input_data_dir_add**(*request*, **args*, ***kwargs*)
View for the "Input Data Directory Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**input_data_dir_edit**(*request*, **args*, ***kwargs*)
View for the "Input Data Directory Edit "<name>"" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *dir_id:* The InputDataDirectory object ID

gsi.views.**input_data_dir_list**(*request*, **args*, ***kwargs*)
View for the "Input Data Directory" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**new_run**(*request*, **args*, ***kwargs*)
View for the "New Run" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**resolution**(*request*, **args*, ***kwargs*)
View for the "Resolutions" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**resolution_add**(*request*, **args*, ***kwargs*)
View for the 'Resolution Add' page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**resolution_edit**(*request*, **args*, ***kwargs*)
View for the "Resolution Edit '<name>'" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *resolution_id:* The Resolution object ID

gsi.views.**run_details**(*request*, **args*, ***kwargs*)
View the details executed of run of the cards.

Arguments

- ***request:* The request is sent to the server when processing the page**
  - *run_id:* The RunStep object ID

gsi.views.**run_progress**(*request*, **args*, ***kwargs*)
View for the "View Run Progress" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**run_setup**(*request*, **args*, ***kwargs*)
View for the "Setup New Run" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**run_update**(*request*, **args*, ***kwargs*)
View for the "Upload Test Data" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id:* The RunBase object ID

gsi.views.**satellite**(*request*, **args*, ***kwargs*)
View for the "Satellite" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**satellite_add**(*request*, **args*, ***kwargs*)
View for the "Satellites Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**satellite_edit**(*request*, **args*, ***kwargs*)
View for the "Satellite Edit '<name>' page.

Arguments

- *request:* The request is sent to the server when processing the page

- *satellite_id:* The Satellite object ID

gsi.views.**sub_card_details**(*request*, **args*, ***kwargs*)
View the details executed of run of the cards.

Arguments

- ***request:*** **The request is sent to the server when processing the page**
  - *run_id:* The RunStep object ID
- *card_id:* The CardItem object ID

gsi.views.**submit_run**(*request*, **args*, ***kwargs*)
View for the "Submit a Run" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**tile_add**(*request*, **args*, ***kwargs*)
View for the "Tile Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**tile_edit**(*request*, **args*, ***kwargs*)
View for the "Tile Edit '<name>'" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *tile_id:* The Tile object ID

gsi.views.**tiles**(*request*, **args*, ***kwargs*)
View for the "Tiles" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**update_qrf_rftrain_card**(*cs*, *cs_cards*)
Update the QRF and RFtrain cards.

Functions The method takes the CardSequence model object and list of cards of CardSequence object. If the card QRF, it is written to the path of configfile CardSequence object. If the card RFtrain, it is written the name of the configfile of the CardSequence object.

Arguments

- *cs*: The CardSequence object
- *cs_cards*: List of the cards from the CardSequence object

gsi.views.**upload_file**(*request*, **args*, ***kwargs*)
View for the "Upload Test Data" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**view_log_file**(*request*, **args*, ***kwargs*)
View details of the files *.err and *.out of the cards.

Arguments

- ***request:*** **The request is sent to the server when processing the page**
  - *run_id:* The RunStep object ID
- *card_id:* The Card object ID
- *status:* Status executed to work of the cards

gsi.views.**view_log_file_sub_card**(*request*, **args*, ***kwargs*)
View details of the files *.err and *.out of the sub-cards.

Arguments

- ***request:*** **The request is sent to the server when processing the page**
  - *run_id:* The RunStep object ID
- *card_id:* The Card object ID
- *count:* The number of sub-cards
- *status:* Status executed to work of the cards

gsi.views.**view_results**(*request*, **args*, ***kwargs*)
View for the "View results '<name>"' page.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id:* The RunBase object ID

gsi.views.**view_results_folder**(*request*, **args*, ***kwargs*)
View for the "View results '<name>"' page.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id:* The RunBase object ID
- *prev_dir:* The prev directory
- *dir:* The current directory

gsi.views.**write_card_to_cs**(*card_sequence*, *query*)
Update the QRF and RFtrain cards.

Functions Method writes an object CardSequence card objects.

Arguments

- *card_sequence*: The CardSequence object
- *query*: request POST

gsi.views.**year_add**(*request*, **args*, ***kwargs*)
View for the "Year Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**year_edit**(*request*, **args*, ***kwargs*)
View for the "Year Edit '<name>"' page.

Arguments

- *request:* The request is sent to the server when processing the page

- *year_id:* The Year object ID

gsi.views.**years**(*request*, **args*, ***kwargs*)
View for the "Years" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**years_group**(*request*, **args*, ***kwargs*)
View for the "Years Groups" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**years_group_add**(*request*, **args*, ***kwargs*)
View for the "Years Group Add" page.

Arguments

- *request:* The request is sent to the server when processing the page

gsi.views.**years_group_edit**(*request*, **args*, ***kwargs*)
View for the "Years Group "<name>" Edit" page.

Arguments

- *request:* The request is sent to the server when processing the page
- *yg_id:* The YearGroup object ID

1.8.3 Gsi.views_card_runid_csid

gsi.views_card_runid_csid.**cs_runid_csid_calcstats_edit**(*request*, **args*, ***kwargs*)
View for to edit CalcStats card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *calcstats_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_collate_edit**(*request*, **args*, ***kwargs*)
View for to edit Collate card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *collate_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_mergecsv_edit**(*request*, **args*, ***kwargs*)
View for to edit MergeCSV card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *mcsv_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_preproc_edit**(*request*, **args*, ***kwargs*)

View for to edit PreProc card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *preproc_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_qrf_edit**(*request*, **args*, ***kwargs*)

View for to edit QRF card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *qrf_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_randomforest_edit**(*request*, **args*, ***kwargs*)

View for to edit RandomForest card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *rf_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_remap_edit**(*request*, **args*, ***kwargs*)

View for to edit Remap card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model

- *remap_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_rfscore_edit**(*request*, **args*, ***kwargs*)
View for to edit RFScore card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *rfscore_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_rftrain_edit**(*request*, **args*, ***kwargs*)
View for to edit RFTrain card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *rftrain_id*: ID of the object of the CardItem model

gsi.views_card_runid_csid.**cs_runid_csid_year_filter_edit**(*request*, **args*, ***kwargs*)
View for to edit YearFilter card of the CardSequence model.

Arguments

- *request:* The request is sent to the server when processing the page
- *run_id*: ID of the RunBase
- *cs_id*: ID of the CardSequence
- *card_id*: ID of the card of the CardSequence model
- *yf_id*: ID of the object of the CardItem model

1.8.4 Gsi.gsi_forms

This module contains methods for creating and editing gsi forms.

class gsi.gsi_forms.**AreasForm**(**args*, ***kwargs*)
Form for editing Areas.

class gsi.gsi_forms.**CardSequenceCardForm**(**args*, ***kwargs*)
Form for editing CardSecuence.

class gsi.gsi_forms.**CardSequenceCreateForm**(**args*, ***kwargs*)
Form for editing CardSecuence.

class gsi.gsi_forms.**CardSequenceForm**(**args*, ***kwargs*)
Form for editing CardSecuence.

class gsi.gsi_forms.**ConfigFileForm**(**args*, ***kwargs*)
Form for editing ConfigFile.

class gsi.gsi_forms.**EnvironmentGroupsForm**(**args*, ***kwargs*)
Form for editing Environment Groups.

class gsi.gsi_forms.**HomeVariablesForm**(**args*, ***kwargs*)
Form for editing Home Variables.

class gsi.gsi_forms.**InputDataDirectoryForm**(**args*, ***kwargs*)
Form for editing InputDataDirectory.

class gsi.gsi_forms.**ResolutionForm**(**args*, ***kwargs*)
Form for editing Resolution.

class gsi.gsi_forms.**RunForm**(**args*, ***kwargs*)
Form for editing RunBase.

class gsi.gsi_forms.**SatelliteForm**(**args*, ***kwargs*)
Form for editing Satellite.

class gsi.gsi_forms.**TileForm**(**args*, ***kwargs*)
Form for editing Tile.

class gsi.gsi_forms.**YearForm**(**args*, ***kwargs*)
Form for editing Year.

class gsi.gsi_forms.**YearGroupForm**(**args*, ***kwargs*)
Form for editing YearGroup.

1.8.5 Gsi.gsi_update_create

This module contains methods to refresh the data in the gsi objects after you create and edit.

gsi.gsi_update_create.**area_update_create**(*form*, *multiple=None*, *item_id=None*, *delete=False*)
Updated Area model.

Arguments

- *form*: Object of the form
- *multiple*: Transmited a list of objects
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)
- *delete*: Boolean value determine the remove objects or not

gsi.gsi_update_create.**configfile_update_create**(*pathname*)
Updated ConfigFile model.

Arguments

- *pathname*: Path name

gsi.gsi_update_create.**create_update_card_sequence**(*form*, *configfile=None*, *cs_id=None*)
Updated CardSequence model.

Arguments

- *form*: Object of the form
- *multiple*: Transmited a list of objects
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)
- *delete*: Boolean value determine the remove objects or not

gsi.gsi_update_create.**data_dir_update_create**(*form*, *item_id=None*)
Updated InputDataDirectory model.

Arguments

- *form*: Object of the form
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)

gsi.gsi_update_create.**resolution_update_create**(*form*, *item_id=None*)
Updated Resolution model.

Arguments

- *form*: Object of the form
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)

gsi.gsi_update_create.**satellite_update_create**(*form*, *multiple=None*, *item_id=None*, *delete=False*)
Updated Satellite model.

Arguments

- *form*: Object of the form
- *multiple*: Transmited a list of objects
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)
- *delete*: Boolean value determine the remove objects or not

gsi.gsi_update_create.**tile_update_create**(*form*, *item_id=None*)
Updated Tile model.

Arguments

- *form*: Object of the form
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)

gsi.gsi_update_create.**var_group_update_create**(*form*, *item_id=None*)
Updated VariablesGroup model.

Arguments

- *form*: Object of the form
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)

gsi.gsi_update_create.**year_update_create**(*form*, *item_id=None*)
Updated Year model.

Arguments

- *form*: Object of the form
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)

gsi.gsi_update_create.**yg_update_create**(*form*, *multiple=None*, *item_id=None*, *delete=False*)
Updated YearGroup model.

Arguments

- *form*: Object of the form
- *multiple*: Transmited a list of objects
- *item_id*: ID of the object. Set when editing (the default=None when you create a card)

- *delete*: Boolean value determine the remove objects or not

1.8.6 Gsi.signals

This module contains the signals that handle some types of events that are generated by the system.

gsi.signals.**added_update_area_for_each_tile**(*sender*, *instance*, ***kwargs*)
When created new Tile object then creating a new Area object and in the field "tile" of Area models added the new Tiles object.

gsi.signals.**create_cs**(*sender*, *instance*, *created*, ***kwargs*)
When created the new RunBase object than creating new the CardSequence object.

gsi.signals.**log_it_run**(*sender*, *instance*, ***kwargs*)
Writes the log information on the status of the object Run.

gsi.signals.**log_it_runbase**(*sender*, *instance*, *created*, ***kwargs*)
Write to log information about the change in RunBase object.

gsi.signals.**mkdir**(*sender*, *instance*, ***kwargs*)
When created new InputDataDirectory object then creating new directory with name new_object.name .

gsi.signals.**remove_empty_area_by_removing_tile**(*sender*, *instance*, ***kwargs*)
When removing Tiles object then removed Area object.

1.8.7 Gsi.tests

This module contains the tests for the gsi app.

class gsi.tests.test_signals.**GsiSignalsTests**(*methodName='runTest'*)
Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

setUp()
We set the initial data.

test_added_update_area_for_each_tile()
The test signal that adds a new tile to the Area model when creating the new tiles

test_remove_empty_area_by_removing_tile()
The test signal that removes the object Area if removing all the tiles from the object Area

class gsi.tests.test_models.**ModelsTestCase**(*methodName='runTest'*)
Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

setUp()
We set the initial data.

test_area_model()
Testing start of the Area model.

test_cardsequence_model()
Testing start of the CardSequence model.

test_configfile_model()
Testing start of the ConfigFile model.

test_homevariable_model()
Testing start HomeVariables model.

Checking with the creation of the test object initialization. Since the model is SingletonModel, then check to create another object model.

test_input_data_directory_model()
Testing start of the InputDataDirectory model.

When created model object InputDataDirectory create automatically the m2m field "full_path".

The variable for the "full_path" field geting from the variable SAT_TIF_DIR_ROOT of the object of the HomeVariables model and the variable "name" of the object of the InputDataDirectory model

test_list_test_files_model()
Testing start of the ListTestFiles model.

test_log_model()
Testing start of the Log model.

test_resolution_model()
Testing start of the Resolution model.

test_runbase_model()
Testing start of the TileType model.

test_satellite_model()
Testing start of the Satellite model.

test_tile_model()
Testing start of the Tile model.

test_tiletype_model()
Testing start of the TileType model.

test_variablesgroup_model()
Testing start of the VariablesGroup model.

test_year_model()
Testing start of the Year model.

test_yeargroup_model()
Testing start of the YearGroup model.

1.9 Log aplication

1.9.1 Log.models

class log.models.**Log**(**args*, ***kwargs*)
Model for the system logs.

Fields **user**: Current user

element: Model name

element_id: Element model id

message: Message for the log file

at: Date create

class log.models.**LogDebug**(**args*, ***kwargs*)

Model for the debug logs.

Fields name: Object name

log: Message for the log file

create_date: Date create

1.9.2 Log.logger

This module provides methods for logging.

log.logger.**get_logs**(*element*, *element_id*, *limit=None*, *user=None*)

Message get from the log file.

Arguments

- *element*: Model name
- *element_id*: Element model id
- ***limit*: The number of latest messages you want to receive**
  - *user*: Object user

log.logger.**log_it**(*user*, *element*, *element_id*, *message*)

Message write to the log file.

Arguments

- *user*: Current user
- *element*: Model name
- *element_id*: Element model id
- *message*: Message for the log file

1.10 Core package

Package with auxiliary functions.

1.10.1 Core.copy_card

The module for copying cards for copying run.

core.copy_card.**calcstats_copy**(*card*)

The method copies the CalcStats card.

Arguments

- *name*: The card name

core.copy_card.**collate_copy**(*card*)

The method copies the Collate card.

Arguments

- *name*: The card name

core.copy_card.**create_copycard**(*card*, *card_type*)
The method determines the type of cards which function to copy the run.

Arguments

- *card*: The card name
- *card_type*: The card type

core.copy_card.**get_copy_name_card**(*name*, *model*)
The method creates a new name for the copied card.

Arguments

- *name*: The card name
- *model*: The Card model object

core.copy_card.**mergecsv_copy**(*card*)
The method copies the MergeCSV card.

Arguments

- *name*: The card name

core.copy_card.**preproc_copy**(*card*)
The method copies the PreProc card.

Arguments

- *name*: The card name

core.copy_card.**qrf_copy**(*card*)
The method copies the QRF card.

Arguments

- *name*: The card name

core.copy_card.**randomforest_copy**(*card*)
The method copies the RandomForest card.

Arguments

- *name*: The card name

core.copy_card.**remap_copy**(*card*)
The method copies the Remap card.

Arguments

- *name*: The card name

core.copy_card.**rfscore_copy**(*card*)
The method copies the RFScore card.

Arguments

- *name*: The card name

core.copy_card.**rftrain_copy**(*card*)
The method copies the RFTrain card.

Arguments

- *name*: The card name

core.copy_card.**yearfilter_copy**(*card*)

The method copies the YearFilter card.

Arguments

- *name*: The card name

1.10.2 Core.get_post

The module for processing POST request.

core.get_post.**add_card_in_cardsequence**(*card*, *cs_id*)

The method added the new Card object and its order in the CardSequence object.

Arguments

- *card*: The Card object
- *cs_id*: The CardSequence object ID

core.get_post.**cs_cards_update**(*form*, *cs_card*, *card_item*)

The method updates the value of the cards in CardSequence object.

Arguments

- *form*: The form object
- *cs_card*: The OrderedCardItem object
- *card_item*: The Card object

core.get_post.**get_post**(*request*, *item_form*, *item*, *reverse_ulr*, *func*, *args=False*, *item_id=None*, *cs_form=False*, *cs_id=None*)

The method processes the POST request and sends a response from the server.

Arguments

- *request*: The request to the server
- *item_form*: The Card form
- *item*: The Card object
- *reverse_ulr*: The URL redirection
- *func*: The function to create or update Card
- *args*: Additional options: Run ID, CardSequence ID, Card ID
- *item_id*: The Card object ID (if the object is created the value default "None")
- *cs_form*: If an object is created or updated in CardSequence object, the value is "True". Default is "False"
- *cs_id*: If an object is created or updated in CardSequence object, the value is CardSequence object ID. Default is "None"

1.10.3 Core.multithreaded

The module for multi-threaded processing cards.

class core.multithreaded.**MultiprocessingCards**(*queue*, *flag='cards'*)

Class receives a card that you want to run. It creates a queue and starts a multithreaded cards.

multiprocessing_cards(*param*)
The method receives the parameters and starts the implementation process for each type.

run()
Method run multi-threaded process.

1.10.4 Core.paginations

The package for the implementation of pagination.

core.paginations.**paginations**(*request*, *model_name*)
The method generates a page-pagination.

Arguments

- *request*: The request form the server
- *model_name*: The Model object

1.10.5 Core.validator_gsi

The package for the validation fields of models.

core.validator_gsi.**validate_order**(*value*)
The method makes the validation of numerical field on a positive number.

Arguments

- *value*: Input value

1.10.6 Core.utils

Package with additional functionality.

class core.utils.**UnicodeNameMixin**
Class inheritance for other classes of models.

Functions When inheriting displays the model name

core.utils.**convert_size_file**(*size*)
Method convert a numeric value of the file size in the text designations: B, KB, MB.

core.utils.**copy_file**(*src*, *dest*, *card_name*)
The method copies the file specification.

core.utils.**create_new_folder**(*dir*)
The method to create a new directory for the model Input Data Directory

core.utils.**create_open_master_script**(*path_runs*, *card_id*, *num*)
The method creates a script file and set it permissions 777.

core.utils.**create_scripts**(*run*, *sequence*, *card*, *step*)
The method for create a scripts at startup RunBase object for the each cards.

core.utils.**create_sub_card_item**(*name*, *run_id*, *card_id*)
The method creates a new object model SubCardItem.

core.utils.**create_sub_dir**(*path*)
Method to create directorys: Results, Scores, Trees.

core.utils.**create_symlink**(*src*, *dest*, *path*)
The method creates a symlink to transmit directory.

Functions Check whether there is already such a symlink. If not, create.

Arguments

- *src*: Folder which do symlink
- *dest*: The folder from which make a symlink
- *path*: The path for check the existence of symbolic link

core.utils.**execute_fe_command**(*params*, *flag='cards'*)
The method returns the principal name.

Functions It is used when creating a copy card in the object CardSequence.

Arguments

- *params*: Current options for the execute
- *flag*: Specifies to which the entity of the process is performed is started

core.utils.**get_area_tiles**(*name*)
The method geting all tiles from an object Area.

core.utils.**get_card_model**(*card_model*, *card_name*)
The method geting type model for the card.

core.utils.**get_copy_name**(*name*)
The method returns the principal name.

Functions It is used when creating a copy card in the object CardSequence.

Arguments

- *name*: Card name

core.utils.**get_dir_root_static_path**()
Method for get the symbolic link.

core.utils.**get_executable**(*run*, *sequence*, *card*, *card_item*)
The method gets a value for the variable EXECUTABLE for the each card for the create_scripts method.

core.utils.**get_files**(*path*, *file_extension*)
The method to get a list of files from a specified directory and file extension.

Returns a list of files and errors in the preparation of a list of files

Arguments

- *path*: The path to the file
- *file_extension*: The file extension

core.utils.**get_files_dirs**(*url_path*, *full_path*)
The method to get a list of all files and directories from a given path.

core.utils.**get_path_folder_run**(*run*)
The method receives to the logs folder a run.

core.utils.**get_statistical_method**(*remap_obj*)
The method geting statistical method.

core.utils.**get_type_file**(*mime_type*)
The method determines the mime-type of file: image, text, pdf, msword, doc, archive.

core.utils.**get_years**(*name*)
The method geting all year from an object YearGroup.

core.utils.**is_run_parallel**(*card*)
The method checks the type of start-up cards: parallel or in series.

core.utils.**make_run**(*run_base*, *user*)
The method starts work on card processing script.

core.utils.**slash_remove_from_path**(*path*)
The method removes superfluous slashes in path.

Functions Checks whether there is in the transmission path of the extra slashes. If found, it replaces them with the standard one.

Arguments

- *path*: Path (string)

core.utils.**update_list_dirs**()
The method updates the list of files and directory for the ListTestFiles and the InputDataDirectory models.

core.utils.**update_list_files**(*obj_dir*)
The method updates the list of files from a specified directory to the model ListTestFiles.

core.utils.**update_root_list_files**()
The method updates the list of files to ListTestFiles model.

core.utils.**validate_status**(*status*)
The method makes the validation status for the API.

Functions Checks whether there is obtaining the status of the list of possible statuses.

Arguments

- *status*: Current status

core.utils.**write_log**(*log_name*, *run*, *path_log*)
The method writes a Log model GSI app.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

a

c

g

l

A

C

R

S

T

U

V

W

Y