# ISPyB Developpers Web meeting

2020-05-11

## Participants

**Diamond**: Karl Levik, Neil Smith; **EMBL Hamburg**: Ivars Karpics; **ESRF**: Solange Delageniere, Alex de Maria, Olof Svensson; **Global Phasing**: Rasmus Fogh; **MaxIV**: Alberto Nardella; **SOLEIL**: Tatiana Isabet, Majid Ounsy, Idrissou Chado. **ALBA**: Daniel Sanchez

## Agenda:

1. Issue raised by Gianluca Santoni, on ESRF priorities

2. New ISPyB prototype by Ivars Karpics

# Meeting:

## 1. ESRF priorities

As Gianluca Santoni could not attend this meeting, this point was postponed to the next meeting

## 2. New IspyB Prototype

### Report by IK on the new prototype

This was presented by IK by email before the meeting (reformatted by RF):

```
Before the next developers meeting I wanted to share a short status report
about the ispyb backend server
(https://github.com/IvarsKarpics/ispyb_backend_prototype) that I am
currently working on. […]

  1. As previously discussed the backed server is based on python and
     current packages:
     1. Flask and flask-restx as web framework.
     2. SQLAlchemy and Flask-sqlalchemy to provide access to database.
     3. Marshmallows for argument parsing and data serializations.

  2. Server is configurable via
     config.py: https://github.com/IvarsKarpics/ispyb_backend_prototype/b
     lob/master/config.py allowing to switch between development, testing
     and production mode.

  3. Server is modular and allows to enable/disable extensions
     (authorization, logging, etc) and modules. One can easily
     enable/disable db modules. Maybe there could be some core db modules
     like proposals, person, samples and auxiliary modules like phasing,
```

```
      autoprocessing and etc.

   4. Each module (app/api/modules) has a structure:
      1. schemas.py : defines marshmallows schema and rest api models
      2. resources.py: defines method and routes
      3. see proposals example:
         https://github.com/IvarsKarpics/ispyb_backend_prototype/tree/maste
         r/app/modules/proposal

   5. In the case of changes in the data base a script
      https://github.com/IvarsKarpics/ispyb_backend_prototype/blob/master/s
      cripts/generate_db_models.sh generated SQLAlchemy models and another
      script
      https://github.com/IvarsKarpics/ispyb_backend_prototype/blob/master/s
      cripts/generate_modules.py generated schemas.py and resources.py (if
      it does not exist)

   6. For authentification json web tokens are used. Expiration and coding
      algorithm is defined in config. One can define a site specific python
      module that will authenticate the user.

   7. For continued integration TravisCI runs pylint -E and coverage pytest
      and sends coverage results to codecov.

   8. Code is formatted with autopep and black.
```

## Discussion

The sense of the meeting (as confirmed by the final round of reactions) was that this prototype was accepted, a good start for future work – and very much appreciated. It was agreed that IK should continue with this development. There was some discussion on individual points.

**2.**
Could configuration be moved from Python code to files, e.g. JSON files (DS)?  IK could see no objection to this. It was proposed that sites might prefer to not commit their configuration files. One proposal was to specify what amounted to an interface, another to specify Python classes for calling the configuration, (to ensure uniformity) but leave the actual configuration parameters to be provided locally. It was agreed to continue this discussion in a Github issue.

**4.**
Does the current organisation have one module per table, and is this the way forward (KL)? IK confirmed that this was the case for now, but need not remain so. It was agreed that the current organisation was a good start for experimentation, but that in the future modules would be organised to cover multiple tables.

It was confirmed that it is possible to enable and disable modules individually, but that the distributed code would contain the full set of modules with only the endpoints disabled.

**5.**
Marshmallow schemas currently appear to be handwritten – could they also be generated automatically (NS)? According to IK  the marshmallow schemas are already generated, just by a different (shell) script. It would be possible to couple or combine the scripts so that both  sets of schemas could be written in tandem.

**6.**

AdM noted that there had been numerous requests from small(er) labs to start using ISPyB, and they were likely to use numerous different authentification systems. Will this prototype be compatible with different systems? IK is himself using LDAP authentication directly, but proposed that people could override the authentication code using local files that were not committed to the central repository. AdM suggested using a RESTful interface and committing only the interface

**Use cases and tasks:**
- Gleb Bourenkov had been looking at use cases but interrupted the work since the necessary wider involvement had fallen victim to the COVID-19 precautions. Issues that would be important to bring in (NS) were queries that spanned multiple tables, and pagination , i.e. allowing large query responses to be truncated or returned in multiple chunks.

- Authentication was mentioned as an important module to get working fast.

- AdM suggested that a sample tracking module would be a good start. Given the changes in working procedures (mail-in and remote working etc.) triggered by the COVID-19 pandemic, improvements to sample tracking were urgently needed. IK worried that fully solving that problem would require preparing all modules up till experimental session, amounting to maybe half of the total code. AdM, however, envisaged that it would be possible to use a new sample tracking module in combination with the existing modules of the current ISPyB.

- IK requested a database and an example data set, so he could be sure he was testing against the right target.

**Miscellaneous:**
- It would be necessary with a blacklist of names that clashed with Python reserved words, e.g. 'global'.

- IK asked which version of the ISPyB database should be used as a starting point. The answer (from NS) was to use the official and agreed version; Diamond would take care of dealing with any discrepancies in the version they use.

- It was agreed that IK should move the prototype from his personal repository on Github to the main ISPyB repository.

- The work should be issued under the LGPL license

- The name of the new project was discussed: ISPyB2? PyISPyB?

## Next Meeting:

Web meeting would be on Monday June 8th 2020, 15h French time
https://whereby.com/ispyb