

# ISPyB Web Services

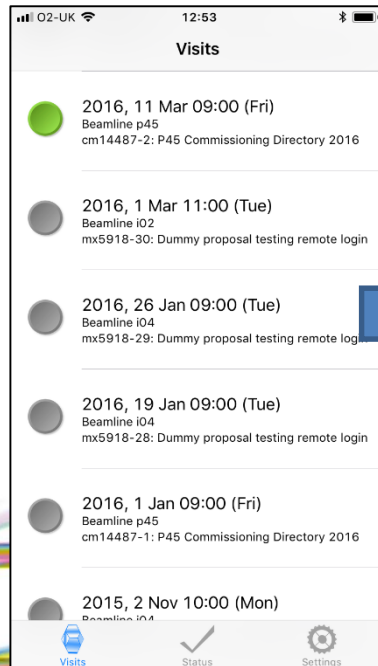
Adapting for SynchLink Mobile App

# Contents

- Introduction to the task and SynchLink
- Documentation via Swagger
- Changes required:
  - CAS authentication
  - Endpoints
- Summary

# SynchLink

## Sessions



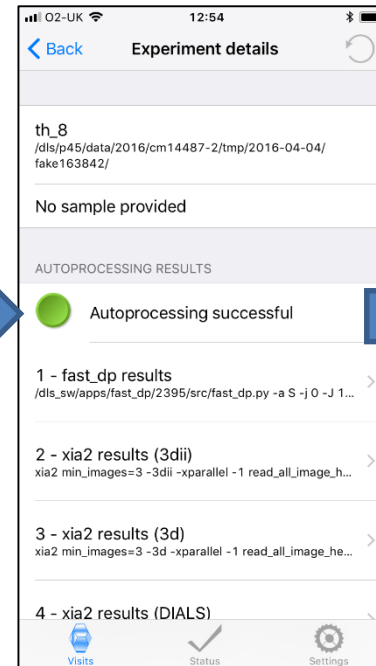
Visits	
	2016, 11 Mar 09:00 (Fri) Beamline p45 cm14487-2: P45 Commissioning Directory 2016
	2016, 1 Mar 11:00 (Tue) Beamline i02 mx5918-30: Dummy proposal testing remote login
	2016, 26 Jan 09:00 (Tue) Beamline i04 mx5918-29: Dummy proposal testing remote login
	2016, 19 Jan 09:00 (Tue) Beamline i04 mx5918-28: Dummy proposal testing remote login
	2016, 1 Jan 09:00 (Fri) Beamline p45 cm14487-1: P45 Commissioning Directory 2016
	2015, 2 Nov 10:00 (Mon) Beamline i04

## Data collections



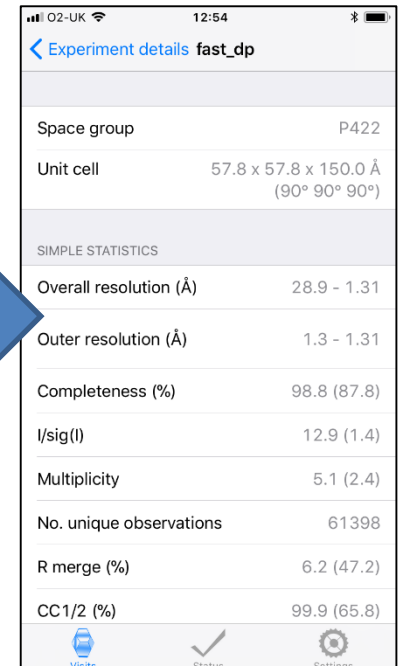
Experiments (p45)	
	th_8 1.51 Å tmp/2016-04-04/fake163842/ PM Rmerge = 0.0 I/Sigma = 10.4
	th_8 1.51 Å tmp/2016-03-29/fake170033/ PM Rmerge = 0.0 I/Sigma = 10.4
	th_8 1.51 Å tmp/2016-03-29/fake144228/ PM Rmerge = 0.0 I/Sigma = 10.4
	th_8 1.51 Å tmp/2016-03-29/fake144213/ PM Rmerge = 0.0 I/Sigma = 10.4
	th_8 1.51 Å tmp/2016-03-29/fake144200/ PM Rmerge = 0.0 I/Sigma = 10.4

## Data collection results



Experiment details	
th_8	/dls/p45/data/2016/cm14487-2/tmp/2016-04-04/fake163842/
No sample provided	
AUTOPROCESSING RESULTS	
	Autoprocessing successful
1 - fast_dp results	/dls_sw/apps/fast_dp/2395/src/fast_dp.py -a S -j 0 -J 1...
2 - xia2 results (3dii)	xia2_min_images=3 -3dii -xparallel -1 read_all_image_h...
3 - xia2 results (3d)	xia2_min_images=3 -3d -xparallel -1 read_all_image_h...
4 - xia2 results (DIALS)	

## Auto processing results



Experiment details fast_dp	
Space group	P422
Unit cell	57.8 x 57.8 x 150.0 Å (90° 90° 90°)
SIMPLE STATISTICS	
Overall resolution (Å)	28.9 - 1.31
Outer resolution (Å)	1.3 - 1.31
Completeness (%)	98.8 (87.8)
I/sig(I)	12.9 (1.4)
Multiplicity	5.1 (2.4)
No. unique observations	61398
R merge (%)	6.2 (47.2)
CC1/2 (%)	99.9 (65.8)

A screenshot of a mobile application interface. The top half of the screen is black and displays the text "Omega = 270.0°" in white. Below this, there is a grey bar containing three icons: a blue car icon on the left, a black checkmark icon in the center, and a black circular icon on the right. The background of the entire slide features abstract, colorful, flowing lines in shades of blue, yellow, and pink.

## Status

Storage ring status (week) >

Storage ring status (day) >

ID and beamline status >

### OPERATIONS CALENDAR

1st day of the week      2nd day of the week      3rd day of the week      4th day of the week

■ No collection, beamline closed  
■ No collection, beamline open  
■ No collection, beamline closed  
■ No collection, beamline open

■ No collection, beamline closed  
■ No collection, beamline open  
■ No collection, beamline closed  
■ No collection, beamline open

Monday 2019-01-07
Monday 2019-01-14
Monday 2019-01-21
Monday 2019-01-28

Visita

Status

Settings

# SynchLink

## iPad

The iPad app interface is shown in two parts. The left part displays a list of experiments under the 'Experiments (i03)' header. Each entry includes a clock icon, a red progress bar, and a 'PM' label. The right part shows the 'Experiment details' screen for a selected experiment, displaying various parameters and a list of associated images.

Experiment Name	Resolution (Å)	Rmerge	I/Sigma
therm3_D11d1	2.00	11.1	8.0
therm3_D11d1	2.00	10.9	8.2
therm2_D12d1	2.00	16.4	6.6
linediffraction_1	2.00	-	-
therm2_D12d1	2.00	17.4	5.7
therm2_D12d1	2.00	17.1	6.0

Experiment details	
COMMENTS	
(-1766,-40425,-41) Aperture: Medium	
ASSOCIATED IMAGES	
No snapshots	
Diffraction images	
EXPERIMENT PARAMETERS	
Images collected	87
Wavelength	0.976 Å
Omega start	-5.0°
Omega end	38.5°
Rotation per image	0.50°
Exposure time	0.200 s
Beamsize X	30.0 µm
Beamsize Y	30.0 µm
Transmission	10.0%
Detector resolution	2.0 Å

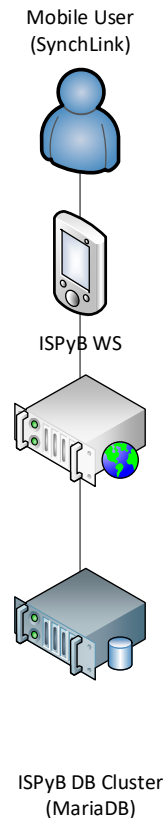
## iPhone

The iPhone app interface shows the 'Experiment details' screen. It displays the experiment name 'th\_8', a file path, and a status 'No sample provided'. Below this, the 'AUTOPROCESSING RESULTS' section shows a green circle icon and the text 'Autoproducting successful'. A list of results follows, including '1 - fast\_dp results', '2 - xia2 results (3dii)', '3 - xia2 results (3d)', and '4 - xia2 results (DIALS)'. Each result has a right arrow indicating further details.

Experiment details	
th_8	
/dls/p45/data/2016/cm14487-2/tmp/2016-04-04/fake163842/	
No sample provided	
AUTOPROCESSING RESULTS	
Autoproducting successful	
1 - fast_dp results	
/dls_sw/apps/fast_dp/2395/src/fast_dp.py -a S -j 0 -J 1...	
2 - xia2 results (3dii)	
xia2 min_images=3 -3dii -xparallel -1 read_all_image_he...	
3 - xia2 results (3d)	
xia2 min_images=3 -3d -xparallel -1 read_all_image_he...	
4 - xia2 results (DIALS)	

# Introduction

- SynchLink was integrated with “GenericWebServices”
- Java software stack using apache tomcat
- Only used by SynchLink
- SQL query built from XML message payload
- Supports file transfer for crystal snapshots etc.
- Task:
  - to migrate to use ISPyB Java web services
  - Gain experience adding/modifying endpoints

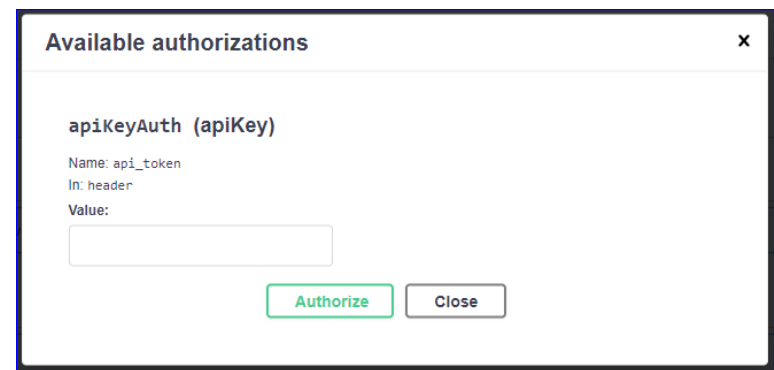
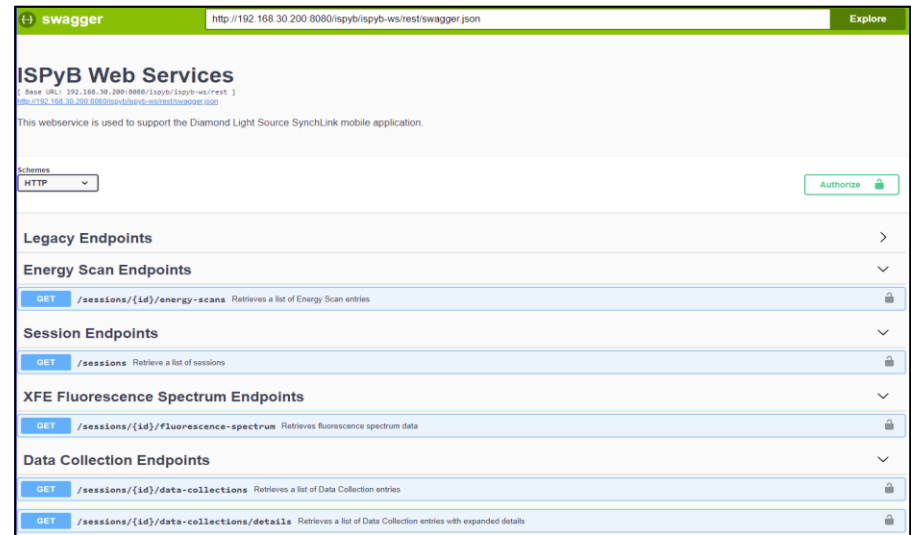


# Challenges

- Documentation
  - Use of web service calls contained in abstracted classes, no clear mapping between page and call
- Documentation
  - Generic Web services means decoding actual SQL query used non-trivial
- Documentation
  - API documentation for endpoints in Java ISPyB web services disabled/commented out

# Swagger => OpenAPI

- A specification for documenting REST APIs
- Inline documentation through code annotations including input parameters and response codes
- Configured to either generate a spec file or self host a UI to test queries
- The existing “authorize” endpoint generates a token that can be input when clicking the “Authorize” button shown. All padlocked endpoints will then be usable if the credentials input are valid.
- Configured in the “RestApplication.java” class in the ISPyB codebase (were commented out?)
- Requires the “io.swagger > swagger.jaxrs” maven dependency within the projects pom.xml file in order for the annotations to be used within the codebase.





# Swagger UI

- When an endpoint in the list is clicked, it is shown with its description and any responses it can return (HTTP error codes etc).
- Selecting “Try it out” will display a form with parameters required before a user can “Execute” the call.
- The response from the server is then shown beneath the Execute button with it’s HTTP code and the response value along with the URL that was called.

The screenshot shows the Swagger UI for a POST endpoint named `/authenticate`. The description states: "Allows a user to input a login name and password (along with the site they are logging in from). If the user credentials are validated, then an authentication token is returned which can be used as a header parameter for further usage of the service."

**Parameters**

Name	Description
login string (formData)	<input type="text" value="login"/>
password string (formData)	<input type="text" value="password"/>
site string (query)	<input type="text" value="site"/>

**Execute**

**Responses**

Response content type: `application/json`

Code	Description
default	<code>successful operation</code>

# Swagger Annotations

The @GET, @Path, @Produces and @PathParam annotations come from Jax-RS.

The tags value of the @ApiOperation allows you to group the API endpoints.

The response value will render an example of the object and its parameters at the bottom of the Swagger UI page

*Inline annotations can become rather large and unwieldy.*

```
@GET
@Path("/proposals/{id}")
@ApiOperation
(
    value = "Retrieve the information of a proposal",
    notes = "Obtain the information relating to a specific proposal (based on the input ID). "+
        "if it is available to the user currently logged in.",
    tags = { SwaggerTagConstants.PROPOSAL_TAG }, response = ProposalResponseDTO.class,
    authorizations = @Authorization( "apiKeyAuth" )
)
@Produces({ "application/json" })
@ApiResponses
(
    {
        @ApiResponse( code = 200, message = "Ok" ),
        @ApiResponse( code = 404, message = "A proposal could not be found for the input ID" )
    }
)
public Response retrieveProposalById
(
    @ApiParam
    (
        name = "id", required = true, example = "131", value = "The ID of the proposal to retrieve"
    ) @PathParam("id") int proposalId
) throws Exception
{
    String methodName = "retrieveProposals";
```

# Authentication

Created a new DLSLoginModule class that connects to a Central Authentication Service (CAS)

Calling “/authenticate” with the “site” parameter set to DLS, triggers the DLS case statement.

Assuming the user is authenticated, the AuthenticationRestWebService method generates a token that is stored in the database, within the Login table.

Appears that the token is random and does not contain any information encoded (e.g. user name). The Login table provides the association to the user.

# Endpoints

When an API endpoint is called the following interactions occur:

- The API token in the header is checked first to determine whether the user is authorised to use the endpoint.
- The relevant data is then retrieved (if possible) using the existing ISPyB service and VO classes that were already in place. We have tried to avoid modifying any code interacting with the database as much as possible.
- The VO classes contain a lot more information than required for the purposes of the Diamond endpoints, so we have created specialised DTO (data transfer objects) classes.
- The DTO classes act as the specialised representation of the data required for our purposes
  - When the data is retrieved from the database, it is passed into a conversion method, which simply takes the data required from the VO instance and places it into a DTO instance using each classes getter/setter methods.
- The data is then placed into a Response object and returned via the method with a relevant HTTP code.

# Endpoints...

A list of the endpoints implemented for the iOS app are shown below:  
These map onto the existing requirements of the application

Swagger grouping	URL
Authentication*	/authenticate
Energy Scan	/sessions/{id}/energy-scans
Session	/sessions
XFE Fluorescence Spectrum	/sessions/{id}/fluorescence-spectrum
Data Collection	/sessions/{id}/data-collections
Data Collection	/sessions/{id}/data-collections/details
Screening	/data-collections/{dcId}/screening-output-lattice/{solId}
Screening	/data-collections/{dcId}/screening-strategy-wedge/{sswId}
Screening	/data-collections/{dcId}/screening-strategy/{solId}
Screening	/data-collections/{dcId}/screening-comments
Proposal	/proposals/{id}

Swagger grouping	URL
Auto Processing	/auto-proc-integrations/{id}
Auto Processing	/auto-proc-scalings/{id}/mx-mr-runs
Auto Processing	/auto-proc-scalings/{id}/statistics
Auto Processing	/auto-proc/{id}
Auto Processing	/data-collections/{id}/auto-processing-results
Beam Line Sample	/beamline-samples/{id}
Crystal Snapshot	/data-collections/{dcId}/crystal-snapshot-paths
Crystal Snapshot	/data-collections/{dcId}/diffraction-images

# Summary

- We have designed new endpoints within ISPyB to support SynchLink
- Testing with an updated SynchLink is still needed
- Should we merge them into existing ISPyB classes?
- Should we create new package to contain DLS web service classes?
- Are there any plans to update the web services during the shutdown?
  - For example remove .../list, .../get from URLs?
  - Guidance on use of SQL resource files vs SQL in code?

# Questions?

# Example

`/token/proposal/proposal/session/sessionId/  
sessionId/list  
/sessions`