**Steps to create Simple Application**

**Step 1 − Install the simplity kernel**

Clone the repo <https://github.com/simplity/simplity> and then run mvn install within the cloned git folder to install the simplity kernel in your local repo.

**Step 2 – Creating Project**

* Create a new "Maven Project" in eclipse.
* Configure web.xml to have the following entry.

|  |
| --- |
| <servlet>  <servlet-name>SimplityStartup</servlet-name>  <servlet-class>org.simplity.http.Startup</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet>  <servlet-name>s</servlet-name>  <servlet-class>org.simplity.http.Serve</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>s</servlet-name>  <url-pattern>\*.\_s</url-pattern>  </servlet-mapping> |

* Other servlet-mappings are
* For authentication

|  |
| --- |
| <servlet>  <servlet-name>i</servlet-name>  <servlet-class>org.simplity.http.DefaultLogin</servlet-class>  </servlet>  <servlet>  <servlet-name>o</servlet-name>  <servlet-class>org.simplity.http.DefaultLogout</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>i</servlet-name>  <url-pattern>\*.\_i</url-pattern>  </servlet-mapping>  <servlet-mapping>  <servlet-name>o</servlet-name>  <url-pattern>\*.\_o</url-pattern>  </servlet-mapping> |

* For file upload/download

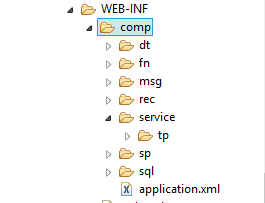
|  |
| --- |
| <servlet>  <servlet-name>f</servlet-name>  <servlet-class>org.simplity.http.Stream</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>f</servlet-name>  <url-pattern>\*.\_f</url-pattern>  </servlet-mapping> |

* Configure pom.xml as follows to access the kernel in application.

|  |
| --- |
| <dependency>  <groupId>org.simplity</groupId>  <artifactId>kernel</artifactId>  <version>x.x.x</version>  </dependency> |

**NOTE:** Version no. should be the version of kernel in your local repo

* Please make sure that folder structure under WEB-INF should be as follows.



* comp folder is where we will be adding components of our applications.
* Copy simplity folder under webapp folder. This has just one .js file that we are going to use on the client.

**Step 3 – Application Configuration**

* Configure application.xml under WEB-INF as follows.

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <application  applicationId=*"HelloWorld"*  autoLoginUserId=*"100"*  xmlns=*"http://www.simplity.org/schema"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*>  </application> |

**NOTE:** applicationId should be a unique name for the application.

**Step 4 – Creating the client.**

* Create helloWorld.html with the following content

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <script type=*"text/javascript"* src=*"js/simplity.js"*></script>  </head>  <body>  <button onclick="submit()">Fetch</button>  <script type=*"text/javascript"*>  **var** submit = **function**(){  Simplity.getResponse("helloWorld",**null**,onSuccess);  }  **var** onSuccess = **function**(json){  console.log(json);  }  </script>  </body>  </html> |

* To send any data from client pass the data in Simplity.getResponse(serviceName,dataToSend,successCallbackFunction,errorCallbackFunction)
* Data should be in JSON format.

**Step 5 – Creating the service**.

* Create a file named helloWorld.xml under comp/service/tp with the following content.

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <service name=*"helloWorld"*  xmlns=*"http://www.simplity.org/schema"*>  <actions>  <setValue fieldName=*"hello"* fieldValue=*"Hello World !!"*></setValue>  </actions>  <outputData fieldNames=*"hello"*></outputData>  </service> |

* To receive data from client add inputData to service as follows

|  |
| --- |
| <inputData>  <inputFields>  <inputField dataType=*"\_text"* name=*"employeeName"*/>  <inputField dataType=*”number”* name=*"employeeId"*/>  <inputFields>  </inputData> |

**NOTE:** inputField names should be the key names of JSON data which is sent from client.

*Now run the application and observe the output in browser console.*

**DB Related Transactions:**

* **DB Configuration**
* Configure DB details in application.xml as follows
* *Connection string details*

|  |
| --- |
| dbVendor=*"h2"*  dbDriverClassName=*"org.h2.Driver"*  connectionString=*"jdbc:h2:file:/db/classicmodels;user=sa;"*  logSqls=*"true"* |

**Or**

* *JNDI details*

|  |
| --- |
| dbVendor=*"h2"*  dataSourceName=*"java:comp/env/jdbc/nomination"*  logSqls=*"true"* |

* **Creating Record to store DB data**
* Create a record named Employees.xml under comp/rec with the following content.

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <record xmlns=*"http://www.simplity.org/schema"*  name=*"Employees"*  tableName=*"EMPLOYEES"*  okToSelectAll=*"true"*>  <fields>  <field name=*"employeeNumber"* dataType=*"\_number"* fieldType=*"primaryKey"*/>  <field name=*"lastName"* dataType=*"\_text"*/>  <field name=*"firstName"* dataType=*"\_text"*/>  <field name=*"email"* dataType=*"\_text*/>  <field name=*"jobTitle"* dataType=*"\_text"*/>  </fields>  </record> |

**NOTE:** *Ensure that tableName and the column/field names in record are present in the DB.*

* **Creating service to get DB data**

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <service name=*"helloWorld"*  dbAccessType=*"readWrite"*  xmlns=*"http://www.simplity.org/schema"*>  <actions>  <filter filterRecordName=*"Employees"* outputSheetName=*"employees"*>  </filter>  </actions>  <outputData>  <outputRecords>  <outputRecord sheetName=*"employees"*/>  </outputRecords>  </outputData>  </service> |

**NOTE:** *Please refer to SimplityManual.docx at* [*https://github.com/simplity/simplity/tree/master/documents*](https://github.com/simplity/simplity/tree/master/documents) *for more attributes of all the simplity xml tags.*

**Utilities:**

* **XML Template proposals**
* Copy schema folder of Simplity to WEB-INF folder.
* Schema has the grammar that will guide us in creating the components.
* Add the two .xsd files as reference using

|  |
| --- |
| windows -> preference -> xml -> xml-catalog -> user specified entries |

**Creating Simplity application using archetype:**

* Install the simplity archetype-partial at <https://github.com/simplity/simplity> in your local repo.
* Create a sample maven web application in eclipse.
* Right click on your application 🡪 run as 🡪 maven build
* In Goals provide the archetype info as below

|  |
| --- |
| archetype:generate  -DinteractiveMode=false  -DarchetypeCatalog=local  -DarchetypeGroupId=<groupid of archetype in your local repo>  -DarchetypeArtifactId=<artifactId of archetype in your local repo>  -DarchetypeVersion=<version of archetype in your local repo>  -DgroupId=<groupId of your project>  -DartifactId=<artifactId of your project>  -Dpackage=<package name you want to create in your project> |

* Apply and run the configuration to generate the simplity folder structure in your application from archetype.