

Sqoop Server Internals

This document details the internals of how the sqoop-server works.



This document is relevant to the release 1.99.5. Further changes can happen in future releases

- [Sqoop Tomcat Server](#)
- [Sqoop Server](#)
- [Sqoop Servlets](#)
- [Sqoop Request Handlers](#)
- [Sqoop Client](#)

Sqoop Tomcat Server

- Sqoop-server uses tomcat web server, it is very bare bones.
- The main entry point is the [TomcatToolRunner](#), it bootstraps the tomcat and loads all the sqoop related classes into its class path. It is invoked from the bash script .

```
/sqoop.sh server start
```

- The main hook for the sqoop server to start is this entry in the web.xml. Tomcat invokes it callbacks as it bootstraps and we use the contextInitialized callback to initialize all the related code.

```
<!-- Listeners -->
<listener>
  <listener-class>org.apache.sqoop.server.ServerInitializer</listener-class>
</listener>
```

Sqoop Server

- The sqoop server is represented by the java class `SqoopServer.java`
- `SqoopServer.initialize()` is called from the `ServerInitiaizer`
- `SqoopServer.destroy()` is called when the tomcat server is shutdown

Sqoop Servlets

- We have servlets for each Sqoop Entity that is exposed via the REST API in [web.xml](#). Refer to [Sqoop 2 \(1.99.4\) Entity Nomenclature and Relationships](#) for more details on the supported Sqoop Entities.
- They receive the requests from the web or the Sqoop Client and process the request.
- They delegate most of the business logic to their corresponding `RequestHandler` implementations.
- All the rest APIs supported via the sqoop-server are documented here: <http://sqoop.apache.org/docs/1.99.4/RESTAPI.html#id1>

```
<!-- Version servlet -->
<servlet>
  <servlet-name>VersionServlet</servlet-name>
  <servlet-class>org.apache.sqoop.server.VersionServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>VersionServlet</servlet-name>
  <url-pattern>/version</url-pattern>
</servlet-mapping>
<!-- Generic Configurable servlet -->
<servlet>
  <servlet-name>v1.ConfigurableServlet</servlet-name>
  <servlet-class>org.apache.sqoop.server.v1.ConfigurableServlet</servlet-class>
```

```

    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>v1.ConfigurableServlet</servlet-name>
    <url-pattern>/v1/configurable/*</url-pattern>
</servlet-mapping>
<!-- Connector servlet -->
<servlet>
    <servlet-name>v1.ConnectorServlet</servlet-name>
    <servlet-class>org.apache.sqoop.server.v1.ConnectorServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>v1.ConnectorServlet</servlet-name>
    <url-pattern>/v1/connector/*</url-pattern>
</servlet-mapping>
<!-- Connectors servlet -->
<servlet>
    <servlet-name>v1.ConnectorsServlet</servlet-name>
    <servlet-class>org.apache.sqoop.server.v1.ConnectorServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>v1.ConnectorsServlet</servlet-name>
    <url-pattern>/v1/connectors/*</url-pattern>
</servlet-mapping>
<!-- Driver servlet -->
<servlet>
    <servlet-name>v1.DriverServlet</servlet-name>
    <servlet-class>org.apache.sqoop.server.v1.DriverServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
.....
<!-- Job servlet -->
<servlet>
    <servlet-name>v1.JobServlet</servlet-name>
    <servlet-class>org.apache.sqoop.server.v1.JobServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>v1.JobServlet</servlet-name>
    <url-pattern>/v1/job/*</url-pattern>
</servlet-mapping>
<!-- Jobs servlet -->
<servlet>
    <servlet-name>v1.JobsServlet</servlet-name>
    <servlet-class>org.apache.sqoop.server.v1.JobsServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>v1.JobsServlet</servlet-name>
    <url-pattern>/v1/jobs/*</url-pattern>
</servlet-mapping>
<!-- Submissions servlet -->
<servlet>
    <servlet-name>v1.SubmissionsServlet</servlet-name>
    <servlet-class>org.apache.sqoop.server.v1.SubmissionsServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>v1.SubmissionsServlet</servlet-name>
    <url-pattern>/v1/submissions/*</url-pattern>
</servlet-mapping>
</web-app>

```

- There is an authentication filter to authenticate all request.

```
<!-- Filter -->
<filter>
  <filter-name>authFilter</filter-name>
  <filter-class>org.apache.sqoop.filter.SqoopAuthenticationFilter</filter-class>
</filter>
```

- There are two authentication mode supported: simple and Kerberos, which could be set in the sqoop.properties.

```
#
# Authentication configuration
#
org.apache.sqoop.security.authentication.type=SIMPLE
org.apache.sqoop.security.authentication.handler=org.apache.sqoop.security.Authentication.
SimpleAuthenticationHandler
org.apache.sqoop.security.authentication.anonymous=true

#org.apache.sqoop.security.authentication.type=KERBEROS
#org.apache.sqoop.security.authentication.handler=org.apache.sqoop.security.Authentication.
KerberosAuthenticationHandler
#org.apache.sqoop.security.authentication.kerberos.principal=sqoop/_HOST@NOVALLOCAL
#org.apache.sqoop.security.authentication.kerberos.keytab=/home/kerberos/sqoop.keytab
#org.apache.sqoop.security.authentication.kerberos.http.principal=HTTP/_HOST@NOVALLOCAL
#org.apache.sqoop.security.authentication.kerberos.http.keytab=/home/kerberos/sqoop.keytab
#org.apache.sqoop.security.authentication.enable.doAs=true
#org.apache.sqoop.security.authentication.proxyuser.#USER#.users=*
#org.apache.sqoop.security.authentication.proxyuser.#USER#.groups=*
#org.apache.sqoop.security.authentication.proxyuser.#USER#.hosts=*
```

Sqoop Request Handlers

Each Sqoop Servlet has its corresponding handler class that handles the request for that servlet. It then internally calls the internal sqoop core/ common code.

```
public interface RequestHandler {
    static final String CONNECTOR_NAME_QUERY_PARAM = "cname";
    static final String JOB_NAME_QUERY_PARAM = "jname";
    JsonBean handleEvent(RequestContext ctx);
}
public class ConnectorRequestHandler implements RequestHandler {
    ...
}
```

Sqoop Client

- Sqoop Client is represented by the java class `SqoopClient.java`
- It has wrapper `ResourceRequest` classes for each sqoop entity, they encapsulate the request/postBody parameters to be sent in the request. Refer to [Sqoop 2 \(1.99.4\) Entity Nomenclature and Relationships](#) for more details on the supported Sqoop Entities.
- It used the bare bones `URLConnection` object to make requests to the Sqoop-server.

```
URLConnection conn = new DelegationTokenAuthenticatedURL().openConnection(url, authToken);
```



SqoopClient used to use jersey REST client for making tomcat requests. Recently it was switched to Hadoop-auth/SPENGO for adding Kerberos support that are documented here

<https://cwiki.apache.org/confluence/display/SQOOP/Security+Guide+On+Sqoop+2>

- Run command to start Sqoop Client.

```
/sqoop.sh client
```

- In Kerberos Authentication mode. Kinit is required to set Kerberos environment.

```
kinit sqoop/server-fqdn@HADOOP.COM
```