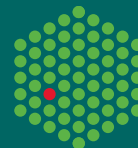


py-ispyb and ssx

Ivars Karpičs

EMBL

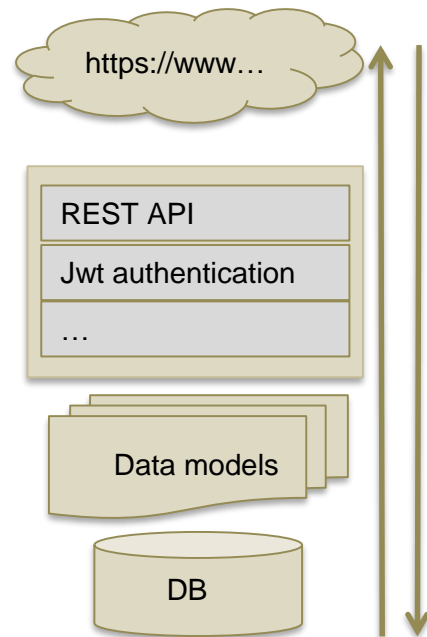


Content

- ISPyB backend server pyispyb.
- Latest developments.
- SSX data model and ssx microservice.

Overview

- **Python 3.x**
- **Flask**>=1.1,<2
- **flask-restx**
- **Flask-Cors**>=3.0.8,<4
- **SQLAlchemy**>=1.3.0,<2
- **Flask-SQLAlchemy**>=2.4,<3
- **marshmallow**>=2.13.5,<3
- **flask-marshmallow**>=0.7,<0.8
- **marshmallow-sqlalchemy**>=0.12,<0.13
- **marshmallow_jsonschema**
- **pyjwt**
- **ruamel.yaml**
- **pdfkit**
- **python-barcode**

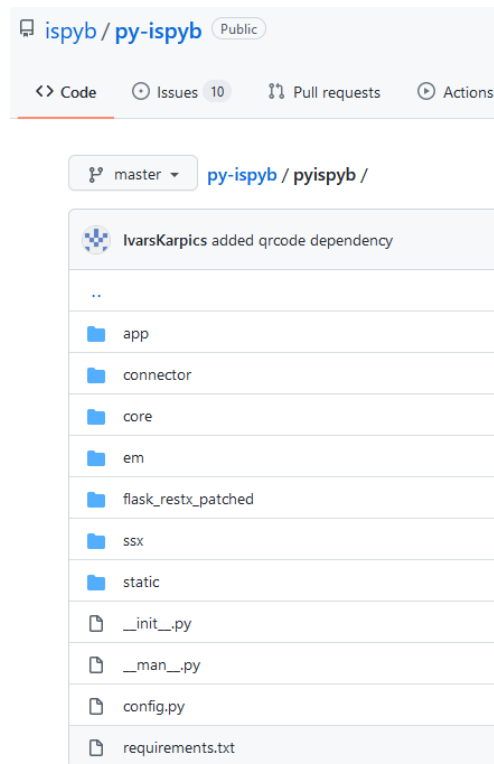


Structure

- Plug-in **extensions**: api, auth, db, logging.
- Microservice definition directories:
 - core
 - ssx
 - em

Each service contains:

- **models.py**: sqlalchemy classes reflecting the data base.
- **schemas**: flask, marshmallow and json schemas.
- **modules**: data handling modules.
- **routes**: API end point definitions.
- Scripts, tests, deploy, docker, setup.py



Versioning and deployment

1. Install pyispyb:

```
pip install pyispyb
```

2. Run scripts to generate data models and schemas:

```
scripts/generate_core_models.sh path_to_config_file.yml  
scripts/generate_core_schemas.py
```

3. Start the pyispyb server:

```
from pyispyb import create_app  
ispyb_app = create_app(config_filename, 'dev')  
ispyb_app.run(host='0.0.0.0', port=5000, debug=True)
```

```
gunicorn -b 127.0.0.0:4000 "app:create_app()"
```

```
from gevent.pywsgi import WSGIServer  
from pyispyb import create_app  
ispyb_app = create_app(config_filename, 'dev')  
ispyb_server = WSGIServer(('', 5000), ispyb_app)  
ispyb_server.serve_forever()
```

The screenshot shows the PyPyB 1.0.0 project page on PyPI. The header is blue with the text 'pyispyb 1.0.0' and a green 'Latest version' button. Below the header, there's a blue button for 'pip install pyispyb' and a release date of 'Released: Jan 15, 2021'. The main content area is white and divided into two columns. The left column contains 'Navigation' with links for 'Project description', 'Release history', and 'Download files'. Below that is 'Project links' with a link to the 'Homepage'. Further down is 'Statistics' showing 'Stars: 3', 'Forks: 2', and 'Open issues/PRs: 10'. The right column contains 'Project description' with the text 'py-ispyb' and a description 'ISPyB backend server based on python flask-restx'. Below that is 'Dependencies' listing 'Python 3.5+ / pypy2', 'flask-restx (+ flask)', 'sqlalchemy (+ flask-sqlalchemy) - Database ORM.', 'marshmallow', and 'ruamel.yaml'. At the bottom right is 'How to run py-ispyb' and 'Install requirements' with a code snippet: 'sudo apt-get install -y python3-mysqldb'.

pyispyb 1.0.0

pip install pyispyb

Released: Jan 15, 2021

ISPyB backend server

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Statistics

GitHub statistics:

- Stars: 3
- Forks: 2
- Open issues/PRs: 10

View statistics for this project via [Libraries.io](#) or by using [our public dataset on Google BigQuery](#)

Project description

py-ispyb

code quality B build passing codecov 85% License LGPL v3

ISPyB backend server based on python flask-restx.

Dependencies

- Python 3.5+ / pypy2
- flask-restx (+ flask)
- sqlalchemy (+ flask-sqlalchemy) - Database ORM.
- marshmallow
- ruamel.yaml

How to run py-ispyb

Install requirements

In case of MySQL or MariaDB you might have to install dev tools:

```
sudo apt-get install -y python3-mysqldb
```

Documentation

- Swagger ui to document and test api.

paths:

- /auth/login: [-]
- /autoproc: [-]
- /autoproc/programs: [-]
- /autoproc/programs/attachments: [-]
- /autoproc/programs/attachments/{attachment_id}: [-]
- /autoproc/programs/{program_id}: [-]
- /autoproc/status: [-]
- /autoproc/status/{status_id}: [-]
- /autoproc/auto_proc_id: [-]
- /contacts/lab_contacts: [-]
- /contacts/lab_contacts/{lab_contact_id}: [-]
- /contacts/labs: [-]
- /contacts/labs/{lab_id}: [-]
- /contacts/persons: [-]
- /contacts/persons/{person_id}: [-]
- /containers: [-]
- /containers/{container_id}: [-]
- /data_collections: [-]
- /data_collections/groups: [-]
- /data_collections/{data_collection_id}: [-]
- /dewars: [-]
- /dewars/{dewar_id}: [-]
- /proposals: [-]
- /proposals/{proposal_id}: [-]
- /proposals/{proposal_id}/info: [-]
- /samples: [-]
- /samples/crystals: [-]
- /samples/crystals/{crystal_id}: [-]
- /samples/proteins: [-]
- /samples/{sample_id}: [-]
- /schemas/available_names: [-]
- /schemas/{name}: [-]
- /sessions: [-]
- /sessions/date: [-]
- /sessions/{session_id}: [-]
- /sessions/{session_id}/info: [-]
- /shipments: [-]
- /shipments/{shipment_id}: [-]
- /shipments/{shipment_id}/info: [-]
- /user_office/sync_all: [-]
- /user_office/update_proposal/{proposal_code}/{proposal_number}: [-]

POST /proposals Adds a new proposal

Parameters

Name	Description
payload	Example Value Model

object (body)

```
{  "proposalId": 0,  "personId": 0,  "title": "string",  "proposalCode": "string",  "proposalNumber": "string",  "blTimeStamp": "2020-11-12T08:49:53.202Z",  "proposalType": "string",  "externalId": 0,  "state": "string"}
```

Parameter content type

application/json

X-Fields

string(\$mask) (header)

An optional fields mask

X-Fields - An optional fields mask

Responses

Code	Description
201	Success

Example Value | Model

```
{  "proposalId": 0,  "personId": 0,  "title": "string",  "proposalCode": "string",  "proposalNumber": "string",  "blTimeStamp": "2020-11-12T08:49:53.231Z",  "proposalType": "string",  "externalId": 0,  "state": "string"}
```

Models

Person >

LabContact >

Laboratory >

Session >

Shipping >

Proposal {

- proposalId integer
- personId integer
- title string
- proposalCode string
- proposalNumber string
- blTimeStamp string(\$date-time)
- proposalType string

Proposal type: MX, BX

- externalId integer
- state string

enum(Open,Closed,Cancelled)

Curl

```
curl -X GET "http://localhost:5000/ispyb/api/v1/schemas/proposal" -H "accept: application/json"
```

Request URL

http://localhost:5000/ispyb/api/v1/schemas/proposal

Server response

Code Details

200

Response body

```
{  "$schema": "http://json-schema.org/draft-07/schema#",  "definitions": {    "ProposalSchema": {      "type": "object",      "properties": {        "blTimeStamp": {          "title": "blTimeStamp",          "type": "string",          "format": "date-time"        },        "externalId": {          "title": "externalId",          "type": "number",          "format": "integer"        },        "personId": {          "title": "personId",          "type": "number",          "format": "integer"        },        "proposalCode": {          "title": "proposalCode",          "type": "string"        },        "proposalId": {          "title": "proposalId",          "type": "integer"        }      }    }  }
```

File download

- Download autoproc program attachment files (single file by id and all files as a directory zip file).
- Download data collection snapshots and other associated files (image quality plot, Wilson plot).

GET /data_collections/{data_collection_id}/file Downloads data collection attribute by id and attribute_name



GET /data_collections/{data_collection_id}/snapshot/{snapshot_index} Downloads data collection attribute by id and attribute_name



GET /autoproc/programs/attachments/{attachment_id} Returns a auto_proc by attachment_id

OPTIONS /autoproc/programs/attachments/{attachment_id} Check which methods are allowed

GET /autoproc/programs/attachments/{attachment_id}/download Downloads autoproc program attachment file by attachment_id

OPTIONS /autoproc/programs/attachments/{attachment_id}/download Check which methods are allowed

GET /autoproc/programs/{program_id} Returns a auto_proc by auto_procid

OPTIONS /autoproc/programs/{program_id} Check which methods are allowed

GET /autoproc/programs/{program_id}/attachments Returns list of autoproc program attachments

OPTIONS /autoproc/programs/{program_id}/attachments Check which methods are allowed

GET /autoproc/programs/{program_id}/attachments/download Downloads zip file with auto proc attachment files

OPTIONS /autoproc/programs/{program_id}/attachments/download Check which methods are allowed

Pdb files

- Upload/download pdb file. Link with <https://www.rcsb.org/>

```
GET /samples/crystals/{crystal_id}/pdb Returns pdb file by crystalId
```

- Downloads *.pdb if the file is on the filesystem. Preferred solution for manually edited pdb files.
- If pdb file is not available then download it from rcsb.org.

Dewar labels

- Dewar pdf labels are generated based on a html template.
- No pdf file templates, data fields etc. Avoids complicated pdf file handling.
- Template is filled based on pyispyb configuration and shipping info.
- Html file is converted to pdf with pdfkit.
- python-barcode and qrcode are used to generate barcode and qrcode.
- New directory static to store html template, images and other resources.

GET /shippings/dewars/{dewar_id}/labels Returns a dewar labels by dewarid

Curl

```
curl -X GET "http://0.0.0.0:5000/ispyb/api/v1/shippings/dewars/306656/labels" -H "accept: application/json" -H "Authorization: Bearer MasterToken"
```

Request URL

http://0.0.0.0:5000/ispyb/api/v1/shippings/dewars/306656/labels

ISPyB Dewar Tracking

LABEL INSTRUCTIONS

Please print the following three labels and use as follows:

- 1) **Dewar Label:** affix this label to your dewar which ensures it can be identified at all times at the Generix.
- 2) **Outward bound Address label:** To be attached to the outside of your transport container for shipment to the Generix.
- 3) **Return bound address label:** The return address for your shipment (Please include this in your shipment, e.g. put it behind the outward bound address or in the transport container)



1) DEWAR LABEL: affix this label to your DEWAR



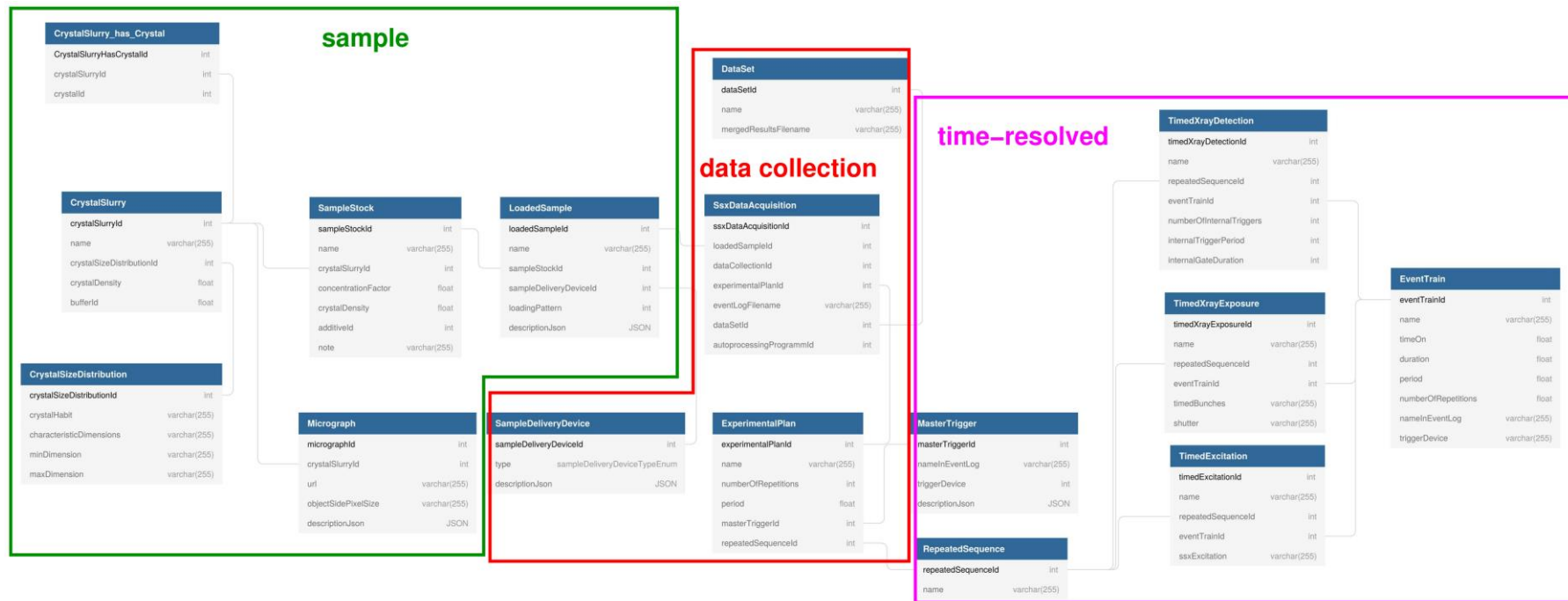
EMBL3066568



Parcel label dew_01

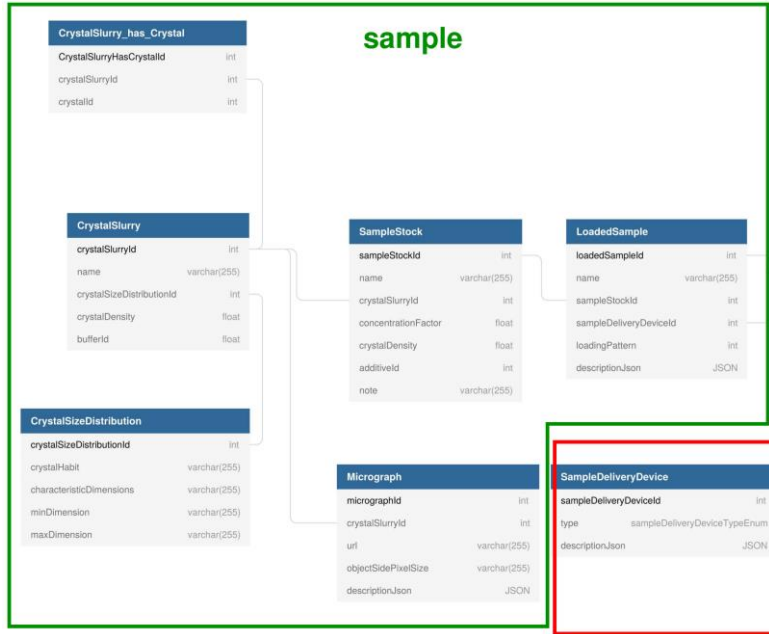
Shipment name	test_shipp
Number of parcels	1
Proposal number	mx1
Laboratory name	EMBL Hamburg
Local contact	5

SSX data model



pyisyb and ssx

- Two new hardware objects for mxcubeq to access pyisyb REST API:
 - PylspybClient.py
 - SsxlsybClient.py
- Prototype widget for mxcubeq to acquire and store information about ssx samples.



File Queue Tools View Graphics Help

Collect SSX

Loaded sample

Refresh Lys

Stock: test

Delivery device: tapeDevice

Loading pattern 10

Description test

Crystal slurry Test crystall slurry

Work in progress

Thank you for your attention!