Ettema Lab Information Management System Documentation

Release 0.1

Ino de Bruijn, Lionel Guy

November 28, 2014

Contents

ETTLIMS is a Lab Information Management System developed for the Ettema Lab by BILS.

This documentation has the following sections:

Pipeline Design Describes the workflow at the Ettema Lab

Database design Describes how the workflow has been implemented in the database

Naming Scheme The naming scheme for barcodes and UID for objects in the database

User guide A user guide for users of the system

Installation How to install the LIMS

Developer guide A guide for developers

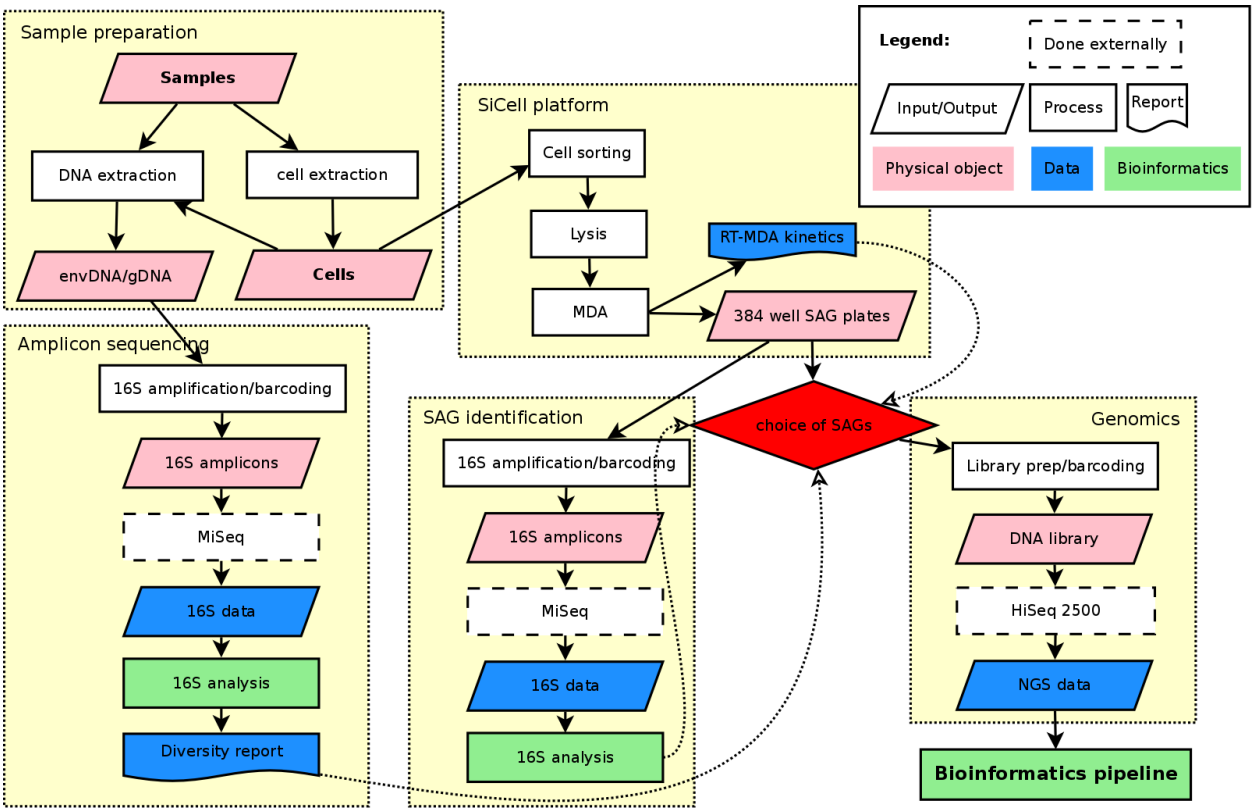
Other links:

- Documentation: http://ettlims.rtfd.org
- Demo: http://bit.ly/scglims-heroku (for admin user/pass inodb)
- Presentation: http://bit.ly/limstalk
- GitHub: https://github.com/BILS/ETTLIMS/
- Free software: GPLv3 License

Contents:

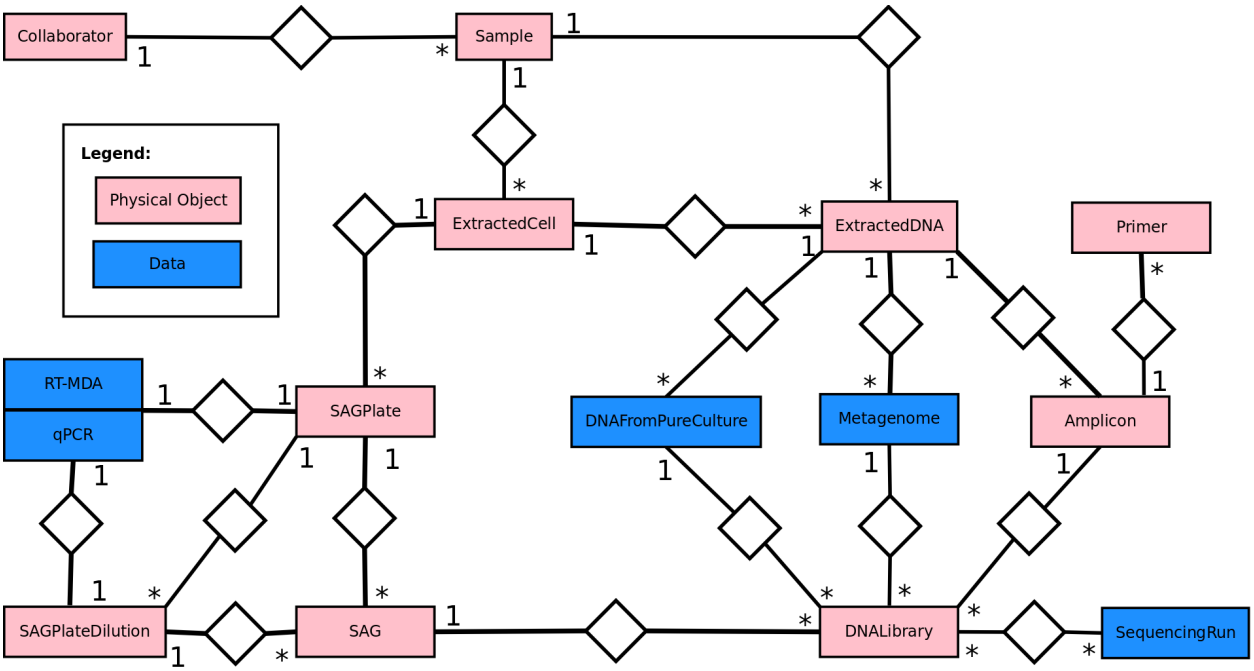
CHAPTER 1

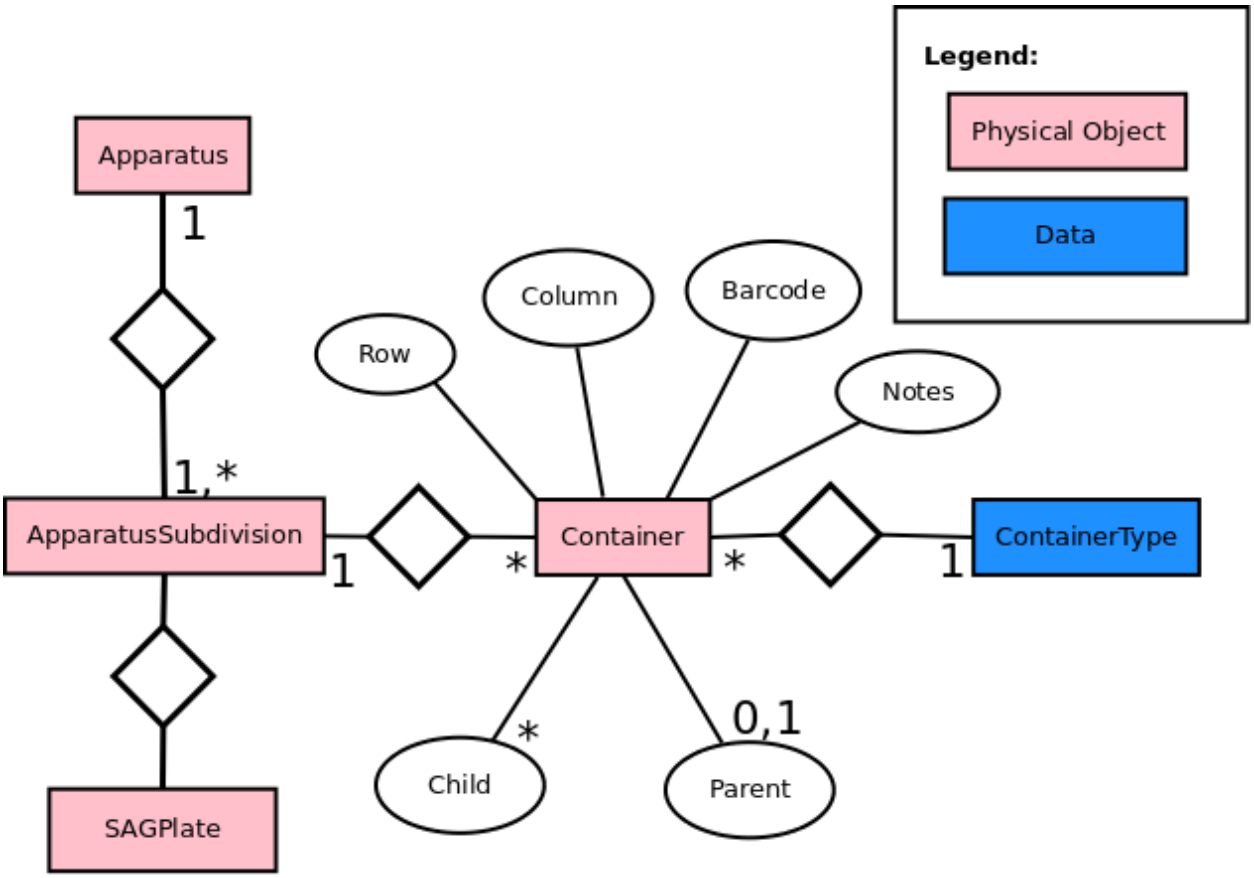
Pipeline Design



CHAPTER 2

Database design




Legend:
Physical Object
Data
Apparatus
1
1,*
ApparatusSubdivision
1
*
Container
*
1
ContainerType
Row
Column
Barcode
Notes
Child
*
0,1
Parent
SAGPlate

CHAPTER 3

Naming Scheme

The constraints are:

- Naming should be stable: it should never change. This implies that enough room should be given to account for many years of future samples
- Name should be informative: one should get an idea of where the gene/SAG comes from from its name.
- Name should be concise: ideally, a gene name (which is the end product in a way) should be kept under 20 characters

Quick guide:

```
sample well asbly reserved
______  ___  _     _
O331AC_C09A_A002340
      -    -  -----
 plate    run gene
```

3.1 Complete description

- ***Sample*:**
  - Five characters, alphanumeric, letter in capitals
  - Should be as informative as possible (e.g. O331A for Ocean Drilling program exp. 331, sample A)
- ***Plates*:**
  - Start with sample ID
  - One extra character (A-Z) to indicate the nth plate from this sample (e.g. O331AC for the third plate sorted from sample O331A)
  - The same extra character (a-z and 0-9) for plate dilutions (e.g. O331Ab for the second plate dilution from sample O331A)
  - For metagenomes, this should indicate different extractions from the same sample.
- ***Well/Metagenome/Pure culture/Amplicons*:**
  - Starts with plate ID, then an underscore
  - Three characters in addition: one letter (A-L) and two digits (00-16) to describe the well.
  - Metagenomes are labeled with X, then 01. Further digits are reserved for later use.

  - Pure cultures are labeled with Z, then 01. Further digits are reserved for later use.
  - Amplicons are labeled with Y, then 01. Further digits are reserved for later use.
  - A1 should be always written A01
  - E.g. O331AC_C09

- ***DNA library***
  - Describes many DNA libraries from the same source: can be different dilution plates, different sequencing runs, different preparations, for example.
  - Starts with the well ID
  - One character (A-Z; 0-9), in order
  - E.g. O331AC_C09A

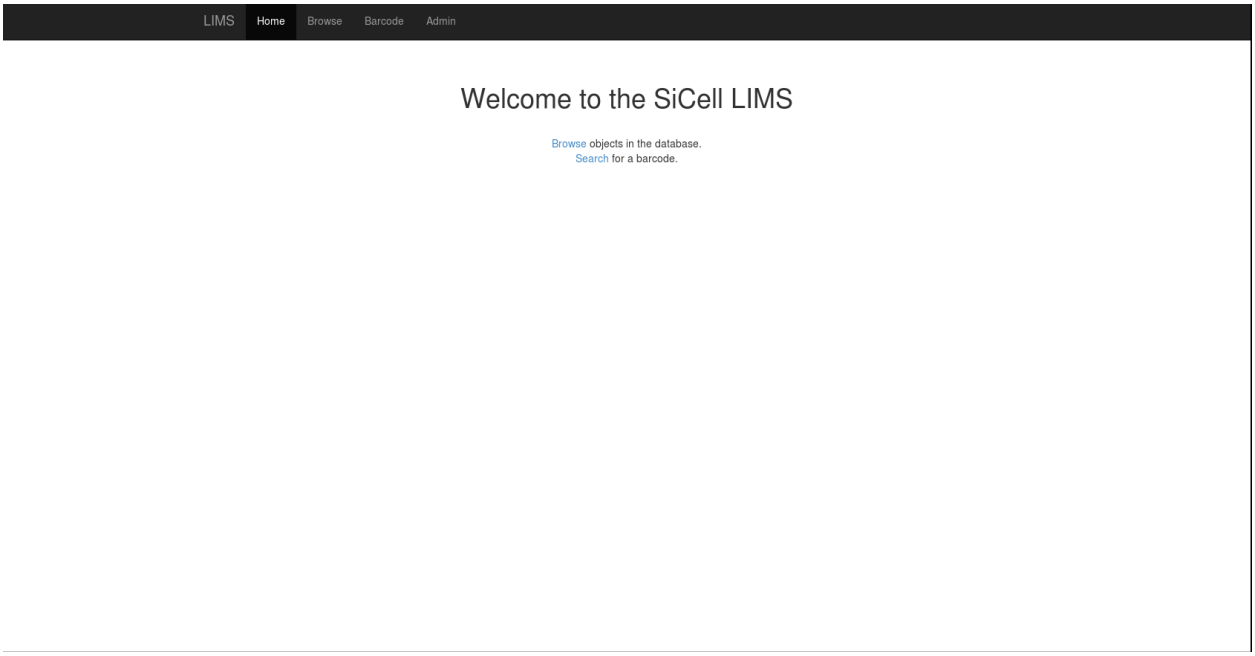
- ***Assembly***
  - Describes one assembly performed on one dataset from one run (SAG)
  - Starts with run ID, then an underscore
  - One character (A-Z; 0-9), in order
  - E.g. O331AC_C09A_B

- ***Genes*:**
  - Starts with assembly ID
  - 6 digits, with a 10 increment to allow adding genes in the middle
  - tRNAs have t prefix, rRNAs a r instead of the first digit
  - E.g. O331AC_C09A_B002340, O331AC_C09A_Bt05610

CHAPTER 4

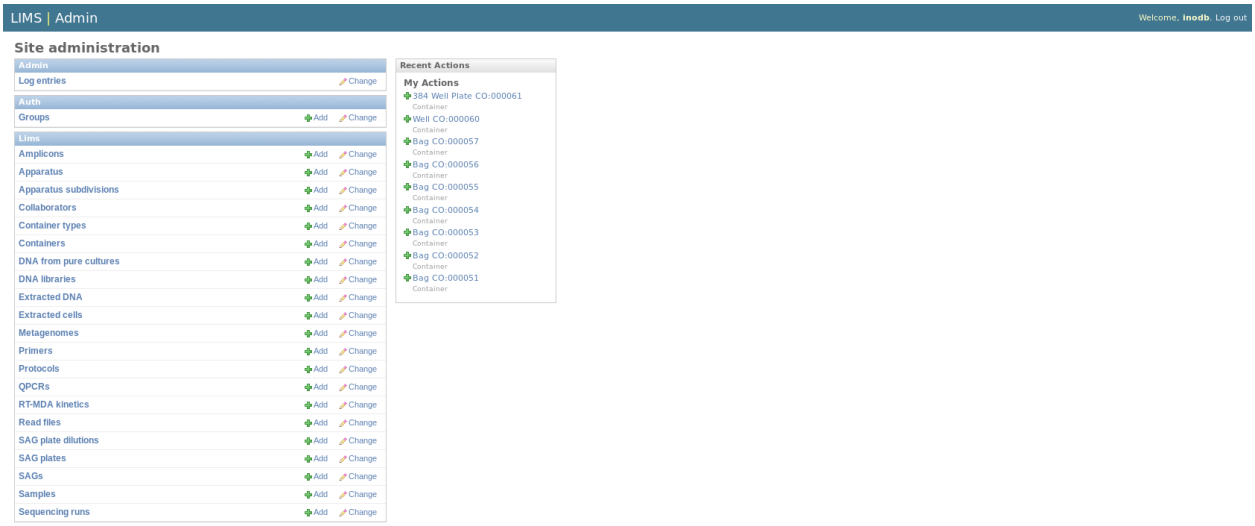
User guide

4.1 Viewing area



The viewing area currently just gives a simple listing of the different objects in the database. The main feature of the viewing area is the ability to see a tree of all related objects with a sample of choice at the root. The listing of the objects is more limited than the admin area, so for browsing objects in the database it is still more user-friendly to use the admin area.

4.2 Admin area



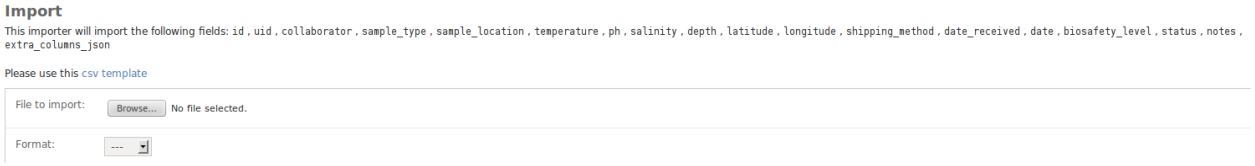
The admin area is where one actually inputs information into the database. Adding, editing, sorting and listing of objects is possible.

4.2.1 Bulk import

The Container and Sample tables have an option to bulk import multiple objects in one go with a csv, json or excel file. This is only recommended for advanced users, because you skip a lot of extra checks done that you would normally have when you fill in the forms. The importer's error messages can be rather cryptic. That being said, here is how to do it.

Import a bunch of Samples in one go

1. Go to the import page and download the csv template



2. Edit the csv template.

id	uid	collaborator	sample_type	sample_location	temperature	ph	salinity	depth	gps	shipping_method	storage_medium	date_received	date	biosafety_level	status	notes
	C001P	9	5	4				1711	22-03.302 119-46.639			2014-02-1 00:00:00	2013-07-19 00:00:00			
	C002P	9	5	4				1635	22-03.23 119-47.04			2014-02-1 00:00:00	2013-07-20 00:00:00			
	C003P	9	5	4				1399	22-03.32 119-47.52			2014-02-1 00:00:00	2013-07-20 00:00:00			
	C004P	9	5	4				1334	22-03.39 119-47.99			2014-02-1 00:00:00	2013-07-20 00:00:00			
	C005P	9	5	4				1326	22-03.40 119-48.06			2014-02-1 00:00:00	2013-07-21 00:00:00			
	C006P	9	5	4				1411	22-03.48 119-48.33			2014-02-1 00:00:00	2013-07-21 00:00:00			
	C007P	9	5	4				1575	22-03.57 119-48.75			2014-02-1 00:00:00	2013-07-21 00:00:00			
	C008P	9	5	4				1638	22-03.73 119-49.59			2014-02-1 00:00:00	2013-07-21 00:00:00			
	C009P	9	5	4				1398	22-04.15 119-47.81			2014-02-1 00:00:00	2013-07-21 00:00:00			
	C010P	9	5	4				1308	22-03.04 119-48.10			2014-02-1 00:00:00	2013-07-22 00:00:00			
	C012P	9	5	4				2893	21-43.93 119-47.67			2014-02-1 00:00:00	2013-07-22 00:00:00			
	C016P	9	5	4				1289	22-13.184 119-54.356			2014-02-1 00:00:00	2013-07-23 00:00:00			
	C018P	9	5	4				1378	22-14.578 119-54.408			2014-02-1 00:00:00	2013-07-24 00:00:00			
	C019P	9	5	4				1316	22-14.035 119-54.714			2014-02-1 00:00:00	2013-07-24 00:00:00			
	C020P	9	5	4				1294	22-16.17 119-55.04			2014-02-1 00:00:00	2013-07-24 00:00:00			
	C021P	9	5	4				1321	22-15.10 119-54.93			2014-02-1 00:00:00	2013-07-24 00:00:00			

The columns that link to other tables should contain the actual id. You can look them up in their corresponding tables. For instance for the `sample_type` you can look at the `Sample types` listing:

ID	Name
7	ocean water
6	water/biofilm
5	ocean sediments
4	Staphylococcus gut DNA
3	Helicobacter free DNA
2	E. Coli
1	lake water

and for `collaborator` in the `Collaborator` listing:

ID	First name
9	Courtney
8	Pauli
7	Leeroy
6	Das
5	Dennis
4	Friend
3	SiCell
2	Steve
1	John

The new ids for the samples don't have to be filled in, since those will be generated by the system itself.

After the `extra_columns_json` you can add as many columns as you like. These will be stored in the `extra_columns_json` field in JSON format on import.

status	notes	extra_columns_json	Sample name old	Length of piston core (cm)	Characteristics	Others
			OYEAH C1P	191	Little ethanol amount at bottom of core	
			OYEAH C2P	110	No ethanol at the bottom of core	samples selected for metatranscriptomic sequencing
			OYEAH C3P	135	High ethanol amount at bottom of core	
			OYEAH C4P	37	Little ethanol amount at bottom of core	
			OYEAH C5P	449	No ethanol at the bottom of core	
			OYEAH C6P	186		
			OYEAH C7P	357		
			OYEAH C8P	345		
			OYEAH C9P	423		
			OYEAH C10P	346	High ethanol amount at bottom of core	
			OYEAH C12P	193	High ethanol amount at bottom of core	samples selected for metatranscriptomic sequencing
			OYEAH C16P	310	No ethanol at the bottom of core	
			OYEAH C18P	436	High ethanol amount at bottom of core	
			OYEAH C19P	478	No ethanol at the bottom of core	
			OYEAH C20P	193	No ethanol but high CO2 at bottom of core	
			OYEAH C21P	210	No ethanol at the bottom of core	

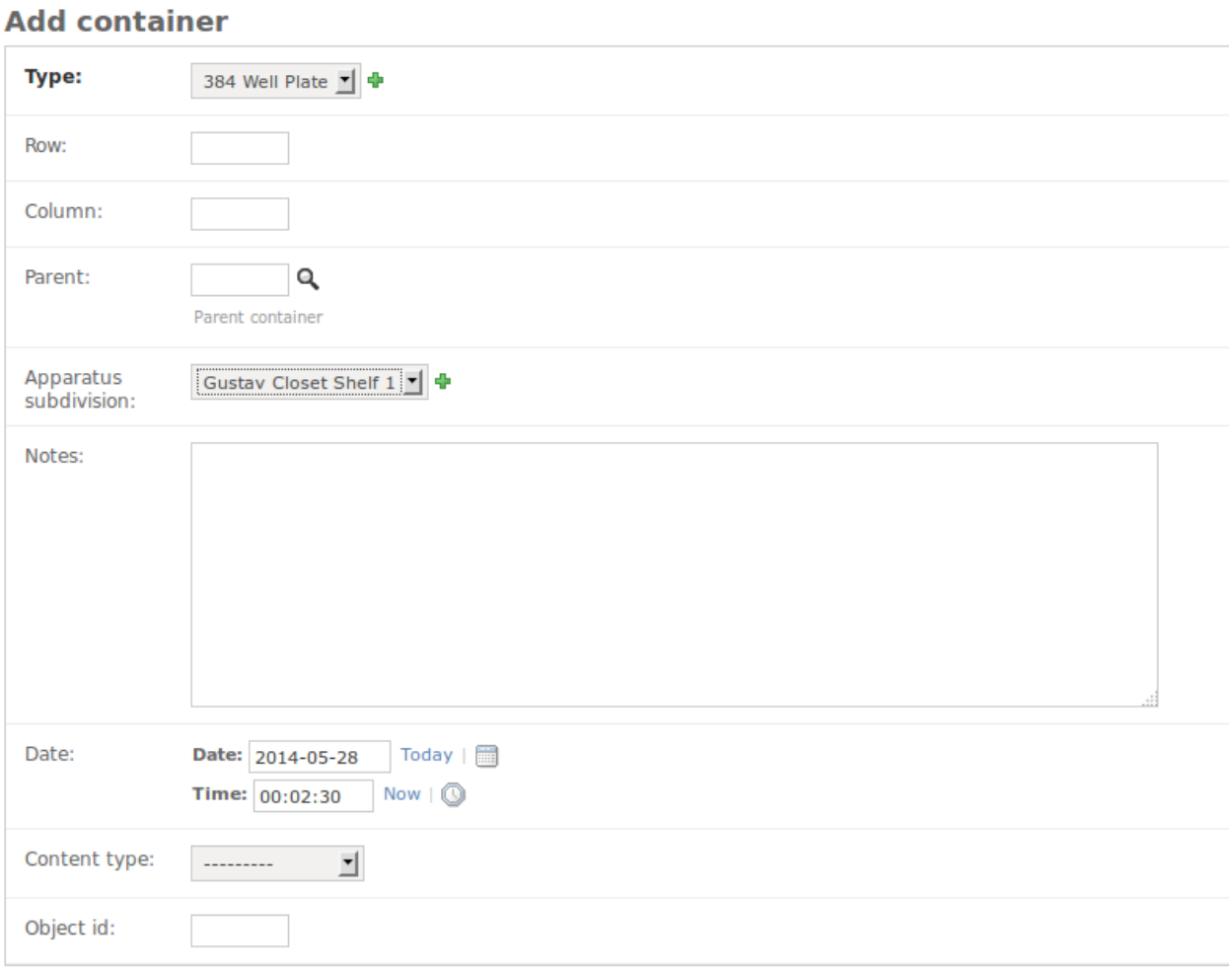
3. Import the edited csv file. Make sure you saved the csv without added quotation marks around the values. Don't mind the crossed out date part, this is a known bug in the importer.

Preview

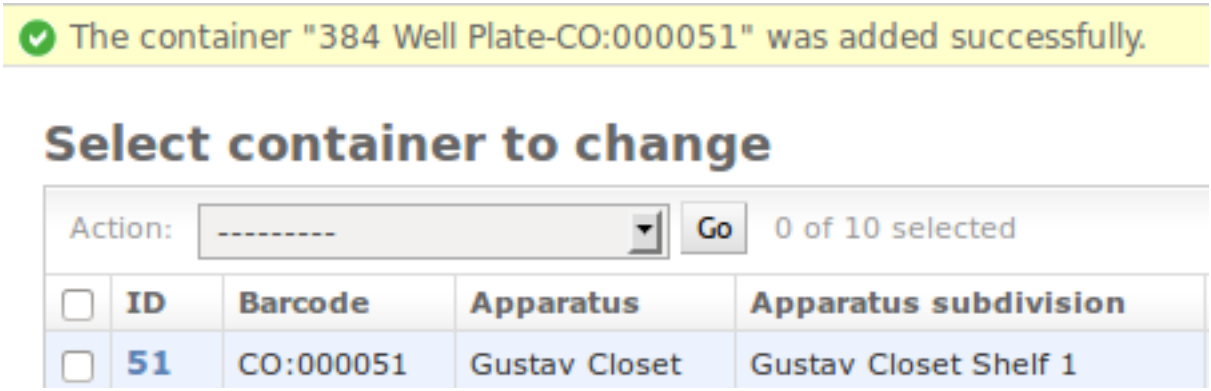
	id	uid	collaborator	sample_type	sample_location	temperature	ph	salinity	depth	latitude	longitude	shipping_method	date_received	date	biosafety_level	status	notes	extra_columns_json
New		C001P	9	5	4				1711				2014-05-27 23:53:332-01 00:00:00	20143-05-27 23:53:337-19 00:00:00				{"Characteristics": "Little ethanol amount at bottom of core", "Others": "", "Length of piston core (cm)": "191", "Sample name old": "OYEAH C1P"}
New		C002P	9	5	4				1635				2014-05-27 23:53:332-01 00:00:00	20143-057-27 23:53:330 00:00:00				{"Characteristics": "No ethanol at the bottom of core", "Others": "samples selected for metatranscriptomic sequencing", "Length of piston core (cm)": "110", "Sample name old": "OYEAH C2P"}
New		C003P	9	5	4				1399				2014-05-27 23:53:332-01 00:00:00	20143-057-27 23:53:330 00:00:00				{"Characteristics": "High ethanol amount at bottom of core", "Others": "", "Length of piston core (cm)": "135", "Sample name old": "OYEAH C3P"}
New		C004P	9	5	4				1334				2014-05-27 23:53:332-01 00:00:00	20143-057-27 23:53:330 00:00:00				{"Characteristics": "Little ethanol amount at bottom of core", "Others": "", "Length of piston core (cm)": "37", "Sample name old": "OYEAH C4P"}
New		C005P	9	5	4				1326				2014-05-27 23:53:332-01 00:00:00	20143-057-27 23:53:331 00:00:00				{"Characteristics": "No ethanol at the bottom of core", "Others": "", "Length of piston core (cm)": "449", "Sample name old": "OYEAH C5P"}

Bulk import a group of wells to store samples in one go

1. Create a 384 Well Plate to connect the Wells to using the admin interface



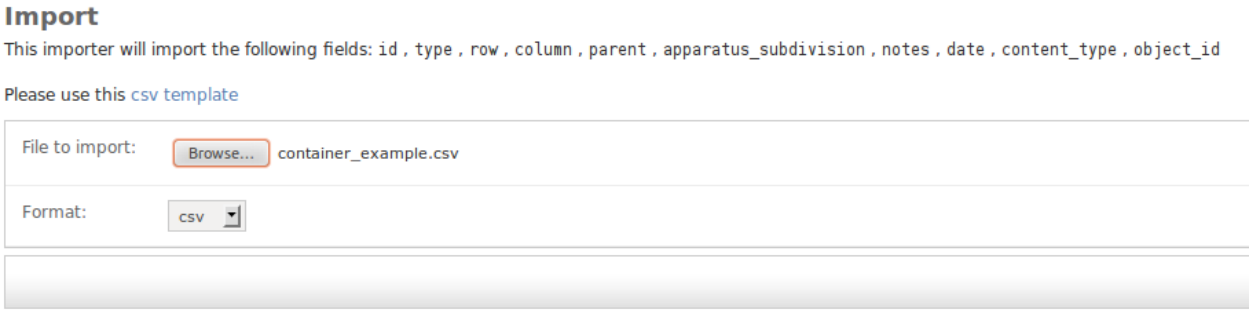
2. Get the id for the new 384 Well Plate (51)



3. Determine the id for the container type you want to add (4)

ID	Name
4	Well
3	Bag
2	384 Well Plate
1	Petri Dish

4. Click on import in the Container listing



5. Change the given csv template. It uses the id from step 2 as the parent container and the sample type from step 3. You don't have to specify an id for the new containers themselves because those will be automatically generated by the system. The `content_type` is an id that refers to the type of object that is stored, in this case 15 which represents the code for a `Sample`. This id might be different on your server though. You can look up the ids for other objects in the `ContentTypes` listing in the admin area. The `object_id` should contain the id of the object, in this example we used the ones from the previous step.

id	type	row	column	parent	apparatus_subdivision	notes	date	content_type	object_id
	4	1	1	51			2014-03-27 00:00:00	15	11
	4	1	2	51			2014-03-27 00:00:00	15	12
	4	1	3	51			2014-03-27 00:00:00	15	13
	4	1	4	51			2014-03-27 00:00:00	15	14
	4	2	1	51			2014-03-27 00:00:00	15	15
	4	2	2	51			2014-03-27 00:00:00	15	16
	4	2	3	51			2014-03-27 00:00:00	15	17
	4	2	4	51			2014-03-27 00:00:00	15	18
	4	3	1	51			2014-03-27 00:00:00	15	19
	4	3	2	51			2014-03-27 00:00:00	15	20
	4	3	3	51			2014-03-27 00:00:00	15	21
	4	3	4	51			2014-03-27 00:00:00	15	22
	4	4	1	51			2014-03-27 00:00:00	15	23
	4	4	2	51			2014-03-27 00:00:00	15	24
	4	4	3	51			2014-03-27 00:00:00	15	25
	4	4	4	51			2014-03-27 00:00:00	15	26

6. Import the csv file.

Preview

	id	type	row	column	parent	apparatus_subdivision	notes	date	content_type	object_id
New		4	1	1	51			2014-053-287 00:11:1700:00	15	11
New		4	1	2	51			2014-053-287 00:11:1700:00	15	12
New		4	1	3	51			2014-053-287 00:11:1700:00	15	13
New		4	1	4	51			2014-053-287 00:11:1700:00	15	14
New		4	2	1	51			2014-053-287 00:11:1700:00	15	15
New		4	2	2	51			2014-053-287 00:11:1700:00	15	16
New		4	2	3	51			2014-053-287 00:11:1700:00	15	17
New		4	2	4	51			2014-053-287 00:11:1700:00	15	18
New		4	3	1	51			2014-053-287 00:11:1700:00	15	19
New		4	3	2	51			2014-053-287 00:11:1700:00	15	20
New		4	3	3	51			2014-053-287 00:11:1700:00	15	21
New		4	3	4	51			2014-053-287 00:11:1700:00	15	22
New		4	4	1	51			2014-053-287 00:11:1700:00	15	23
New		4	4	2	51			2014-053-287 00:11:1700:00	15	24
New		4	4	3	51			2014-053-287 00:11:1700:00	15	25
New		4	4	4	51			2014-053-287 00:11:1700:00	15	26

4.3 Presentation

There's also a presentation of the system available that incorporates several movies of how to use the system.

4.4 Issues

Check out the *Issues* section if you run into errors.

CHAPTER 5

Installation

5.1 Requirements

- Python 2.7+
- Django 1.6.1
- Local: https://github.com/BILS/ETTLIMS/blob/master/lims_project/requirements/local.txt
- Development: https://github.com/BILS/ETTLIMS/blob/master/lims_project/requirements/development.txt

5.2 Installation

Clone the repository to your computer:

```
git clone https://github.com/BILS/ETTLIMS
```

Local installation:

```
pip install -r lims_project/requirements/local.txt
```

5.3 Running ETTLIMS

5.3.1 Locally with SQLite

Without example data:

```
cd lims_project
python manage.py syncdb --settings=lims_project.settings.local && \
python manage.py runserver 127.0.0.1:8000 --settings=lims_project.settings.local
```

With all example data:

```
cd lims_project
python manage.py syncdb --settings=lims_project.settings.local && \
python manage.py loaddata example barcode_example --settings=lims_project.settings.local && \
python manage.py runserver 127.0.0.1:8000 --settings=lims_project.settings.local
```

Remove `example` from the `loaddata` command if you only want the `barcode_example` data and vice versa.

Empty the database:

```
cd lims_project
rm lims_project/lims_project.sqlite
```

5.3.2 Development with PostgreSQL database

First you need to set up PostgreSQL. You'll have to create a database django and user django, see the development settings. Then the commands are similar to running it locally, but you change `lims_project.settings.local` to `lims_project.settings.development` e.g.:

```
cd lims_project
python manage.py syncdb --settings=lims_project.settings.development && \
python manage.py loaddata example barcode_example --settings=lims_project.settings.development && \
python manage.py runserver 127.0.0.1:8000 --settings=lims_project.settings.development
```

To empty the PostgreSQL database you can run:

```
python manage.py sqlclear sessions admin lims auth contenttypes \
    --settings=lims_project.settings.development | \
grep -v parent_id | \
python manage.py dbshell --settings=lims_project.settings.development
```

The `grep -v parent_id` is necessary to remove a non-existing constraint from the generated sql statements by django. This is a known django bug.

CHAPTER 6

Developer guide

6.1 Contribute

Contributions are greatly appreciated, please read the contribution instructions.

CHAPTER 7

Issues

7.1 Known issues

- Date part gets crossed out in bulk import. The importer thinks the date is being edited from the default value

7.2 Reporting issues

An up to date list of issues can be found at: https://github.com/bils/ettlims/issues. Check if your issue is there, if not add it.