

11 - Integration with i2b2 software

To make sure that your new SHRINE instance can communicate with the i2b2 instance, there are a couple of places that we will address. First, you will need to make sure the PM, ONT, CRC cell urls are correctly defined within the **shrine.conf** file. These urls are what SHRINE uses to communicate with the i2b2 backend:

- In the main shrine {} section:

```
pmEndpoint {
  url = "http://localhost:9090/i2b2/services/PMService/getServices"
}

ontEndpoint {
  url = "http://localhost:9090/i2b2/services/OntologyService"
}
```

- In the adapter {} section:

```
crcEndpoint {
  url = "http://localhost:9090/i2b2/services/QueryToolService"
}
```

Next, you will need to add the SHRINE project into the pm_project_data table within the i2b2pm schema:

```
$ psql -U postgres i2b2

i2b2=# set search_path = i2b2pm;

i2b2=# insert into PM_PROJECT_DATA (project_id, project_name, project_wiki, project_path, status_cd) values
('SHRINE', 'SHRINE', 'http://open.med.harvard.edu/display/SHRINE', '/SHRINE', 'A');
```

After loading the ontology into SHRINE, you will need to add a SHRINE CRC connection to the pm_cell_data table within the i2b2pm schema:

```
$ psql -U postgres i2b2

i2b2=# set search_path = i2b2pm;

i2b2=# insert into PM_CELL_DATA (cell_id, project_path, name, method_cd, url, can_override, status_cd) values
('CRC', '/SHRINE', 'SHRINE Federated Query', 'REST', 'https://your_shrine_url:6443/shrine/rest/i2b2/', 1, 'A');
```

This should result in something that looks like this:

cell_id	project_path	name	method_cd	url
CRC	/	Data Repository	REST	http://localhost:9090/i2b2/services/QueryToolService/
CRC	/SHRINE	SHRINE Federated Query	REST	https://your_shrine_url:6443/shrine/rest/i2b2/
FRC	/	File Repository	SOAP	http://localhost:9090/i2b2/services/FRService/
IM	/	IM Cell	REST	http://localhost:9090/i2b2/services/IMService/
ONT	/	Ontology Cell	REST	http://localhost:9090/i2b2/services/OntologyService/
WORK	/	Workplace Cell	REST	http://localhost:9090/i2b2/services/WorkplaceService/

After doing that, you will also need to add the Shrine Ontology data source (DS) to the ont-ds.xml file within the i2b2 instance. This file should be located within your i2b2 deployments directory (**/opt/wildfly/standalone/deployments**):

```

<datasource jta="false" jndi-name="java:/ShrineOntologyDS" pool-name="ShrineOntologyDS" enabled="true" use-ccm="
false">
  <connection-url>jdbc:postgresql://localhost:5432/i2b2</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
  <driver>postgresql-9.2-1002.jdbc4.jar</driver>
  <security>
    <user-name>shrine_ont</user-name>
    <password>demouser</password>
  </security>

  <validation>
    <validate-on-match>false</validate-on-match>
    <background-validation>false</background-validation>
  </validation>

  <statement>
    <share-prepared-statements>false</share-prepared-statements>
  </statement>
</datasource>

```

Next, you will need to template the **i2b2_config_data.js** file to reflect the correct PM cell url for logging into the SHRINE webclient. The file is located in **/opt/shrine/tomcat/webapps/shrine-webclient/i2b2_config_data.js**. It should look like this:

```

{
  urlProxy: "/shrine-proxy/request",
  urlFramework: "js-i2b2/",
  loginTimeout: 15, // in seconds
  username_label: "SHRINE Username:", //Username Label
  password_label: "SHRINE Password:", //Password Label
  clientHelpUrl: 'help/pdf/shrine-client-guide.pdf',
  networkHelpUrl: 'help/pdf/shrine-network-guide.pdf',
  wikiBaseUrl: 'https://open.med.harvard.edu/wiki/display/SHRINE/',
  obfuscation: 10,
  resultName: "patients",

  // -----
  // THESE ARE ALL THE DOMAINS A USER CAN LOGIN TO
  lstDomains: [
    { domain: "i2b2demo",
      name: "SHRINE",
      urlCellPM: "http://localhost:9090/i2b2/services/PMSERVICE/",
      allowAnalysis: false,
      debug: true,
      isSHRINE: true
    }
  ]
  // -----
}

```

Once all of that is done, you will need to restart WildFly and Tomcat.