

Web Services Generator Tool – Phase-I User Manual		TEM 010
Prepared By	Goverdhan/Ashwinkumar M	Documentation Type : QMS - Template
Checked By	Saurabh Bagchi	Document Ref : SDLC_TEM_User Manual
Approved By		Version No.: 1
Date of Release: 21-09-2018		No. Of Pages (Including cover page): 34

---

### Revision History

Change Description	Section Numbers Changed	Version No.	Date of Revision	Revised By	Approving Authority
Baseline Draft	All	0.1	28-03-2018	Ashwinkumar M	Censof
Phase-I	All	0.2	30-07-2018	Ashwinkumar M	Censof
Phase-I Final	All	0.3	18-09-2018	Ashwinkumar M	Censof
Phase-I Final-Extended-1	Jboss Configuration added	0.4	19-09-2018	Ashwinkumar M	Censof
Phase-I Final-Extended-2	Modification as per feedback	1	21-09-2018	Ashwinkumar M	Censof

Table of Contents

Table of Contents ..... 3

1 Introduction ..... 4

    1.1 Overview of the System ..... 4

    1.2 General Information ..... 4

    1.3 Users of this Manual..... 4

2 User Interface ..... 4

    2.2 Menu Structure ..... 5

        2.2.3 Download Bundle File..... 25

        2.2.4 Template Management ..... 26

        2.2.5 User Management..... 27

## 1 Introduction

### 1.1 Overview of the System

A tool to generate Web Service based on predefined Templates. This is for a system where many Web Services needs to be written and this tool will eliminate the tedious work of writing the Web Services, WSDL & WADL code.

### 1.2 General Information

The overview of the project is to develop a tool which will generate source code, compile source code and build the OSGI bundle file for Restful web services to deploy in the JBoss Fuse and Fabric server.

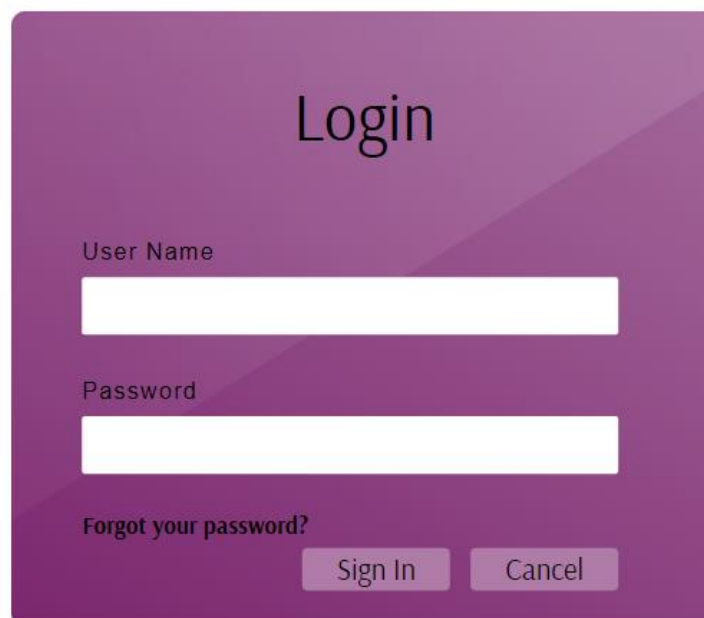
### 1.3 Users of this Manual

This Software will be used by engineers/technicians who are trained on the application.

## 2 User Interface

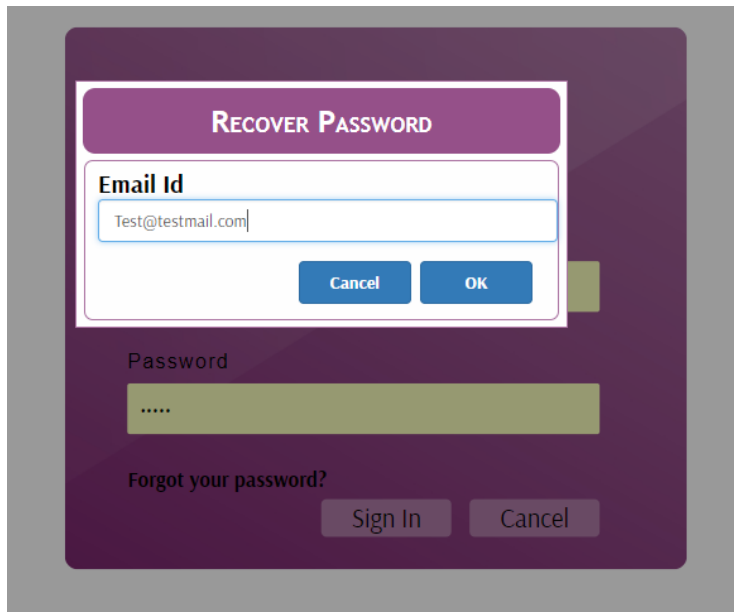
### 2.1.1 Login Screen.

- User can login with valid user Name and Password as shown below.

A purple rectangular login screen with rounded corners. At the top center, the word "Login" is written in a large, white, sans-serif font. Below it, the text "User Name" is in a smaller white font, followed by a white rectangular input field. Below that, the text "Password" is in a smaller white font, followed by another white rectangular input field. At the bottom left, the text "Forgot your password?" is in a smaller white font. At the bottom right, there are two white rectangular buttons with rounded corners; the left one says "Sign In" and the right one says "Cancel", both in a small, dark font.

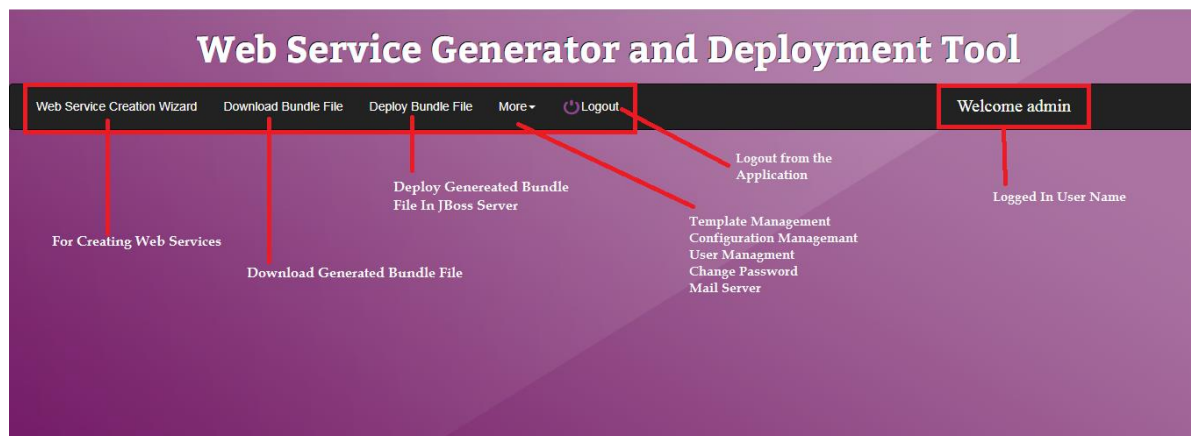
### 2.1.2 Forgot Password.

- User can be Reset the Password using Forgot Password feature.
- User will receive the email with Password after submitting the valid email-ID in Forgot Password popup.
- The user shall use the password sent in the email to login



## 2.2 Menu Structure

Below screen give the overview of menus which are available in the application.



### 2.2.1 Screen Navigation and screen descriptions per screen type (if, any)

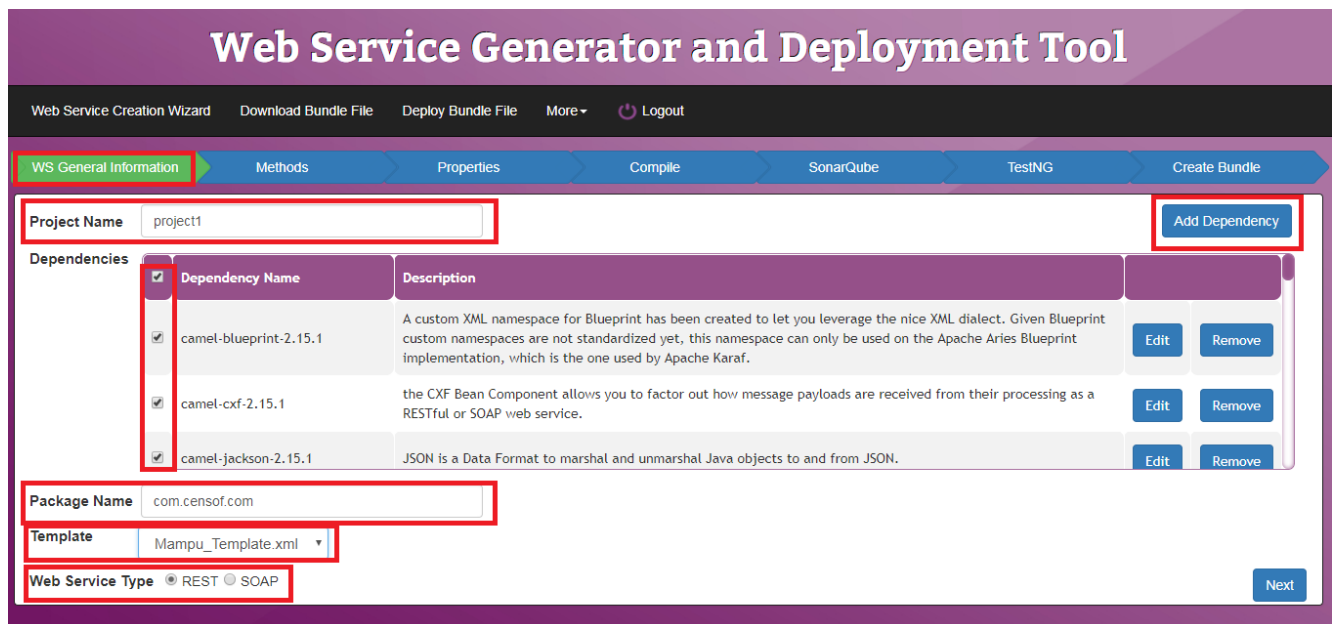
#### 2.2.2 Web Service Creation Wizard:

In Web service wizard section user can create web service using different parameters and methods as stated in following steps.

#### 2.2.2.1 Web Service General Info

User Need to Enter Web Service General Information As shown in Image.

- Enter Project Name (Mandatory)
- Select Dependencies (Mandatory)
- Enter Package Name (Mandatory)
- Select Template (Mandatory)
- Select Web Service Type (Mandatory)
- User can Add Dependencies (Optional)



**Web Service Generator and Deployment Tool**

Web Service Creation Wizard   Download Bundle File   Deploy Bundle File   More ▾   Logout

WS General Information   Methods   Properties   Compile   SonarQube   TestNG   Create Bundle

Project Name: project1   Add Dependency

Dependencies

<input checked="" type="checkbox"/>	Dependency Name	Description	
<input checked="" type="checkbox"/>	camel-blueprint-2.15.1	A custom XML namespace for Blueprint has been created to let you leverage the nice XML dialect. Given Blueprint custom namespaces are not standardized yet, this namespace can only be used on the Apache Aries Blueprint implementation, which is the one used by Apache Karaf.	Edit Remove
<input checked="" type="checkbox"/>	camel-cxf-2.15.1	the CXF Bean Component allows you to factor out how message payloads are received from their processing as a RESTful or SOAP web service.	Edit Remove
<input checked="" type="checkbox"/>	camel-jackson-2.15.1	JSON is a Data Format to marshal and unmarshal Java objects to and from JSON.	Edit Remove

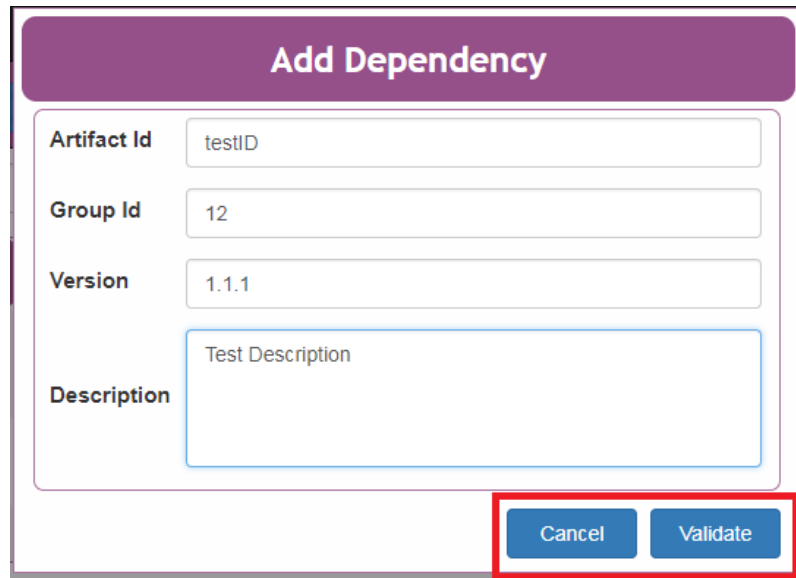
Package Name: com.censof.com

Template: Mampu\_Template.xml ▾

Web Service Type: ☒ REST ☐ SOAP

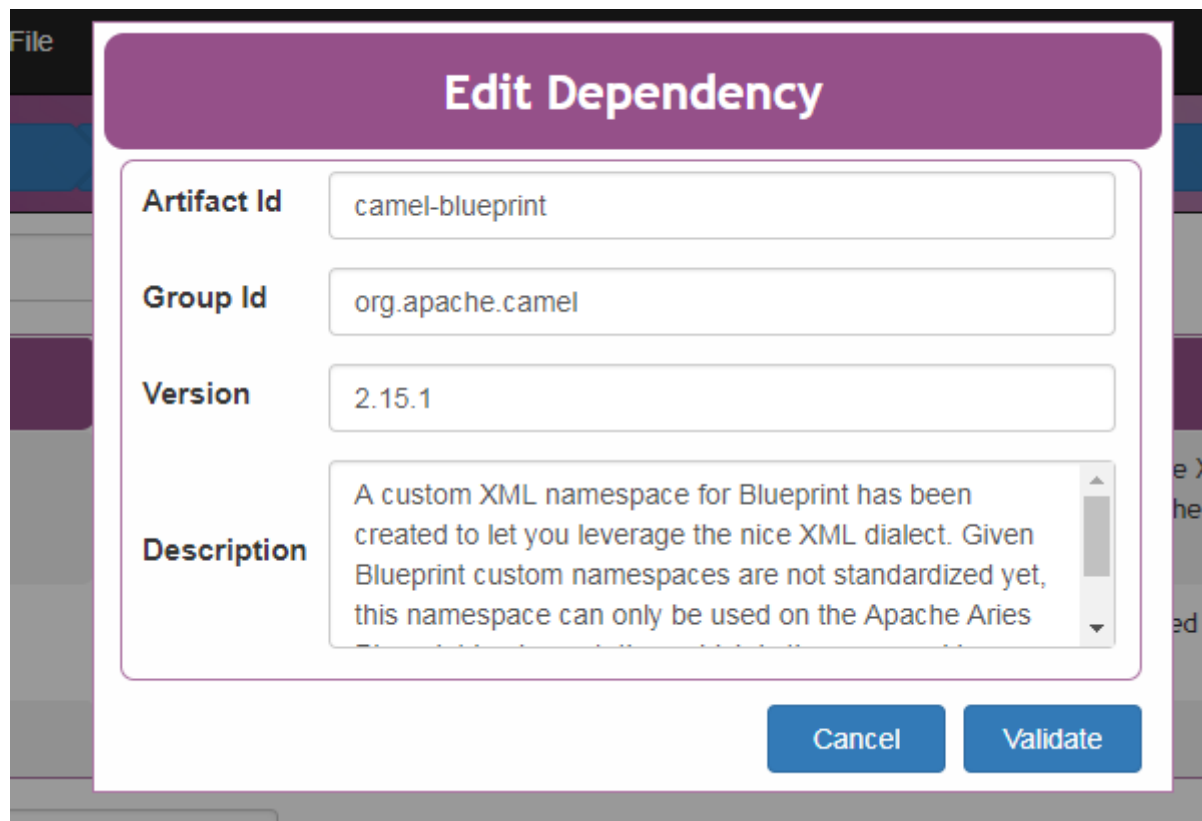
Next

##### 2.2.2.1.1 Add Dependency:



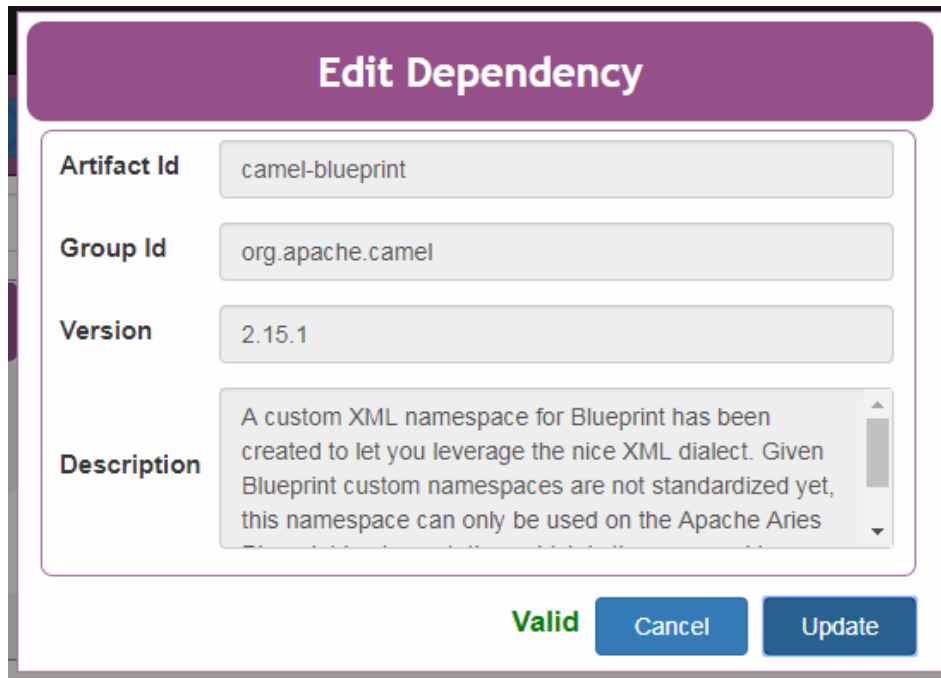
The 'Add Dependency' dialog box features a purple header with the title 'Add Dependency'. Below the header, there are four input fields: 'Artifact Id' with the value 'testID', 'Group Id' with the value '12', 'Version' with the value '1.1.1', and a larger 'Description' field containing the text 'Test Description'. At the bottom right, there are two blue buttons: 'Cancel' and 'Validate', which are highlighted by a red rectangular border.

#### 2.2.2.1.2 Edit Dependency:



The 'Edit Dependency' dialog box has a purple header with the title 'Edit Dependency'. It contains four input fields: 'Artifact Id' with 'camel-blueprint', 'Group Id' with 'org.apache.camel', and 'Version' with '2.15.1'. The 'Description' field is a text area containing the text: 'A custom XML namespace for Blueprint has been created to let you leverage the nice XML dialect. Given Blueprint custom namespaces are not standardized yet, this namespace can only be used on the Apache Aries'. At the bottom right, there are two blue buttons: 'Cancel' and 'Validate'.

#### 2.2.2.1.3 Update Dependency:



**Edit Dependency**

**Artifact Id** camel-blueprint

**Group Id** org.apache.camel

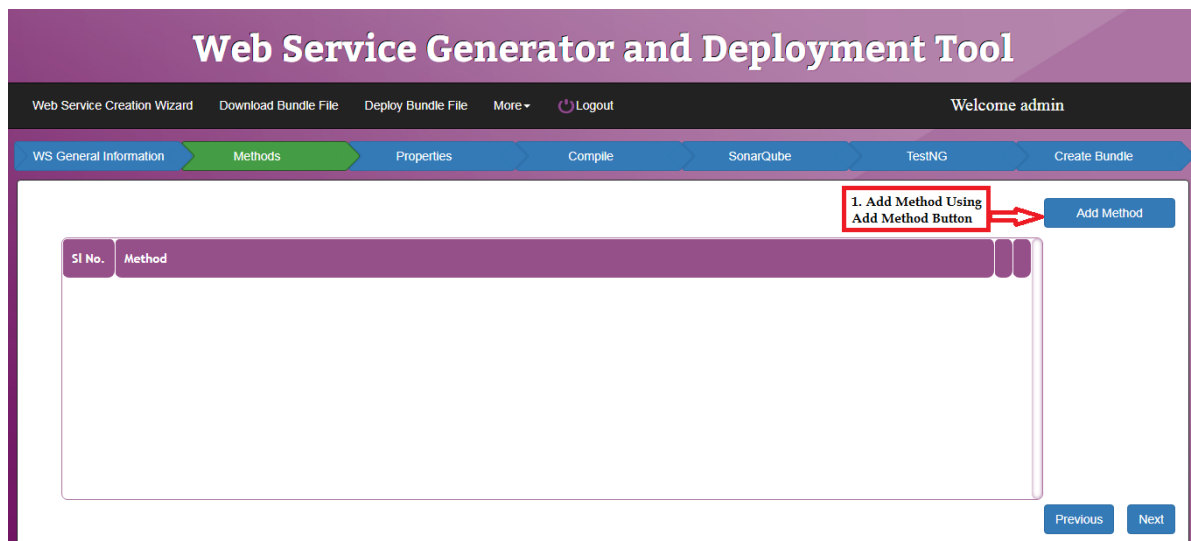
**Version** 2.15.1

**Description** A custom XML namespace for Blueprint has been created to let you leverage the nice XML dialect. Given Blueprint custom namespaces are not standardized yet, this namespace can only be used on the Apache Aries

**Valid** Cancel Update

### 2.2.2.3 Methods:

- a. Add Methods using 'Add Method' Button



**Web Service Generator and Deployment Tool**

Web Service Creation Wizard Download Bundle File Deploy Bundle File More Logout Welcome admin

WS General Information **Methods** Properties Compile SonarQube TestNG Create Bundle

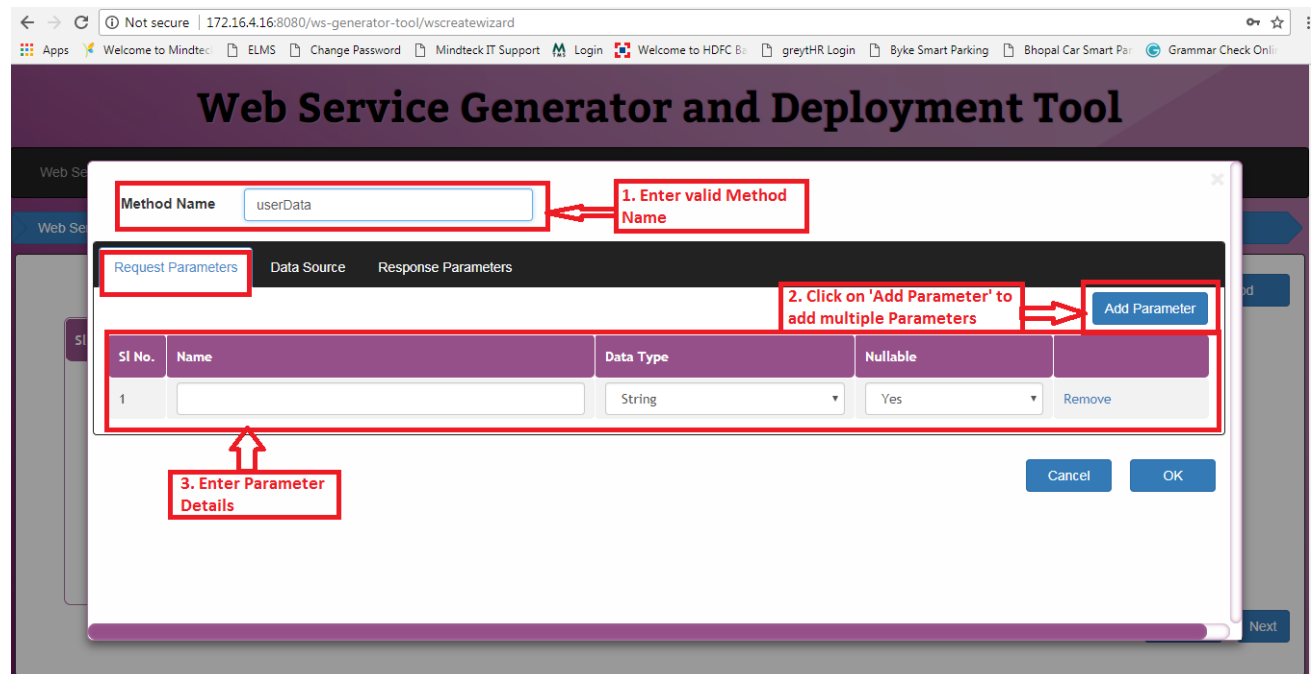
1. Add Method Using Add Method Button Add Method

Sl No.	Method
--------	--------

Previous Next

#### 2.2.2.3.1 Add Method Request Parameter





**Web Service Generator and Deployment Tool**

Method Name:

1. Enter valid Method Name

2. Click on 'Add Parameter' to add multiple Parameters

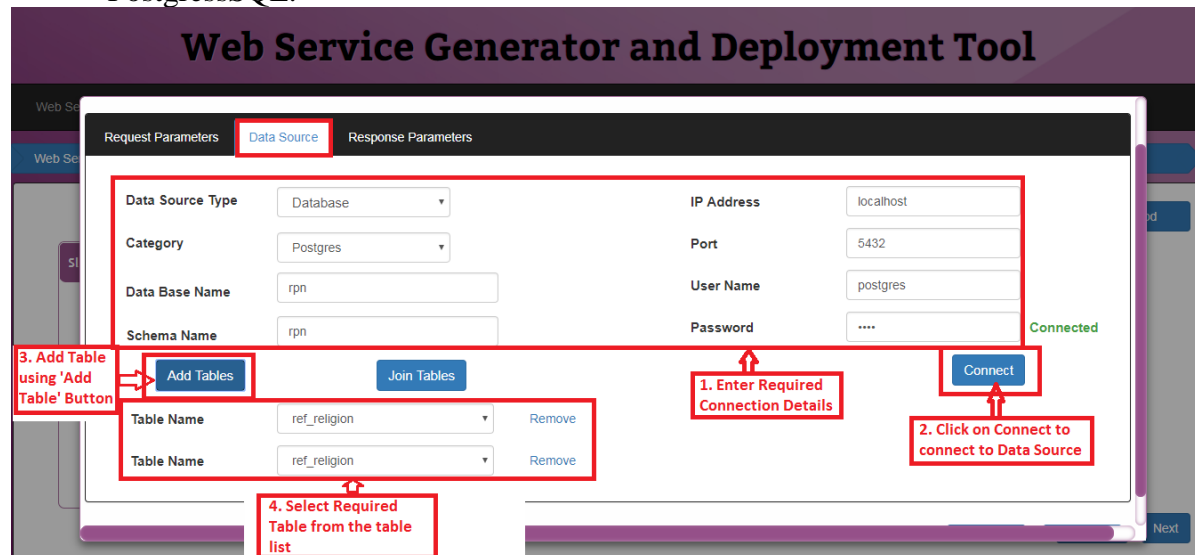
3. Enter Parameter Details

Sl No.	Name	Data Type	Nullable	
1	<input type="text"/>	String	Yes	Remove

Buttons: Add Parameter, Cancel, OK

### 2.2.2.3.2 Add Data Source

- PostgressSQL.



**Web Service Generator and Deployment Tool**

Request Parameters | **Data Source** | Response Parameters

Data Source Type: Database | IP Address: localhost

Category: Postgres | Port: 5432

Data Base Name: rpn | User Name: postgres

Schema Name: rpn | Password: ....

3. Add Table using 'Add Table' Button

1. Enter Required Connection Details

2. Click on Connect to connect to Data Source

4. Select Required Table from the table list

Buttons: Add Tables, Join Tables, Connect

Table Name	
ref_religion	Remove
ref_religion	Remove

- MSSQL

Request Parameters

Data Source

Response Parameters

Data Source Type

Database

Category

MSSQL

Data Base Name

TestDB

Add Tables

Join Tables

Table Name

Persons

Remove

Table Name

Inventory

Remove

IP Address

localhost

User Name

sa

Password

\*\*\*\*\*

Connected

Connect

- Oracle

Request Parameters

Data Source

Response Parameters

Data Source Type

Database

Category

Oracle

Service Name

TestDB

Add Tables

Join Tables

IP Address

localhost

Port

1630

User Name

SA

Password

\*\*\*\*\*

Connected

Connect

- MySQL.

Request Parameters

Data Source

Response Parameters

Data Source Type

Database

Category

MySQL

Data Base Name

wstool

Add Tables

Join Tables

Table Name

user\_roles

Remove

Table Name

user\_details

Remove

IP Address

localhost

Port

3306

User Name

root

Password

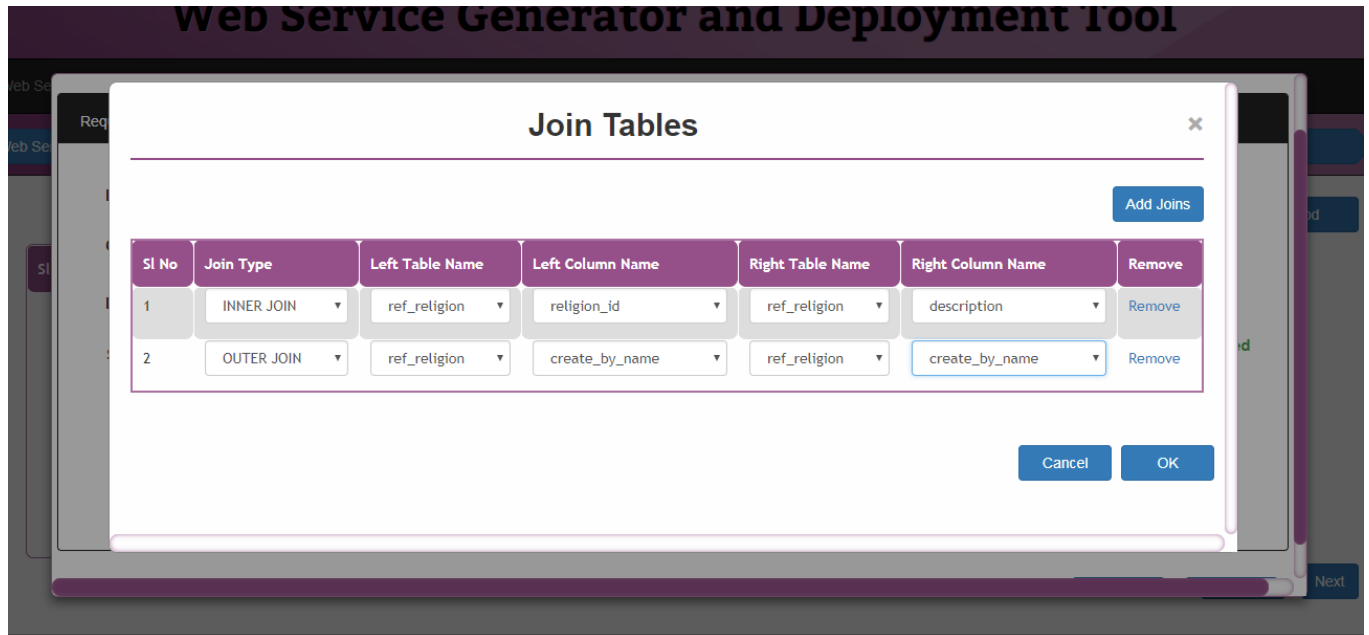
\*\*\*\*\*

Connected

Connect

### 2.2.2.3.3 Add Joins in Data Source

a. Add Required Joins shown below



Sl No	Join Type	Left Table Name	Left Column Name	Right Table Name	Right Column Name	Remove
1	INNER JOIN	ref_religion	religion_id	ref_religion	description	Remove
2	OUTER JOIN	ref_religion	create_by_name	ref_religion	create_by_name	Remove

### Examples of different types of Joins:

Tables With Common Column :- ws\_user, role, project\_details

Step1 – Joining the tables using common column/field.

1<sup>st</sup> JOIN -Inner join between the tables ws\_user and project\_details which have the common column user\_id.

2<sup>nd</sup> JOIN – Left join between the tables ws\_user and role which have the common column role\_id.

### Join Tables

Sl No	Join Type	Left Table Name	Left Column Name	Right Table Name	Right Column Name	Remove
1	INNER JOIN	ws_user	user_id	project_detail	user_id	Remove
2	LEFT JOIN	ws_user	role_id	role	role_id	Remove

Add JoinsCancelOK

Step 2- Getting the select and from clause from the response parameter table and displaying the query

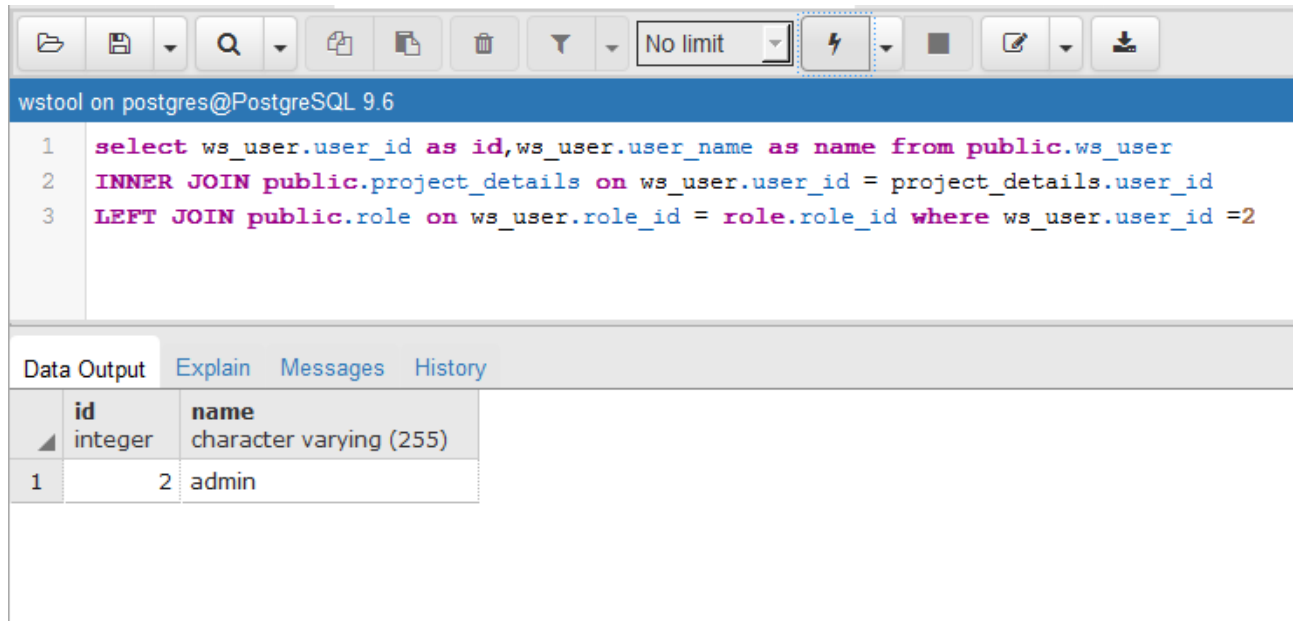
Query ::

```
select ws_user.user_id as id,ws_user.user_name as name from public.ws_user INNER JOIN public.project_details on ws_user.user_id = project_details.user_id LEFT JOIN public.role on ws_user.role_id = role.role_id where ws_user.user_id =[id]
```

ws_user	user_id	<input checked="" type="checkbox"/>		= [id]
ws_user	user_name	<input checked="" type="checkbox"/>		

View Query

Step3 – Executing query in postgres.



The screenshot shows the wstool interface for PostgreSQL 9.6. The top toolbar includes icons for file operations, search, and execution. The SQL editor contains the following query:

```
1 select ws_user.user_id as id,ws_user.user_name as name from public.ws_user
2 INNER JOIN public.project_details on ws_user.user_id = project_details.user_id
3 LEFT JOIN public.role on ws_user.role_id = role.role_id where ws_user.user_id =2
```

Below the editor, the 'Data Output' tab is active, displaying the results of the query in a table:

	id	name
	integer	character varying (255)
1	2	admin

Example2 :-

1<sup>st</sup> Join – Right join between the tables project\_details and ws\_user

2<sup>nd</sup> Join – Full join between the tables ws\_user and role.

### Join Tables

Sl No	Join Type	Left Table Name	Left Column Name	Right Table Name	Right Column Name	Remove
1	RIGHT JOIN	project_detail	user_id	ws_user	user_id	Remove
2	FULL JOIN	ws_user	role_id	role	role_id	Remove

Add JoinsCancelOK

Query for example 2.

Query ::

```
select project_details.project_id as ProjId,ws_user.user_name as UserName,role.role_id as roleId from public.project_details RIGHT JOIN public.ws_user on project_details.user_id = ws_user.user_id FULL JOIN public.role on ws_user.role_id = role.role_id where project_details.project_id =[id]
```

roleId

Integer

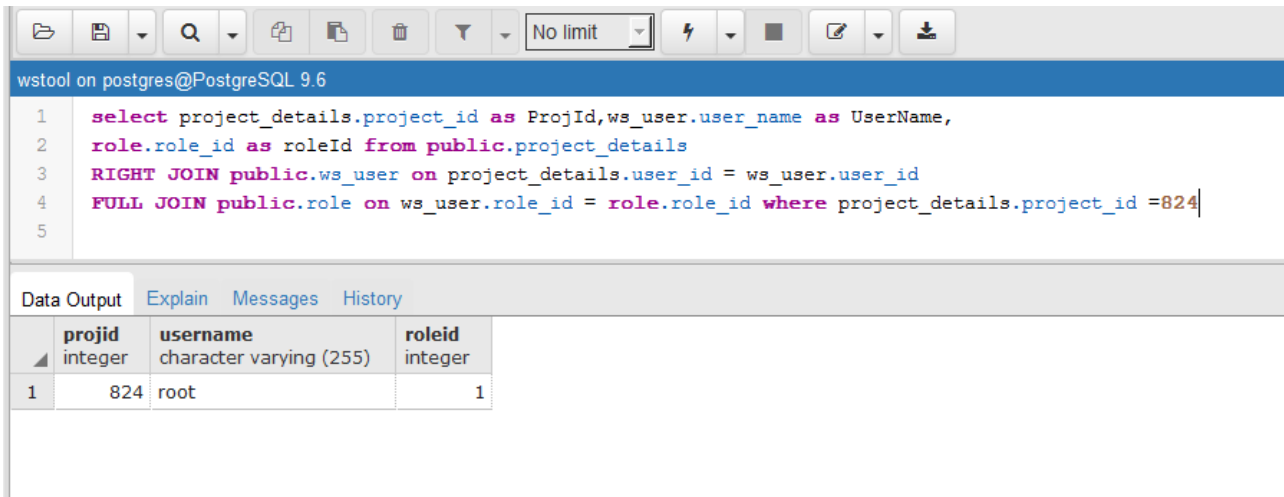
role

role\_id

☒

View Query

Result for the above query.



wstool on postgres@PostgreSQL 9.6

```

1  select project_details.project_id as ProjId,ws_user.user_name as UserName,
2  role.role_id as roleId from public.project_details
3  RIGHT JOIN public.ws_user on project_details.user_id = ws_user.user_id
4  FULL JOIN public.role on ws_user.role_id = role.role_id where project_details.project_id =824
5

```

Data Output Explain Messages History

	projid	username	roleid
	integer	character varying (255)	integer
1	824	root	1

Example 3.

1<sup>st</sup> Join – Inner Join between the tables role and ws\_user.

@nd Join- Full Join between the tables ws\_user and project\_details.



### Join Tables

Sl No	Join Type	Left Table Name	Left Column Name	Right Table Name	Right Column Name	Remove
1	INNER JOIN	role	role_id	ws_user	role_id	Remove
2	FULL JOIN	ws_user	user_id	project_detail	user_id	Remove

Buttons: Add Joins, Cancel, OK

Query For Example 3.

Query ::

```
select role.role_id as RoleId,ws_user.first_name as UserFirstName,project_details.project_name as ProjName from public.role INNER JOIN public.ws_user on role.role_id = ws_user.role_id FULL JOIN public.project_details on ws_user.user_id = project_details.user_id where role.role_id =[id]
```

ProjName String project\_details project\_name ☒

[View Query](#)

Result for the above query.

ws tool on postgres@PostgreSQL 9.6

```
1 select role.role_id as RoleId,ws_user.first_name as UserFirstName,project_details.project_name as ProjName from public.role
2 INNER JOIN public.ws_user on role.role_id = ws_user.role_id
3 FULL JOIN public.project_details on ws_user.user_id = project_details.user_id where role.role_id =1
```

Data Output Explain Messages History

	roleid integer	userfirstname character varying (255)	projname character varying (255)
1	1	root	test
2	1	admin	test1

Example 4.

1<sup>st</sup> Join – Left join between the tables project\_details and ws\_user.

2<sup>nd</sup> Join- Right join between the tables role and ws\_user.

**Join Tables**

Sl No	Join Type	Left Table Name	Left Column Name	Right Table Name	Right Column Name	Remove
1	LEFT JOIN	project_detail	user_id	ws_user	user_id	<a href="#">Remove</a>
2	RIGHT JOIN	ws_user	role_id	role	role_id	<a href="#">Remove</a>

[Add Joins](#)

[Cancel](#) [OK](#)



Query for Example 4.

Query ::

```
select project_details.project_id as ProjId,ws_user.is_active as UserActive,role.role_name as RoleName from public.project_details LEFT JOIN public.ws_user on project_details.user_id = ws_user.user_id RIGHT JOIN public.role on ws_user.role_id = role.role_id where project_details.project_id =[id]
```

View Query

Result for the above query.

```
wstool on postgres@PostgreSQL 9.6

1 select project_details.project_id as ProjId,ws_user.is_active as UserActive,role.role_name as RoleName from public.project_details
2 LEFT JOIN public.ws_user on project_details.user_id = ws_user.user_id
3 RIGHT JOIN public.role on ws_user.role_id = role.role_id where project_details.project_id =824
```

Data Output Explain Messages History

	projid	useractive	rolename
	integer	boolean	character varying (255)
1	824	true	ADMIN

#### 2.2.2.3.4 Add Response Parameter

**Web Service Generator and Deployment Tool**

Method Name:

Request Parameters | Data Source | **Response Parameters**

1. Click on 'Add Response Parameter' button

Add Response Parameter

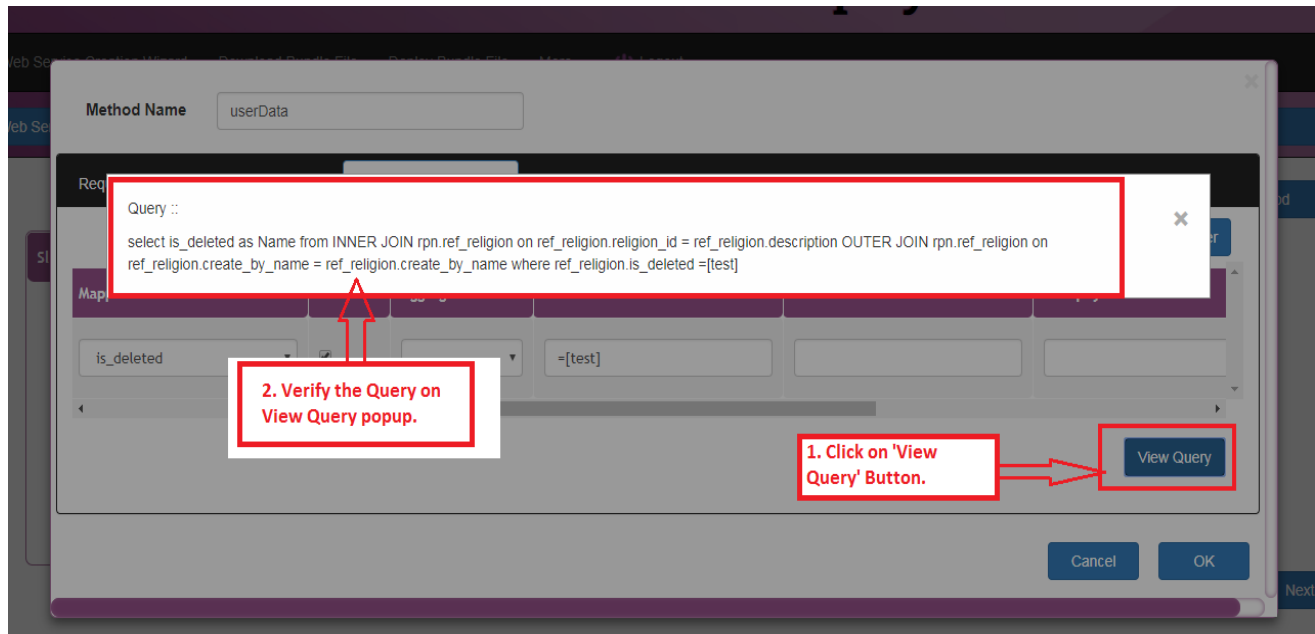
Sl No.	Name	Data Type	Table Name	Mapped Data Field	Visible	Aggregate	Where
1	<input type="text"/>	String	ref_religion	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

2. Enter Query to get response from data source

View Query

Cancel OK

#### 2.2.2.3.5 View Query for Response parameter



#### 2.2.2.3.6 View Query for Response parameter

	Data Type	Table Name	Mapped Data Field	Visible	Aggregate	Where Condition
	String	ref_religion	is_deleted	<input checked="" type="checkbox"/>		=[test]

View Query

1. Click on OK button to complete Method Creation

OK

### 2.2.2.3.7 Navigate Properties page from Methods page

- User can edit the method which is create by using edit feature.
- Navigate to Properties page using 'Next' button.

The screenshot shows the 'Web Service Generator and Deployment Tool' interface. At the top, there's a navigation bar with links: 'Web Service Creation Wizard', 'Download Bundle File', 'Deploy Bundle File', 'More', and 'Logout'. Below this is a progress bar with five steps: 'Web Service General Information', 'Methods' (highlighted in green), 'Properties', 'Compile', and 'Create Bundle'. The main content area is titled 'Methods' and contains a table with the following data:

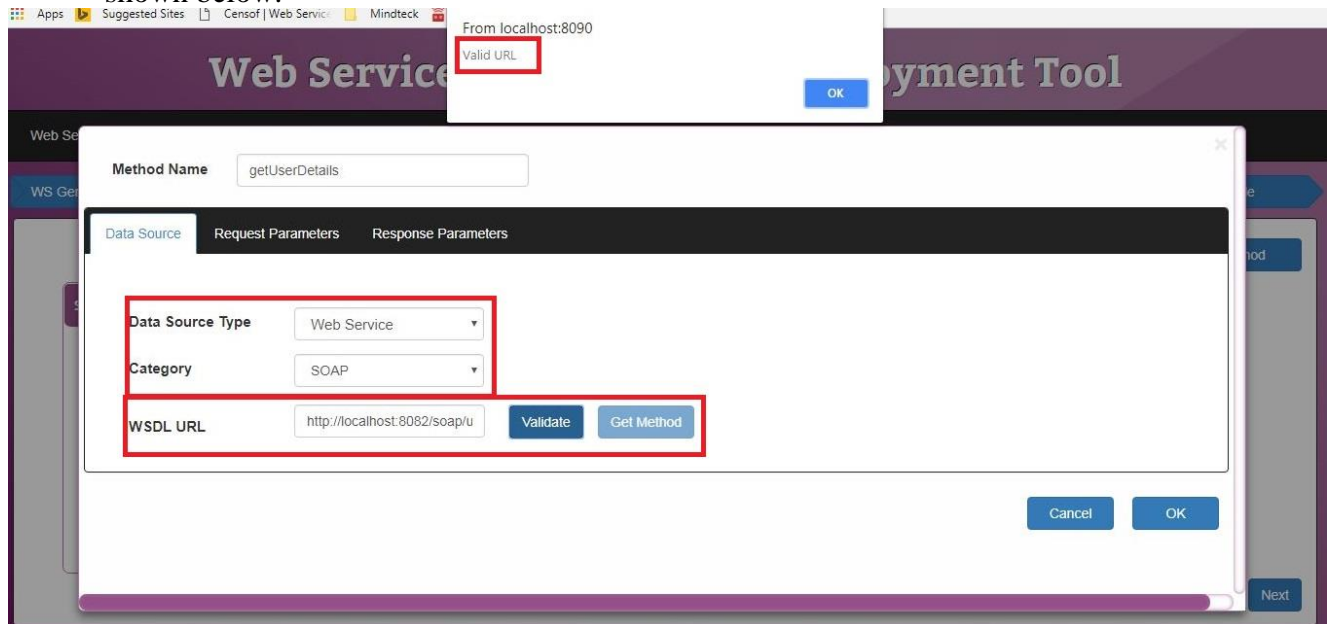
Sl No.	Method		
1	userData(test)	Edit	Remove

Annotations in the image include:

- A red box around the table with an arrow pointing to it and the text: "1. Methos will be displayed on the section".
- A red box around the 'Next' button with an arrow pointing to it and the text: "2. Click on Next after method is added".

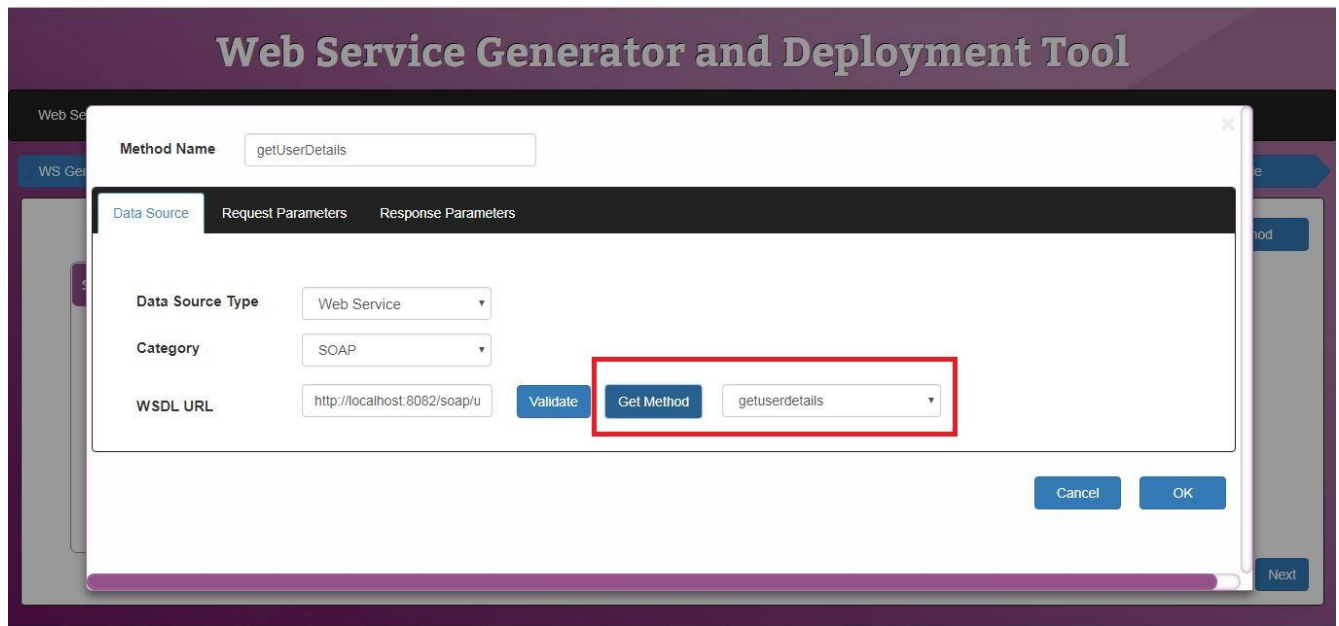
#### 2.2.2.4.1 Add Web Service as Data source

- Select Data Source as SOAP.
- User can enter WSDL URL and validate it as shown below and validate the URL as shown below.



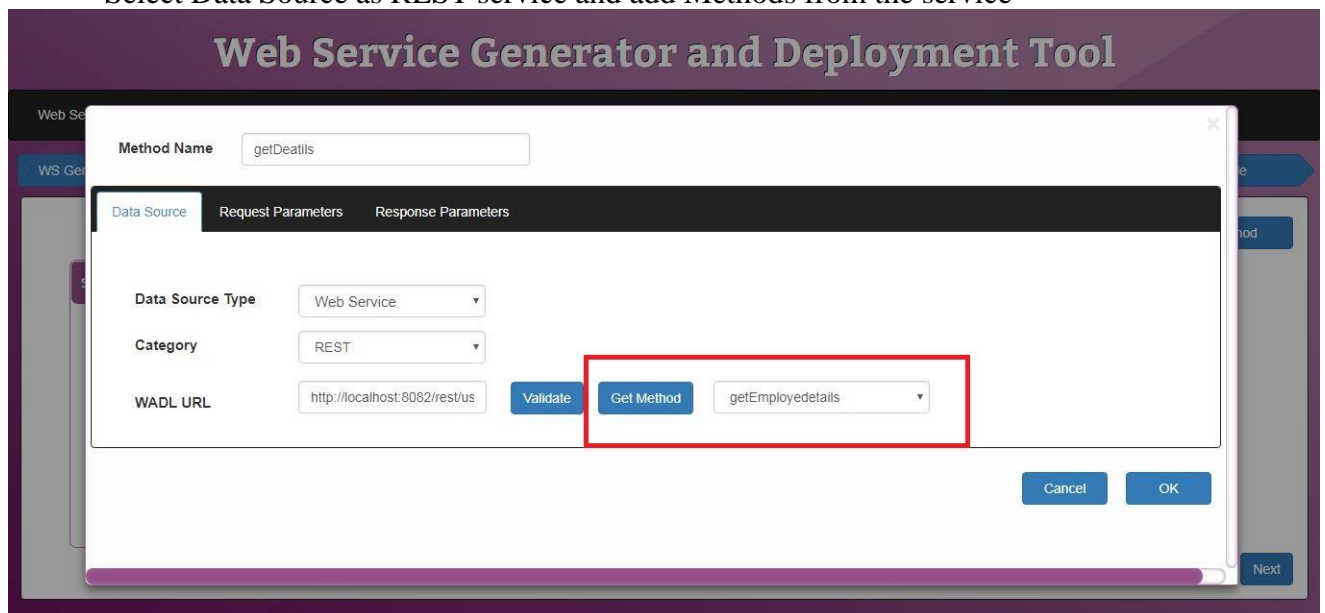
#### 2.2.2.4.2 Add Web Service as Data source

- If the URL is valid user can see the Methods dropdown and can select any method.



#### 2.2.2.5.1 Add Web Service as Data source

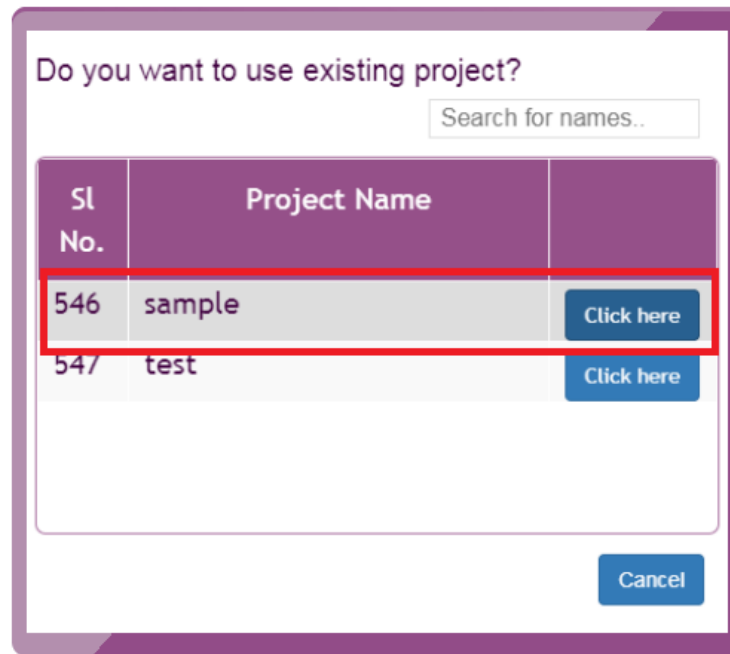
- Select Data Source as REST service and add Methods from the service



#### 2.2.2.5.2 Resume User Activity

- User can retrieve the activity if signed out while creating web service.
- If user logged out and logged in again in to the application User will see a popup with the list of old pending projects.

- Project creation previous data will be retrieved after User clicks on 'Click Here' button of the project.



Do you want to use existing project?

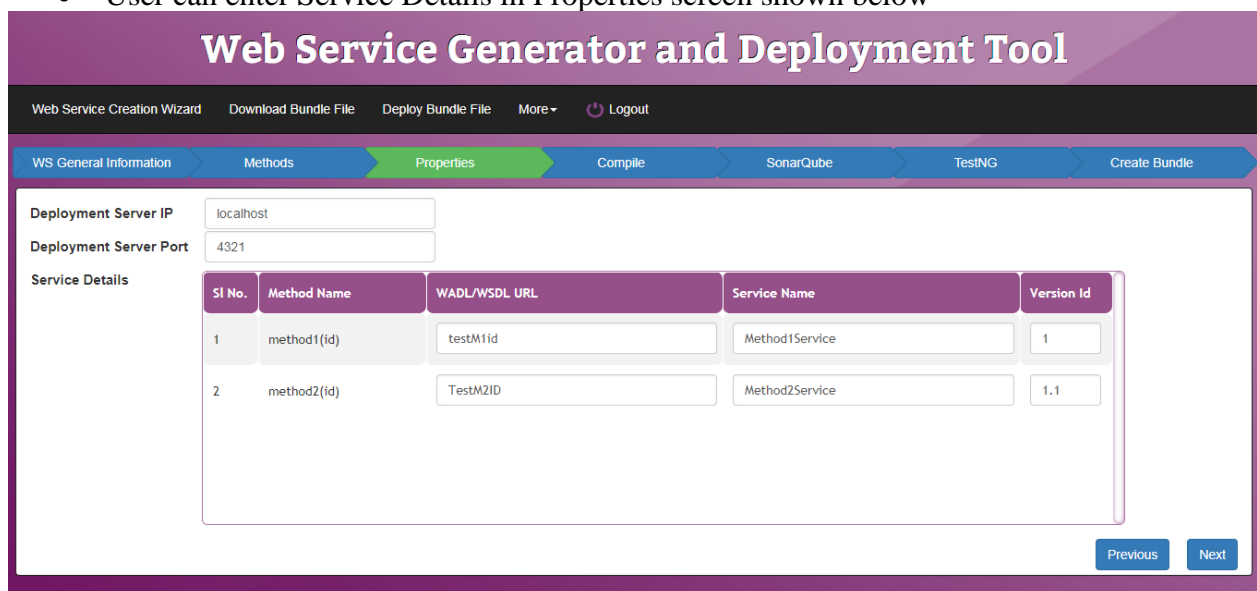
Search for names..

Sl No.	Project Name	
546	sample	<a href="#">Click here</a>
547	test	<a href="#">Click here</a>

Cancel

#### 2.2.2.6 Properties page

- User can enter Deployment Server details
- User can enter Service Details in Properties screen shown below



**Web Service Generator and Deployment Tool**

Web Service Creation Wizard   Download Bundle File   Deploy Bundle File   More ▾   Logout

WS General Information   Methods   **Properties**   Compile   SonarQube   TestNG   Create Bundle

Deployment Server IP: localhost

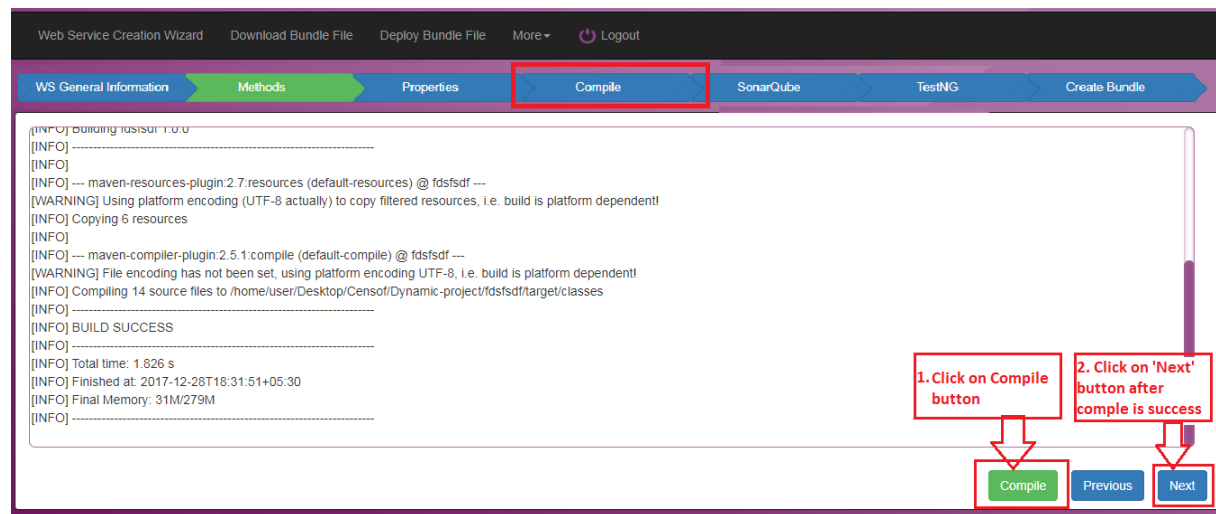
Deployment Server Port: 4321

Service Details

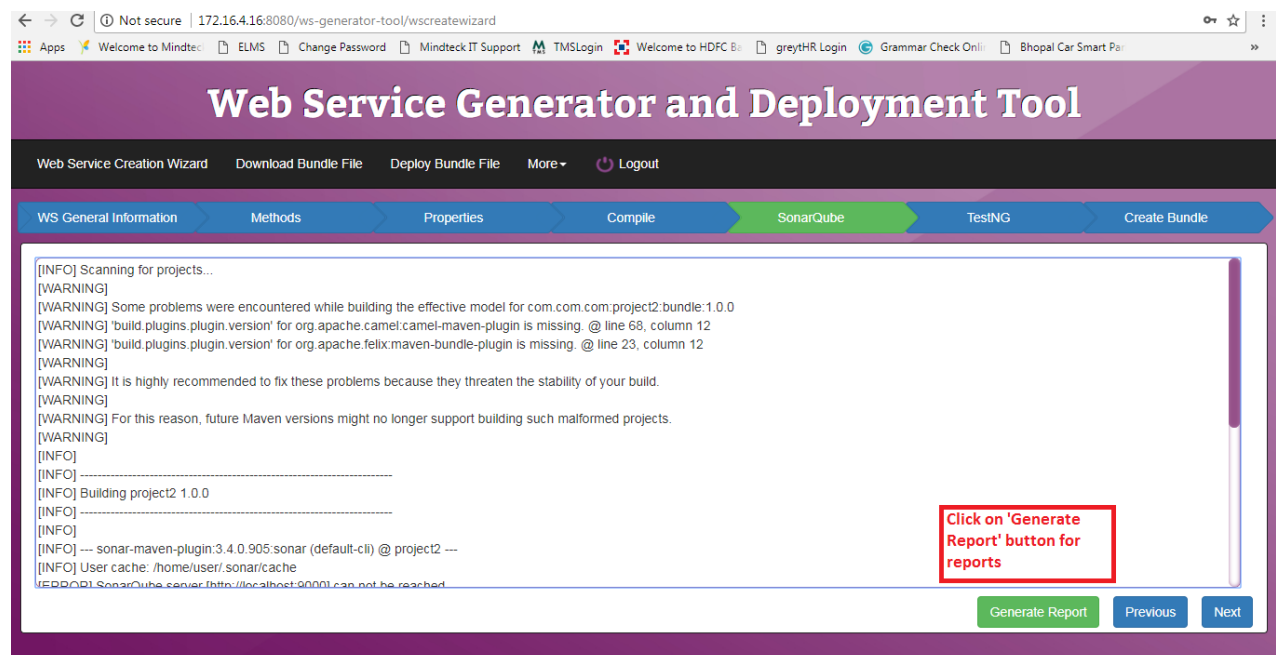
Sl No.	Method Name	WSDL/WSDL URL	Service Name	Version Id
1	method1(id)	testM1id	Method1Service	1
2	method2(id)	TestM2ID	Method2Service	1.1

Previous   Next

#### 2.2.2.7 Compile page

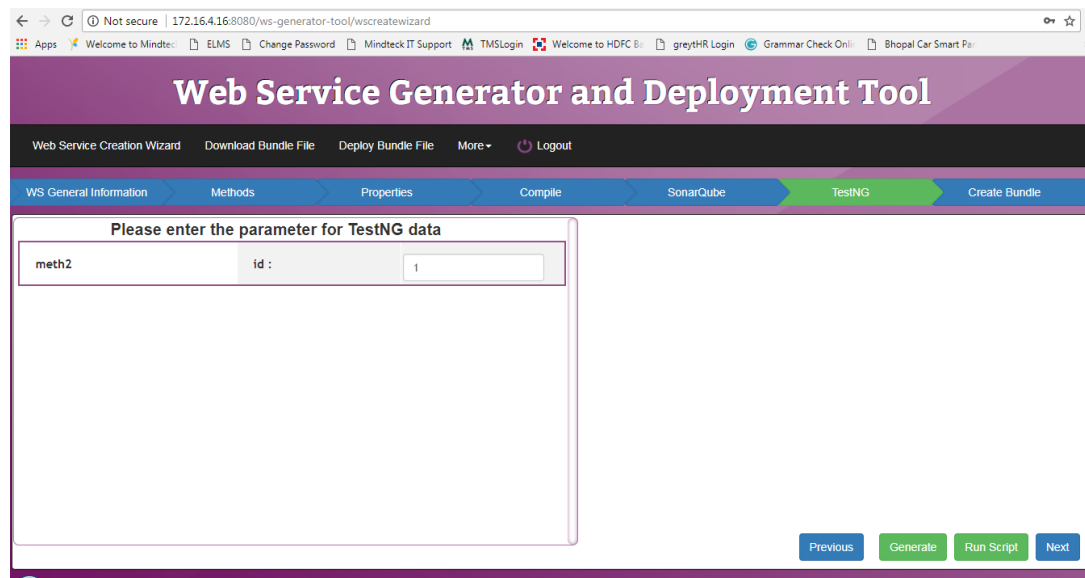


## 2.2.2.8 SonarQube



## 2.2.2.9 TestNG Screen

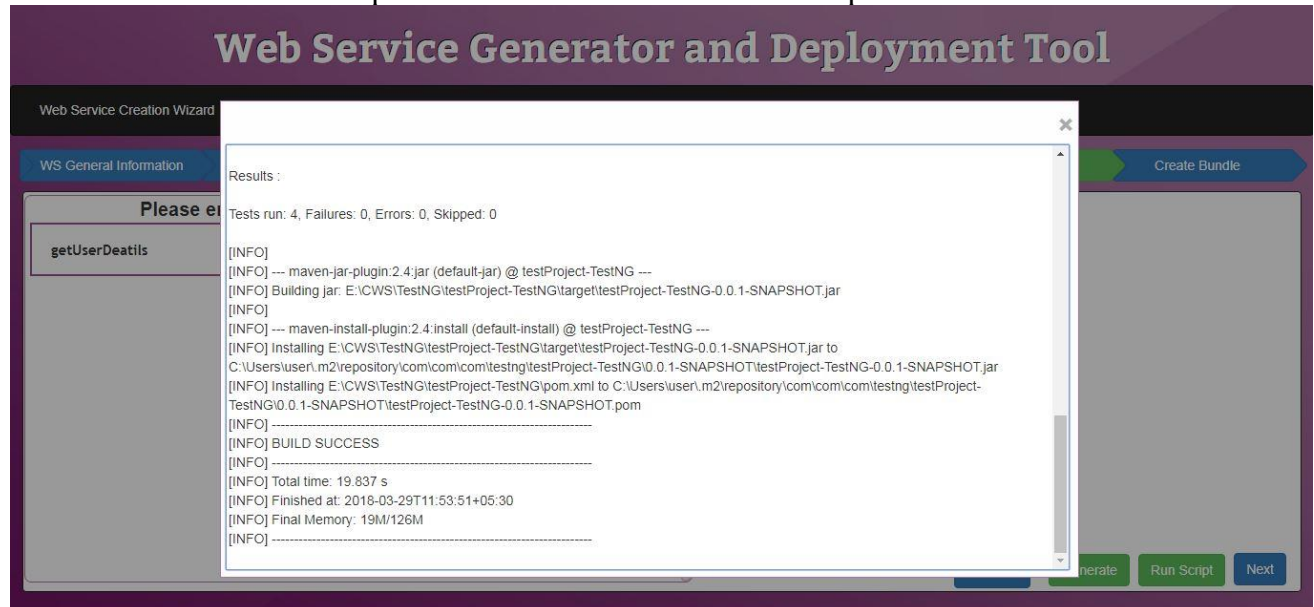
- User can Test the method response using by Generating TestNG scripts
- User can enter method variables and generate script.



The screenshot shows the 'Web Service Generator and Deployment Tool' interface. The 'TestNG' tab is selected in the top navigation bar. Below the navigation bar, there is a section titled 'Please enter the parameter for TestNG data'. This section contains a table with two columns: 'meth2' and 'id :'. The 'meth2' column has the value 'meth2' and the 'id :' column has the value '1'. At the bottom right of the interface, there are four buttons: 'Previous', 'Generate', 'Run Script', and 'Next'.

#### 2.2.2.9.1 TestNG result

- User can run the script and see the result in the TestNG report screen shown below



The screenshot shows the 'Web Service Generator and Deployment Tool' interface with the 'TestNG' tab selected. A modal window titled 'Results :' is displayed, showing the TestNG results. The results indicate that 4 tests were run, with 0 failures, 0 errors, and 0 skipped. The modal window also displays the following information:

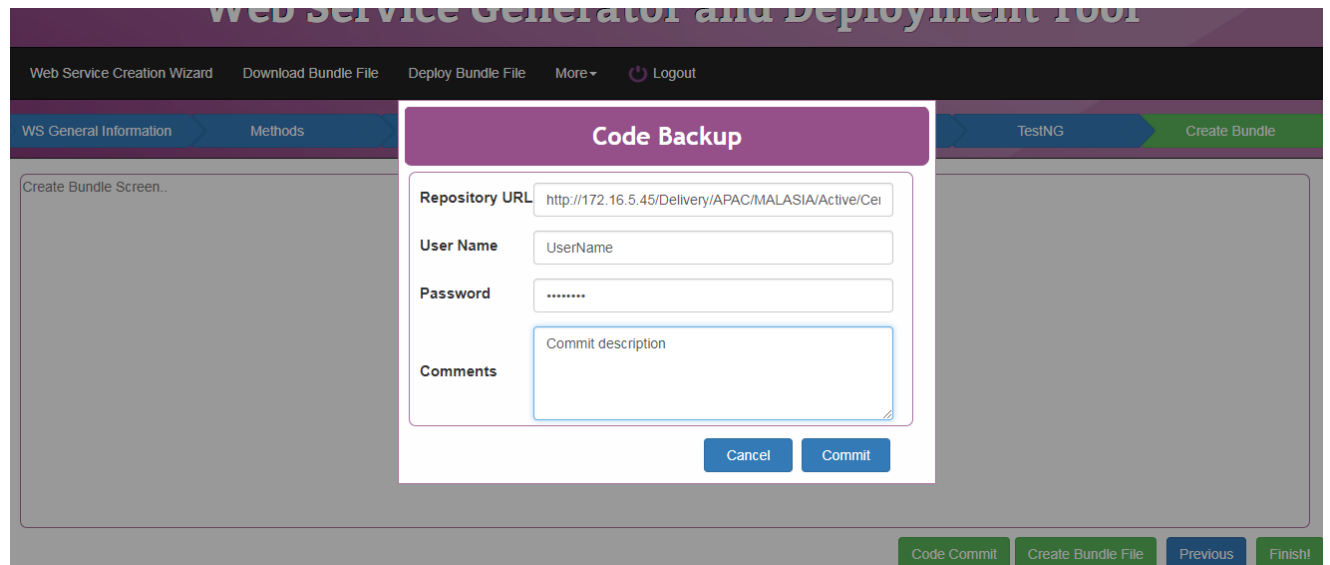
```
Results :
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ testProject-TestNG ---
[INFO] Building jar: E:\CWS\TestNG\testProject-TestNG\target\testProject-TestNG-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ testProject-TestNG ---
[INFO] Installing E:\CWS\TestNG\testProject-TestNG\target\testProject-TestNG-0.0.1-SNAPSHOT.jar to
C:\Users\user\m2\repository\com\com\com\testng\testProject-TestNG\0.0.1-SNAPSHOT\testProject-TestNG-0.0.1-SNAPSHOT.jar
[INFO] Installing E:\CWS\TestNG\testProject-TestNG\pom.xml to C:\Users\user\m2\repository\com\com\com\testng\testProject-TestNG\0.0.1-SNAPSHOT\testProject-TestNG-0.0.1-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 19.837 s
[INFO] Finished at: 2018-03-29T11:53:51+05:30
[INFO] Final Memory: 19M/126M
[INFO] -----
```

#### 2.2.2.10 Commit create code to Repository

- User can do commit to by using code commit feature.
- User will be allowed to enter Repo URL, User name, Password and Description for commit.





Web Service Generator and Deployment Tool

Web Service Creation Wizard Download Bundle File Deploy Bundle File More Logout

WS General Information Methods

Code Backup

Repository URL http://172.16.5.45/Delivery/APAC/MALASIA/Active/Cei

User Name UserName

Password \*\*\*\*\*

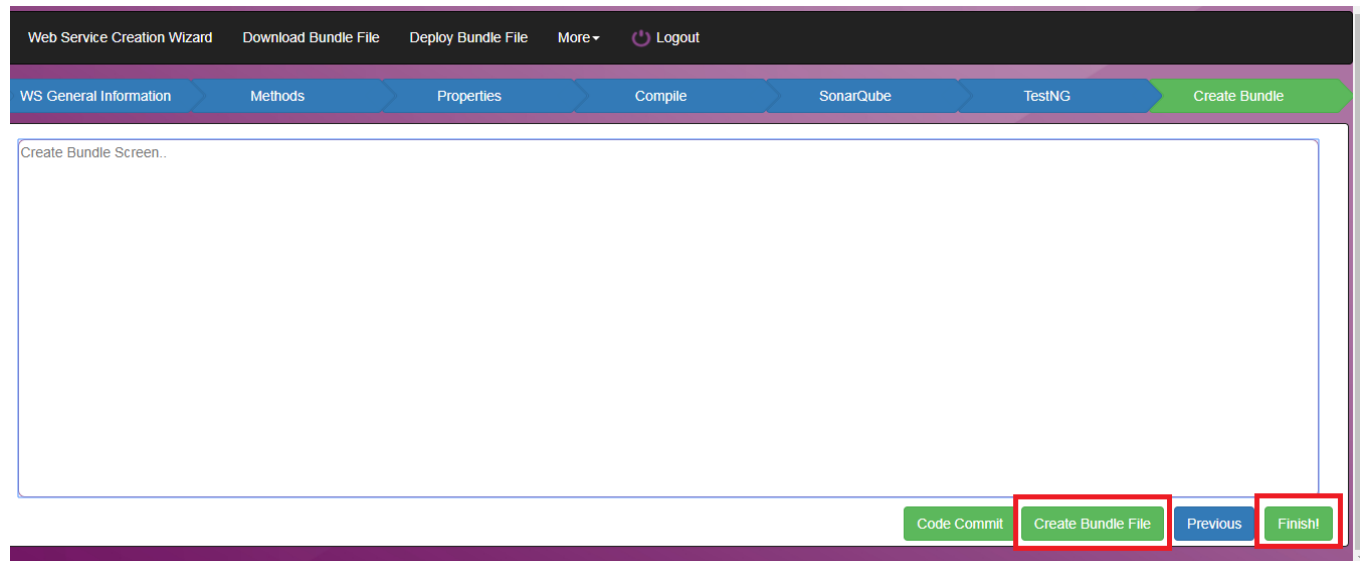
Comments Commit description

Cancel Commit

Code Commit Create Bundle File Previous Finish!

### 2.2.2.11 Create Bundle page

a. Click on 'Create Bundle file' button if the file is success click on 'Finish' button



Web Service Creation Wizard Download Bundle File Deploy Bundle File More Logout

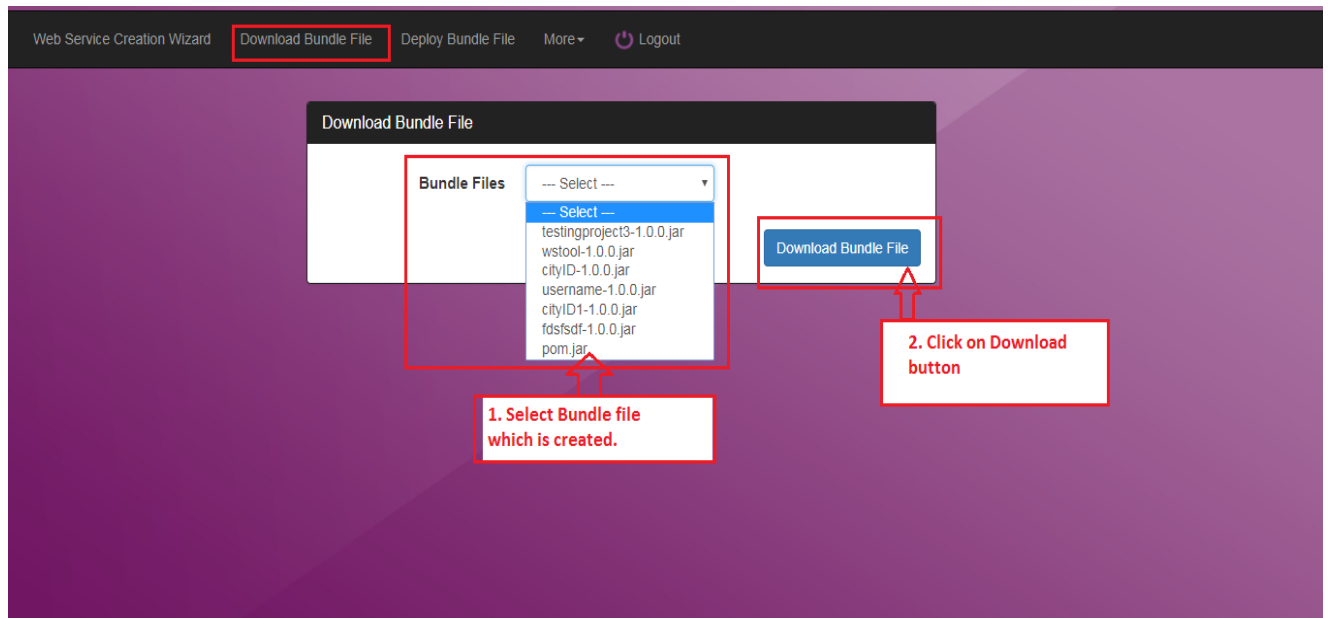
WS General Information Methods Properties Compile SonarQube TestNG Create Bundle

Create Bundle Screen..

Code Commit Create Bundle File Previous Finish!

### 2.2.3 Download Bundle File

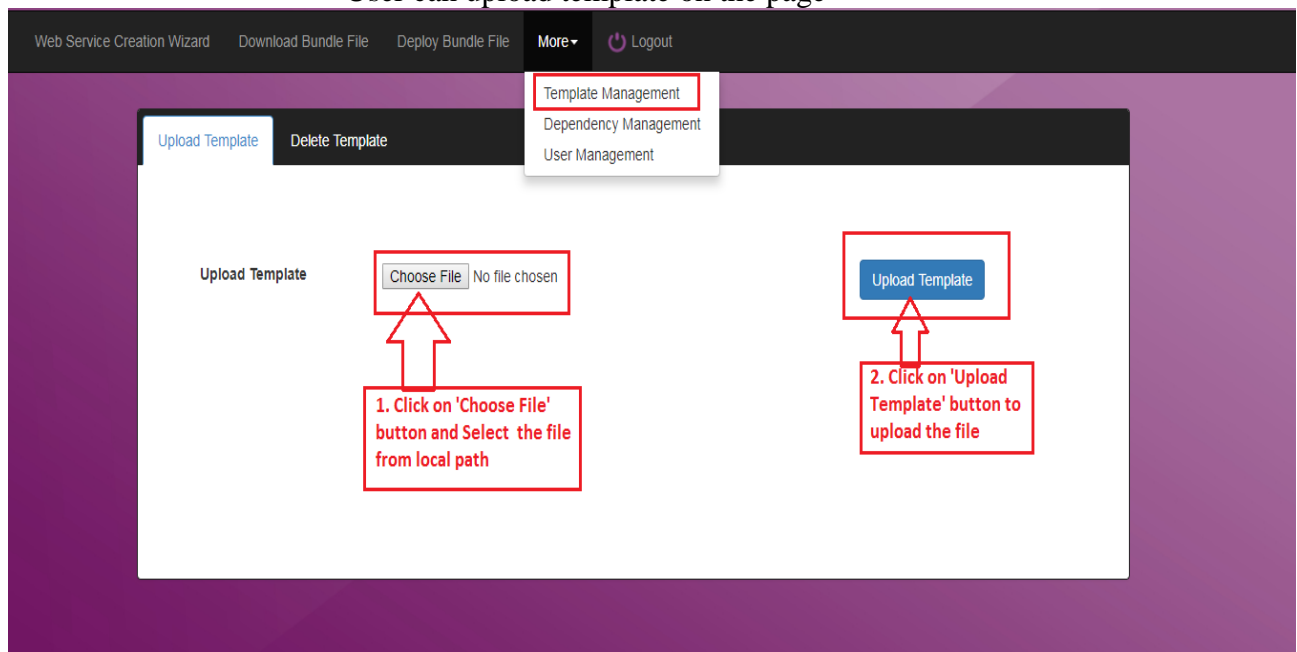
Created bundle file should be displayed on Bundle file Dropdown on 'Download bundle file section as shown below.



## 2.2.4 Template Management

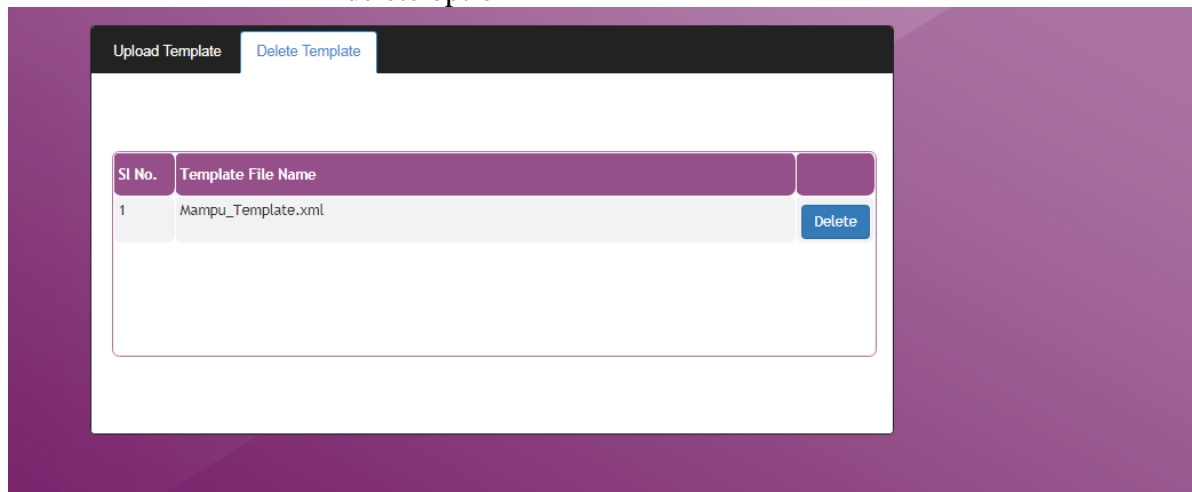
### 2.2.4.1 Upload Template

- User can upload template on the page



### 2.2.4.2 Delete Template

- User can delete uploaded template on the page by clicking on delete option



### 2.2.5 User Management

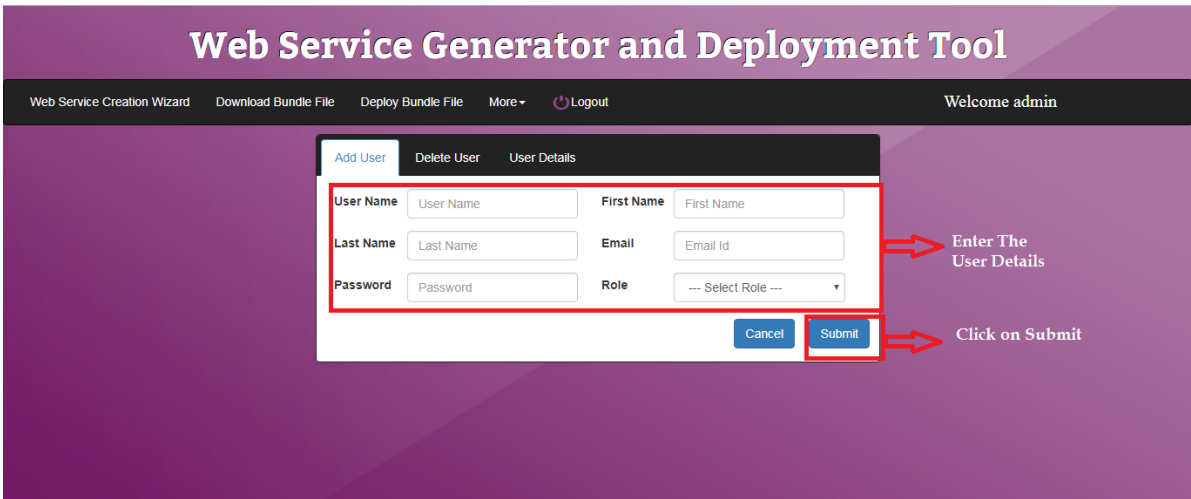
User roles and their responsibilities.

There are two role

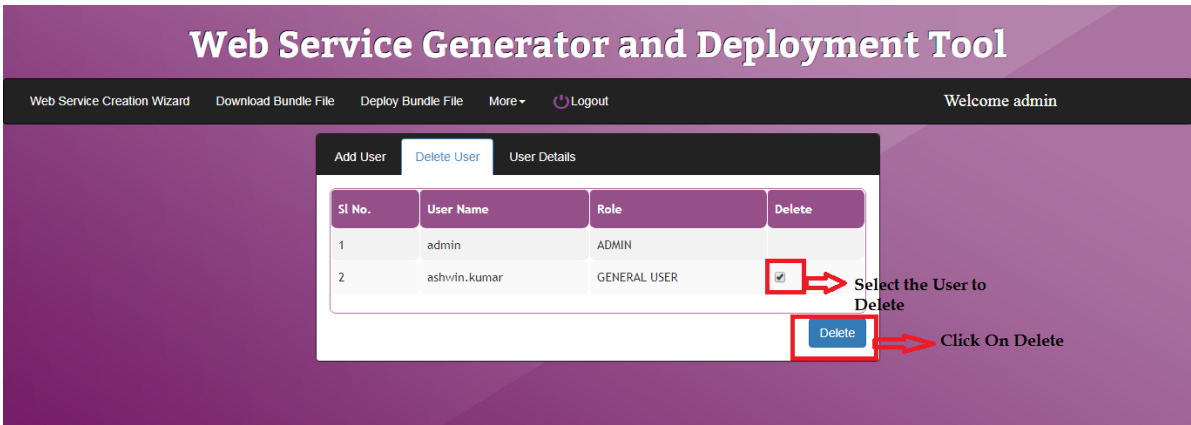
- Admin
  - Admin user can add new user, delete the existing user and can edit the existing user.
- General User
  - General User can only view the users who can use the tool.
  - General User don't have privileges to add, edit and delete the user

#### 2.2.5.1 Add User

User can be added from the user management add user page as shown below

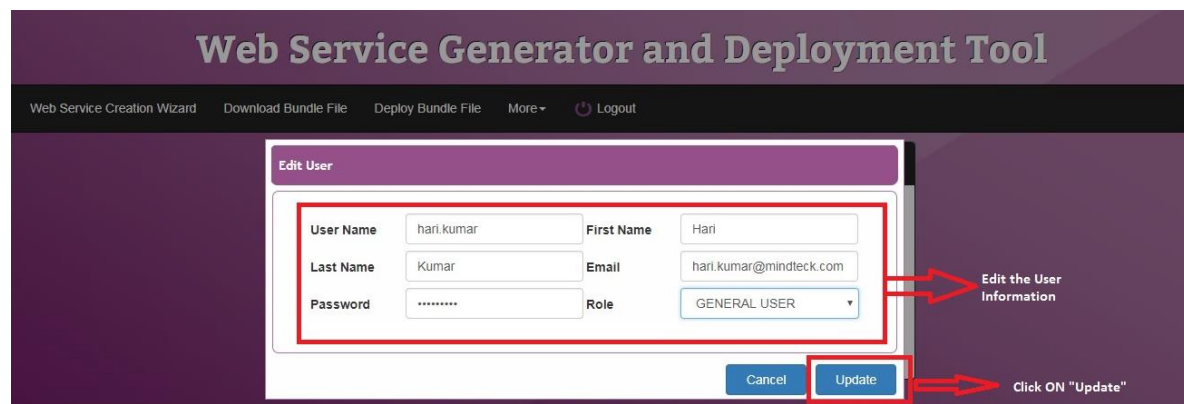
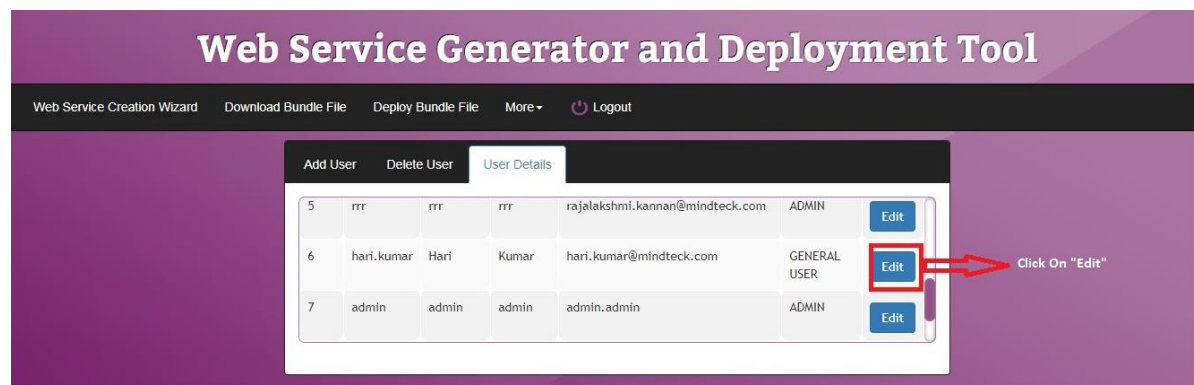


2.2.5.2 Delete User



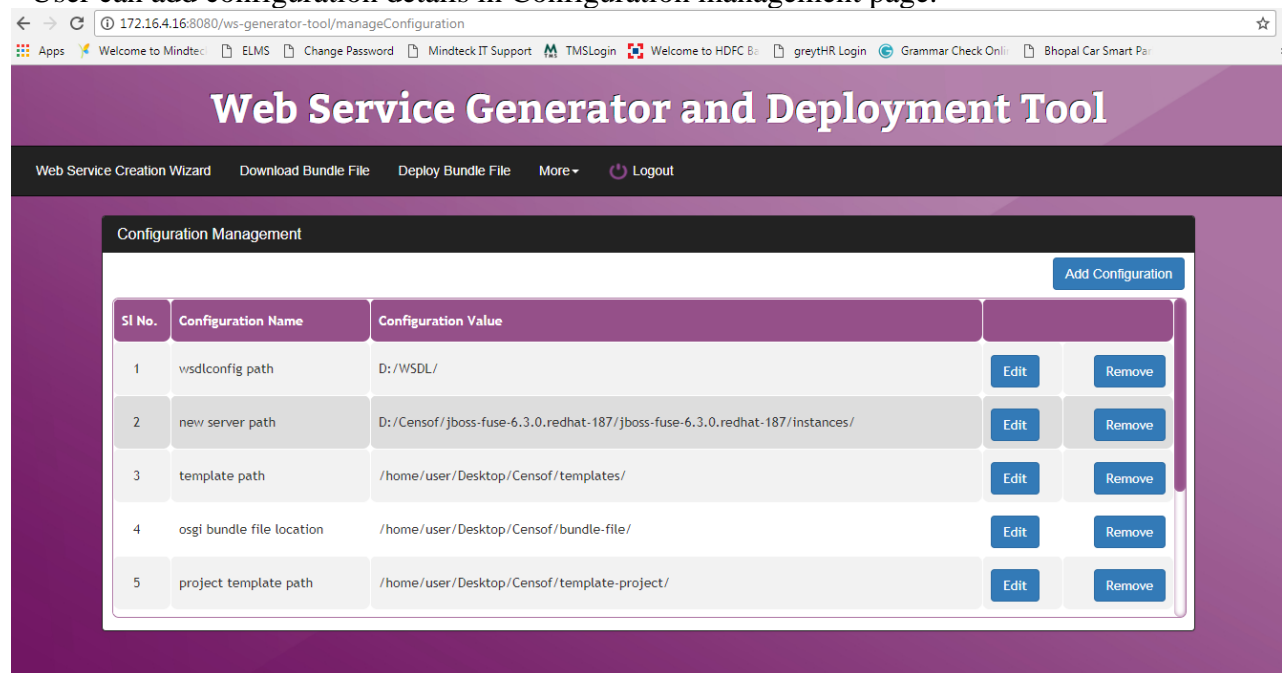
2.2.5.3 Edit User Details

- Admin User can edit the User Information.



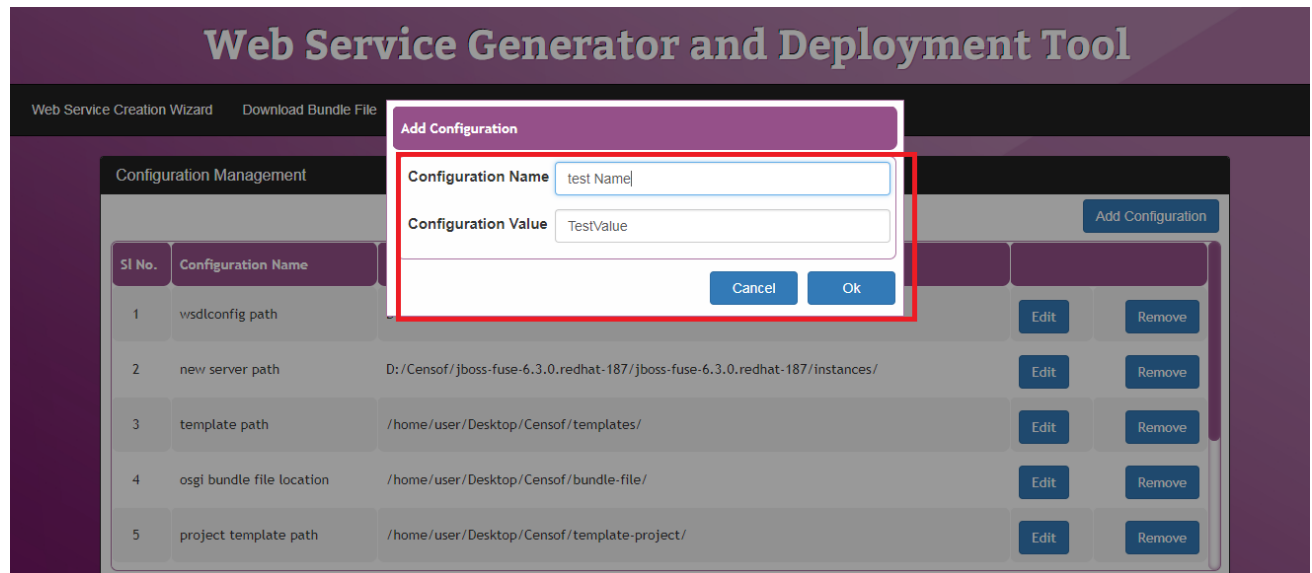
## 2.2.6 Configuration Management:

- User can add configuration details in Configuration management page.



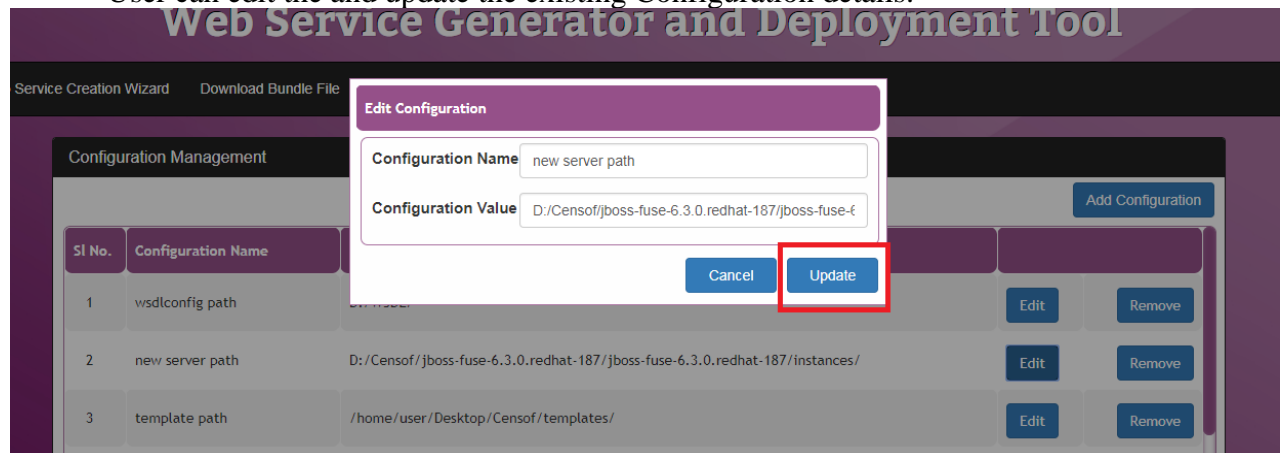
### 2.2.6.2 Add Configuration details

- User can be able to add configuration details in the configuration management.



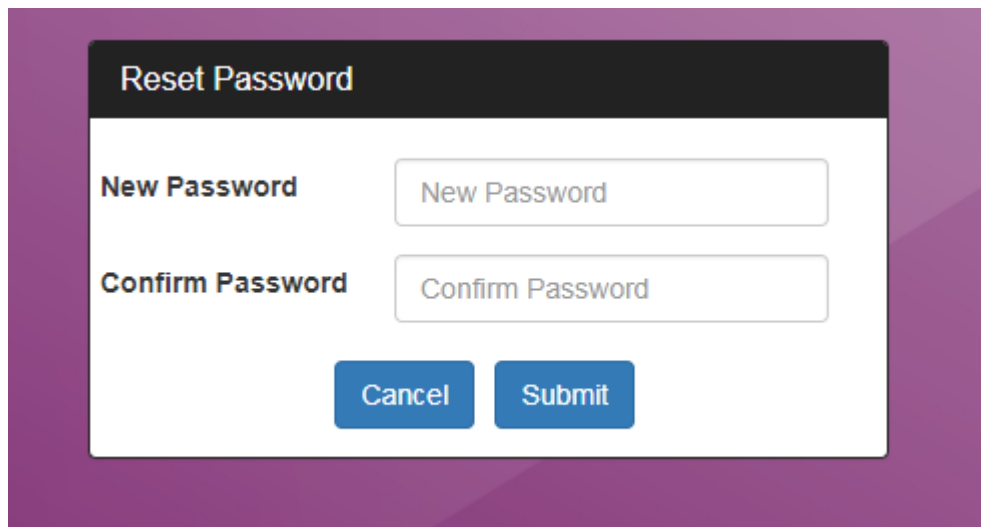
### 2.2.6.3 Update Configuration details:

- User can edit the and update the existing Configuration details.



### 2.2.7 Change Password

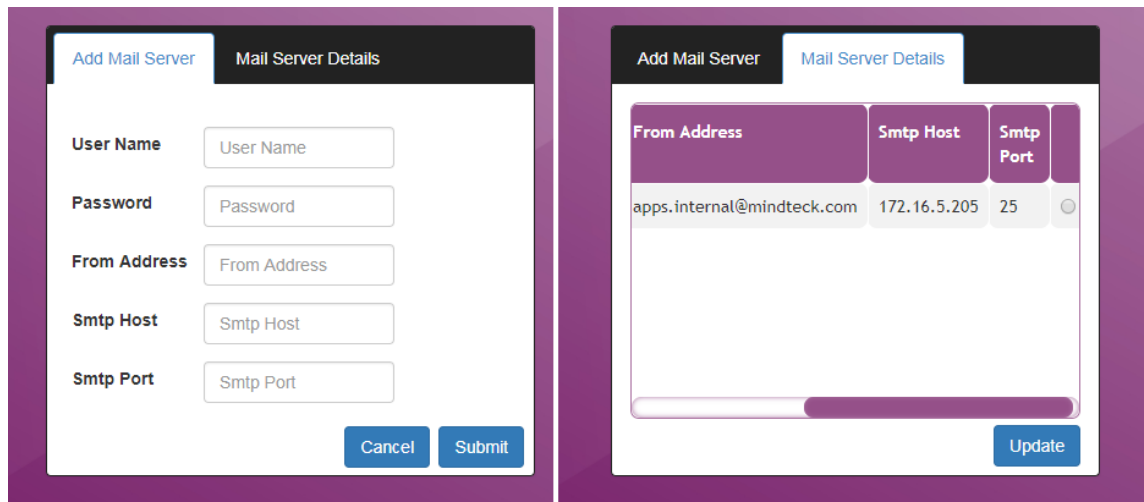
- Users can change the Password using the feature Change Password under More options shown below.



The image shows a 'Reset Password' form. It has a title bar 'Reset Password'. Below it, there are two input fields: 'New Password' and 'Confirm Password'. At the bottom, there are two buttons: 'Cancel' and 'Submit'.

### 2.2.8 Mail Server

- User can add multiple mail servers using the mail server feature and activate which one is required in Mail server Details screen

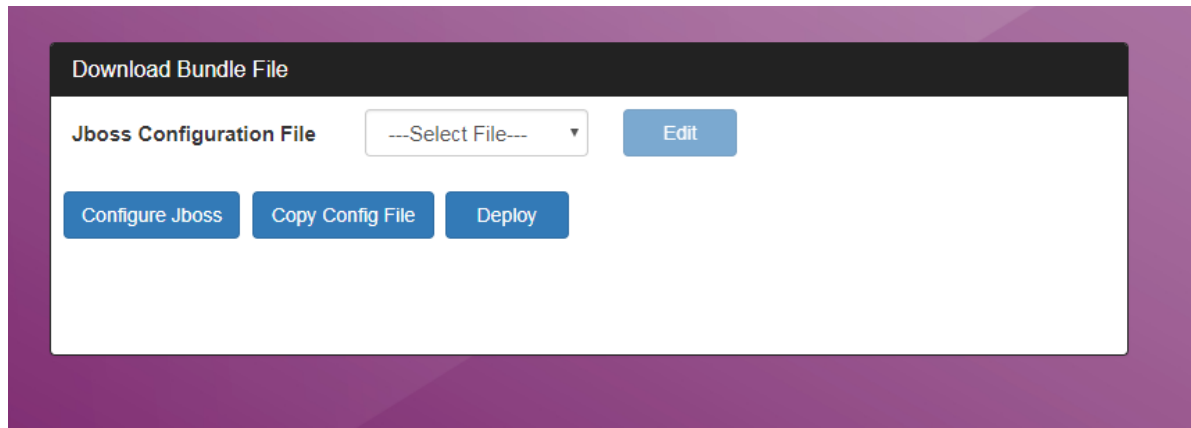


The image shows two screenshots of the Mail Server configuration interface. The left screenshot shows the 'Add Mail Server' screen with fields for 'User Name', 'Password', 'From Address', 'Smtp Host', and 'Smtp Port'. The right screenshot shows the 'Mail Server Details' screen with a table of mail servers and an 'Update' button.

From Address	Smtp Host	Smtp Port	
apps.internal@mindteck.com	172.16.5.205	25	<input type="radio"/>

### 2.2.9 Deploy Bundle file

- User can deploy the created bundle file using following screen
- User can edit the Jboss Configuration file by clicking on edit button
- By Clicking on Configure Jboss, the Server will get configured in backend.



- JBoss Fuse Configuration file



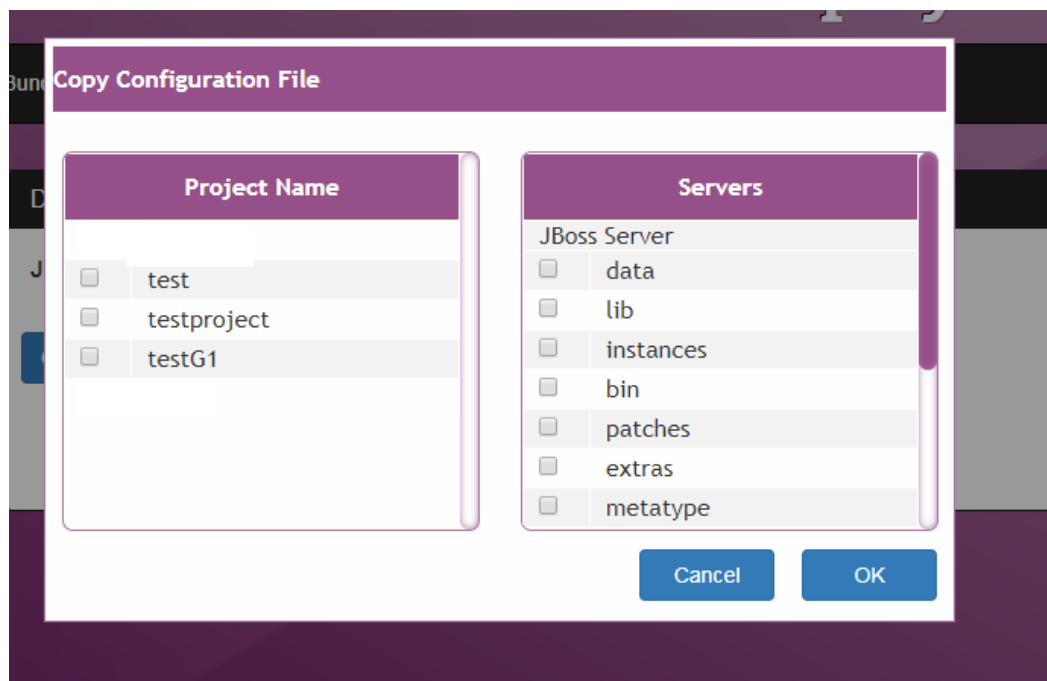
- Fuse Commands
  - fabric create = fabric:create --new-user <<User Name>> --new-user-password <<Password>>
  - container create = fabric:container-create-child root <<Instance Name>>
  - container list = fabric:container-list



- fabric profile = fabric:profile-create --parent jboss-fuse-minimal <<Profile Name>>
- fabric container = fabric:container-add-profile <<Instance Name>> <<Profile Name>>
- profile edit = profile-edit --feature camel-bean-validator mindteck
- Add repositories = fabric:profile-edit --repositories mvn:com.test.ws/testdeployment/1.0.0/xml/features mindteck
- profile feature = profile-edit --feature testdeployment/1.0.0 mindteck
- profile feature = profile-edit --feature testdeployment-dependencies/1.0.0 mindteck

### 2.2.9.1 Copy Configuration Files

- After clicking on Copy Config File button user can see the project files and Server folders
- User can Select particulate file and copy in to the server by clicking on Copy button.



### 2.2.9.2 Deploy Bundle file

- After clicking on Deploy button Deployment process will happened in Backend and URL will be displayed on the screen

