SHRINE 4.0.0 Chapter 8.2 - Configuring a Hub

Hub Admins Only

This section is intended for hub administrators only.

Here is a sample shrine.conf file for a system running SHRINE 4.0.0 , for a node supporting researchers and distributing queries.

shrine.conf

```
shrine {

  shrineHubBaseUrl = "https://localhost:6443" //The shrine hub's URL as observed from this tomcat server
  i2b2BaseUrl = "http://i2b2.example.com:9090" //The local i2b2's URL as observed from this tomcat server
  i2b2Domain = "exampleDomain"
  i2b2ShrineProjectName = "SHRINE"

  nodeKey = "somethingHub" //node key to get information from the hub about itself as a node.

  //shrineDatabaseType = "mysql" // "mysql" by default. It can be "sqlserver" "mysql" or "oracle"

  webclient {
    siteAdminEmail = "shrine-admin@example.com"
  }

  hiveCredentials {
    username = "demo"
    crcProjectId = "Demo"
  }//hiveCredentials

  hub {
    create = true

    messagequeue {
      blockingqWebApi {
        enabled = true  //run shrine's MoM system at the hub.
      }
    }//messagequeue
  }//hub

  adapter {
    create = false
  }//adapter

  keystore {
    privateKeyAlias = "shrine-hub"
    caCertAliases = ["shrine-ca"]
  }//keystore

  steward {
    emailDataSteward {
      //provide the email address of the shrine node system admin, to handle bounces and invalid addresses
      from = "shrine-admin@example.com"
      //provide the email address of the shrine node system admin, to handle bounces and invalid addresses
      to = "shrine-steward@example.com"
      //provide the externally-reachable URL for the data steward
      externalStewardBaseUrl = ${shrine.shrineHubBaseUrl}/shrine-api/shrine-steward
    }
  }//steward
}//shrine
```

It is rare but possible to have a set of patient data at the hub. Simply include the adapter section of qep-and-adapter-shrine.conf in your shrine.conf , tailored to your system as explained earlier in this chapter.

Set the shrine i2b2 user password in the password.conf file in /opt/shrine/tomcat/lib .

password.conf

```
shrine.hiveCredentials.password = "changeit"
```

Next, configure the initial network structures and queues for the hub.

Download the shrine network lifecycle tool into /opt/shrine:

```
cd /opt/shrine


wget https://repo.open.catalyst.harvard.edu/nexus/content/groups/public/net/shrine/shrine-network-lifecycle-tool
/4.0.0/shrine-network-lifecycle-tool-4.0.0-dist.zip -O shrine-network-lifecycle.zip unzip shrine-network-
lifecycle.zip

cd shrine-network-lifecycle
```

Inside the conf directory, edit the override.conf file to use your database username and password:

```
shrine {
  queryEntryPoint{
    audit {
      database {
        dataSourceFrom = "testDataSource" //Can be JNDI or testDataSource . Use testDataSource for tests and
command line tools, JNDI everywhere else
        testDataSource {
          driverClassName = "com.mysql.cj.jdbc.Driver" //JDBC driver class name
          url = "jdbc:mysql://localhost:3306/qepAuditDB?serverTimezone=UTC" //URL to your database
          credentials {
            username = "yourUserName"
            password = "yourDatabasePassword"
```

Next, create a file named network.conf to meet your needs. At a minimum include the network section and a section for the hub's QEP:

```
shrine {
  network {
    network {
      name = "Network Name"
      hubQueueName = "hub"
      adminEmail = "yourEmail@yourhospital.edu"
      momId = "HubQueue"
    }
    nodes = [
      {
        name = "Hub's QEP"
        key = "hub-qep"
        userDomainName = "network-hub"
        queueName = "shrinehub"
        sendQueries = "false"
        adminEmail = "yourEmail@yourhospital.edu"
        momId = "HubQepQueue"
      }
    ]
  }
}
```

Finally use the shrineLifecycle tool to set up the network:

```
./shrineLifecycle createNetwork network.conf
```