Chapter 8.4 - Configuring a Hub

This section is intended for hub administrators only. For the ACT network, the hubs have already been set up and hosted at Harvard Catalyst, and remote sites should skip this section.

Here is a sample shrine.conf file for a system running SHRINE 2.0.0 , for a node supporting researchers (via the QEP) and distributing queries (as the network's hub). (It is rare but possible to have a set of patient data at the hub. Simply include the adapter section of shrine.conf from Chapter 8 in your shrine.conf):

```
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"

shrine {

  nodeKey = "somethingHub" //node key to get information from the hub about itself as a node.
  shrineDatabaseType = "mysql" //can be mysql, sqlserver, oracle

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

  webclient {
    domain = ${i2b2Domain}
    name  = ${i2b2ShrineProjectName}
    siteAdminEmail = "shrine-admin@example.com"
  }
  pmEndpoint {
    url = ${i2b2BaseUrl}/i2b2/services/PMService/getServices
  }

  ontEndpoint {
    url = ${i2b2BaseUrl}/i2b2/services/OntologyService
  }

  hiveCredentials {
    domain = ${i2b2Domain}
    username = "demo"
    password = "changeit"
    crcProjectId = "Demo"
    ontProjectId = ${i2b2ShrineProjectName}
  }

  breakdownResultOutputTypes {
    PATIENT_AGE_COUNT_XML {
      description = "Age patient breakdown"
    }
    PATIENT_RACE_COUNT_XML {
      description = "Race patient breakdown"
    }
    PATIENT_VITALSTATUS_COUNT_XML {
      description = "Vital Status patient breakdown"
    }
    PATIENT_GENDER_COUNT_XML {
      description = "Gender patient breakdown"
    }
  } //end breakdown section

  hub {
    create = true
    client {
      serverUrl = ${shrineHubBaseUrl}
    }

    //This part of the configuration is only used if no network is found in the hub's database
    //If no network is foundIt is loaded into the database when the hub starts.
    //To update the network or nodes, use the appropriate curl commands, not this config section.
    //Specified nodes are added only if the node key does not already exist in the hub's database.
    ifNoNetwork {
      network {
        name = "Shrine Example Network" //Name of your network
        hubQueueName = "hub" //queue used to send messages to the hub, different from the queue used to send
messages to a QEP and adapter colocated with the hub
        adminEmail = "yourname@example.com"
      }
      //Nodes in this network
      //In ifNoNetwork - nodes are added only if the node key does not already exist in the hub's database.
      nodes = [
        {
          name = "Network Hub", //human-readable name for this node
          key = "somethingHub", //machine-friendly key used to identify this node. Never change this.
          userDomainName = "example.com", //domain name for users from this node.
          adminEmail = "yourname@example.shrine-dev1.com", //the email address for the admin of this node
          queueName = "exampleHub", //queue used to send messages to the qep and adapter at this node.
          sendQueries = "false" //true to send queries to an adapter at this node. An optional field, true by
default.
        }
      ]
    }

    networkHealth {
      email {
        networkName = "Enter network name"
        networkSignature = "Enter network signature"
        grantDescription = "Enter description of grant"
        helpContactName = "Enter help contact name"
        helpContactEmail = "yourname@example.com" //The email address for the node admin to contact for help if
there are questions about an error that occurred after running the connectivity test
      }
      webpage {
        networkName = "Enter network name"
        webPageNote = "This is a note" //Optional alert message to display at the bottom of the network health
webpage
        alertMessage = "This is an alert message"  //Optional alert message to display at the top of the
network health webpage
      }
      connectivityTest {
        //queryId = "123456789"  //The query Id of a previously run query to copy for the connectivity test.
Leave this out for now, come back and set it later
        interval = "24 hours" //How often to run the connectivity test
        timeLimit = "10 minutes"  //How long to wait for the query to finish running. If the query does not
finish by this time, it will be marked as an error.
      }
    }
  } //end network health section

  queryEntryPoint {
    broadcasterServiceEndpoint {
      url = ${shrineHubBaseUrl}/shrine/rest/broadcaster/broadcast
    }
  }


  adapter {
    create = false
  } //end adapter section

  keystore {
    file = "/opt/shrine/shrine.keystore"
    password = "changeit"
    privateKeyAlias = "shrine-hub"
    keyStoreType = "JKS"
    caCertAliases = ["shrine-ca"]
  } //end keystore section

  steward {
    createTopicsMode = Approved //the default is Pending - the most secure - but most sites use Approved

    emailDataSteward {
      sendAuditEmails = false  //false to turn off the whole works of emailing the data steward
      //interval = "1 day" //Audit researchers daily
      //timeAfterMidnight = "6 hours" //Audit researchers at 6 am. If the interval is less than 1 day then this
delay is ignored.
      //maxQueryCountBetweenAudits = 30 //If a researcher runs more than this many queries since the last audit
audit her
      //minTimeBetweenAudits = "30 days" //If a researcher runs at least one query, audit those queries if this
much time has passed

      //You must provide the email address of the shrine node system admin, to handle bounces and invalid
addresses
      //from = "shrine-admin@example.com"
      //You must provide the email address of the data steward
      //to = "shrine-steward@example.com"

      //subject = "Audit SHRINE researchers"
      //The baseUrl for the data steward to be substituted in to email text. Must be supplied if it is used in
the email text.
      //stewardBaseUrl = "https://example.com:6443/shrine-api/steward/"
      externalStewardBaseUrl = "https://example.com:6443/shrine-api/steward/"

      //Text to use for the email audit.
      //AUDIT_LINES will be replaced by a researcherLine for each researcher to audit.
      //STEWARD_BASE_URL will be replaced by the value in stewardBaseUrl if available.
      //emailBody = """Please audit the following users at STEWARD_BASE_URL at your earliest convenience:
AUDIT_LINES"""
      //note that this can be a multiline message

      //Text to use per researcher to audit.
      //FULLNAME, USERNAME, COUNT and LAST_AUDIT_DATE will be replaced with appropriate text.
      //researcherLine = "FULLNAME (USERNAME) has run COUNT queries since LAST_AUDIT_DATE."
    }
  } //end steward section

} //end shrine
```