

Web Services Generator Tool – User Manual		TEM 010
Prepared By	Goverdhan/Ashwinkumar M	Documentation Type : User Manual
Checked By	Saurabh Bagchi	Document Ref : SDLC_TEM_User Manual
Approved By		Version No.: 2.0
Date of Release: 11-10-2018		No. Of Pages (Including cover page): 33

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	1.0	21-09-2018	Ashwinkumar M	Censof
Phase -II Final	All	2.0	11-10-2018	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.

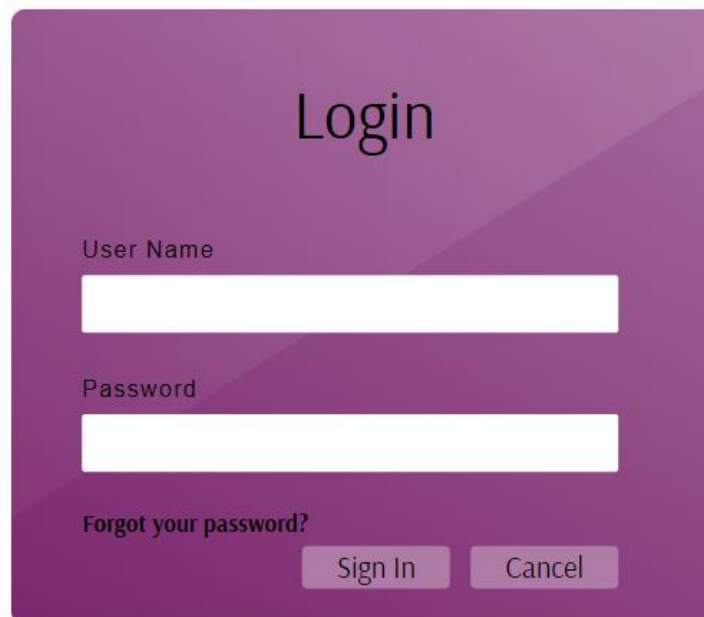
1.3 Users of this Manual

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 mockup of a login screen with a purple gradient background. The word "Login" is centered at the top in a large, white, sans-serif font. Below it, the text "User Name" is followed by a white rectangular input field. Further down, the text "Password" is followed by another white rectangular input field. At the bottom left, the text "Forgot your password?" is displayed. At the bottom right, there are two white buttons with purple text: "Sign In" and "Cancel".

Login

User Name

Password

Forgot your password?

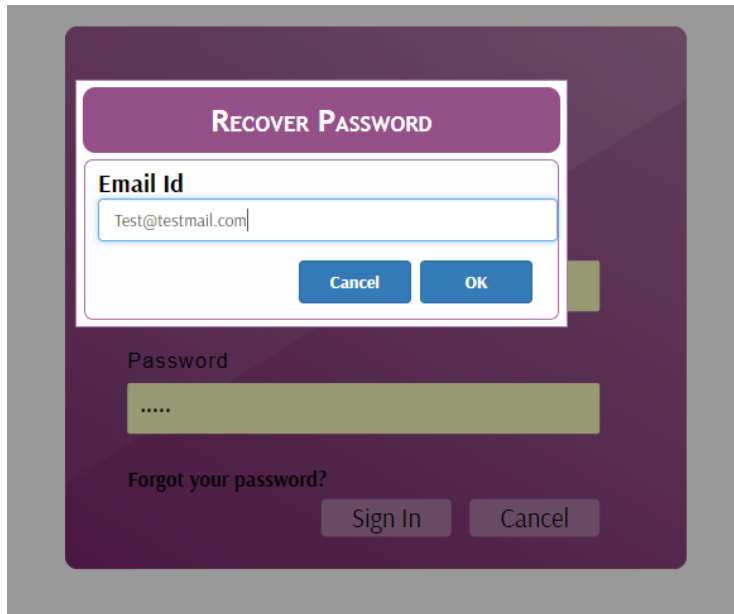
Sign In Cancel

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.

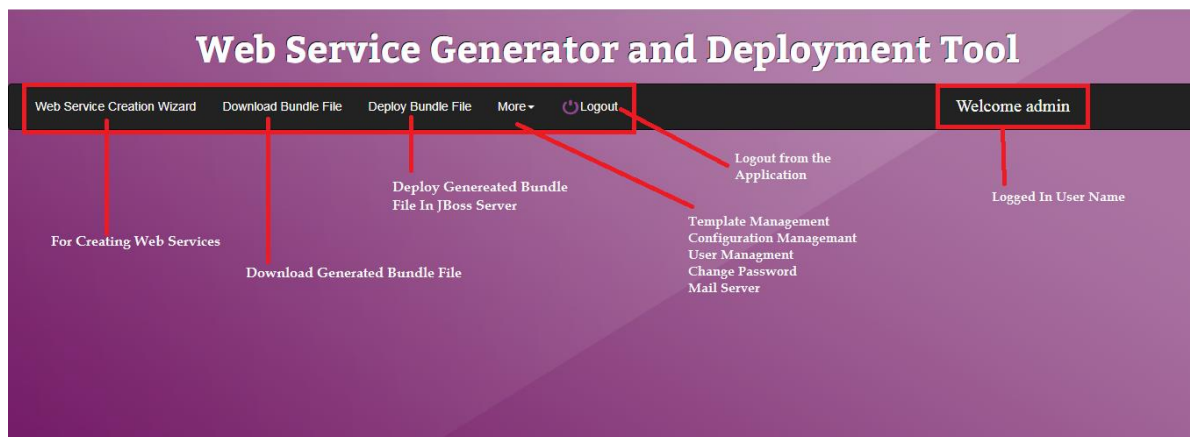
Steps to Rest Password.

1. Click on forget Password.
2. Enter the valid Email Id
3. Click ok
4. Mail will be generated and sent to Email Id
5. User The generated mail password 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)

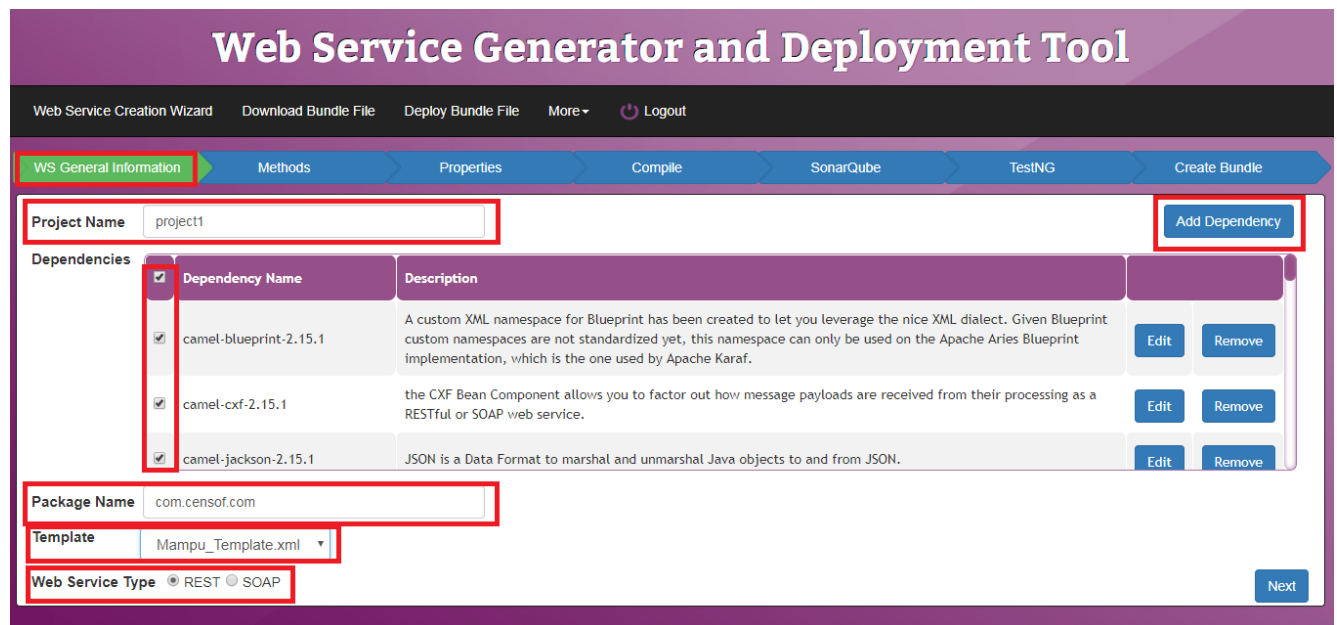
2.2.2 Web Service Creation Wizard

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.
All below the fields are mandatory.

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

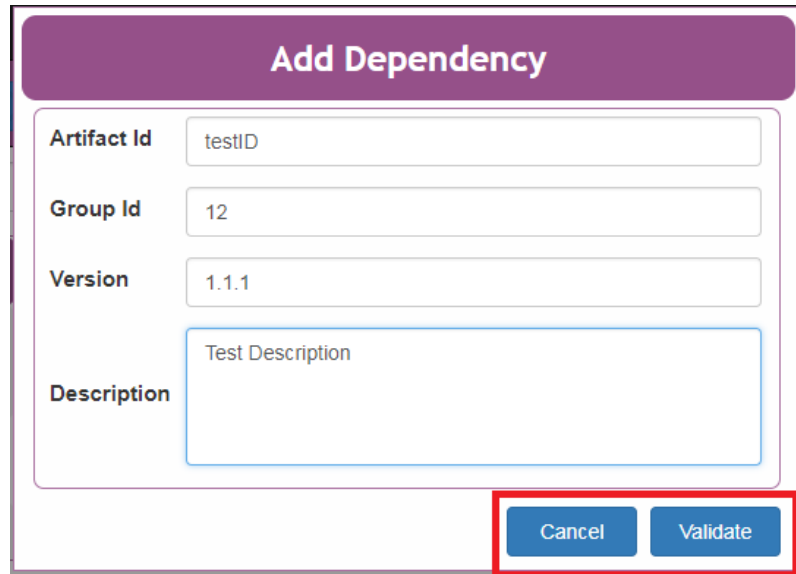


The screenshot displays the 'Web Service Generator and Deployment Tool' interface. The 'WS General Information' tab is active, showing a form with the following fields and options:

- Project Name:** A text input field containing 'project1'.
- Add Dependency:** A button located to the right of the Project Name field.
- Dependencies:** A table with three columns: 'Dependency Name', 'Description', and 'Actions' (containing 'Edit' and 'Remove' buttons). The table lists three dependencies, all of which are checked:

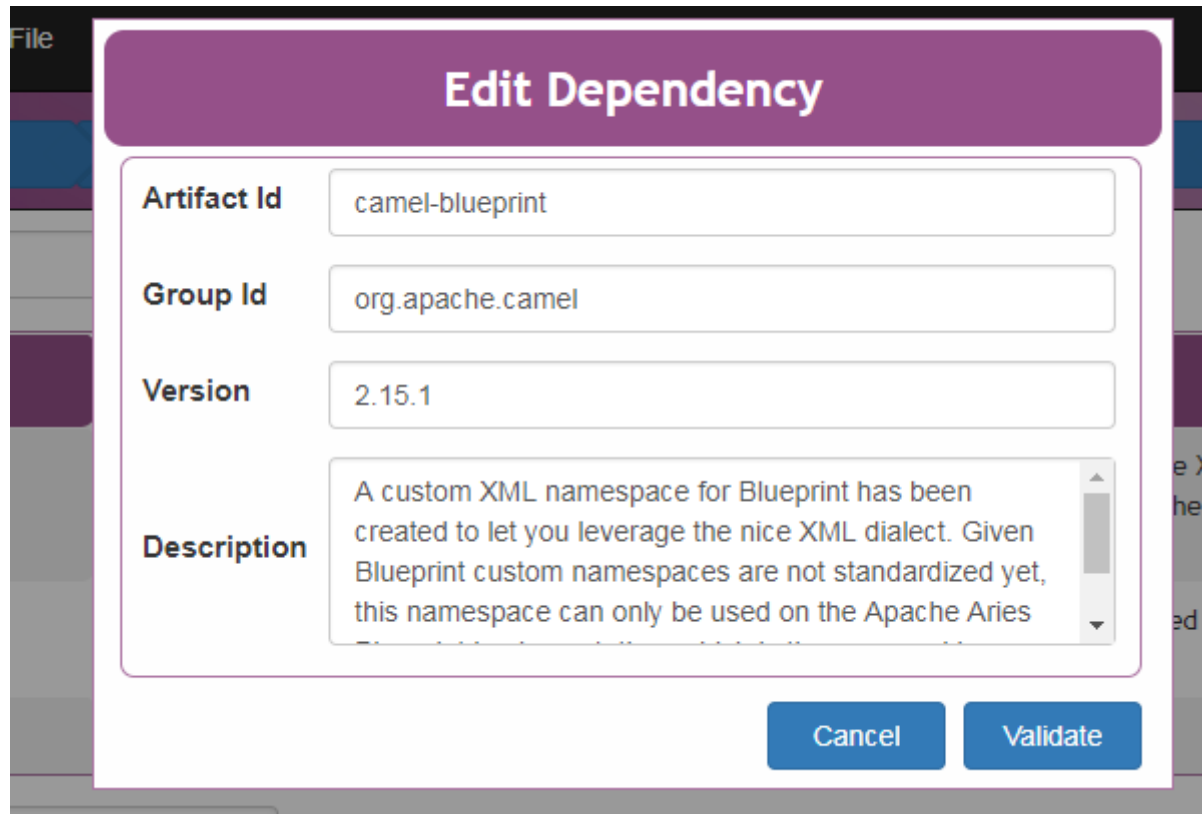
Dependency Name	Description	Actions
<input checked="" type="checkbox"/> camel-blueprint-2.15.1	A custom XML namespace for Blueprint has been created to let you leverage the nice XML dialect. Given Blueprint custom namespaces are not standardized yet, this namespace can only be used on the Apache Aries 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:** A text input field containing 'com.censof.com'.
- Template:** A dropdown menu showing 'Mampu_Template.xml'.
- Web Service Type:** Radio buttons for 'REST' (selected) and 'SOAP'.
- Next:** A button located at the bottom right of the form.

2.2.2.1.1 Add Dependency



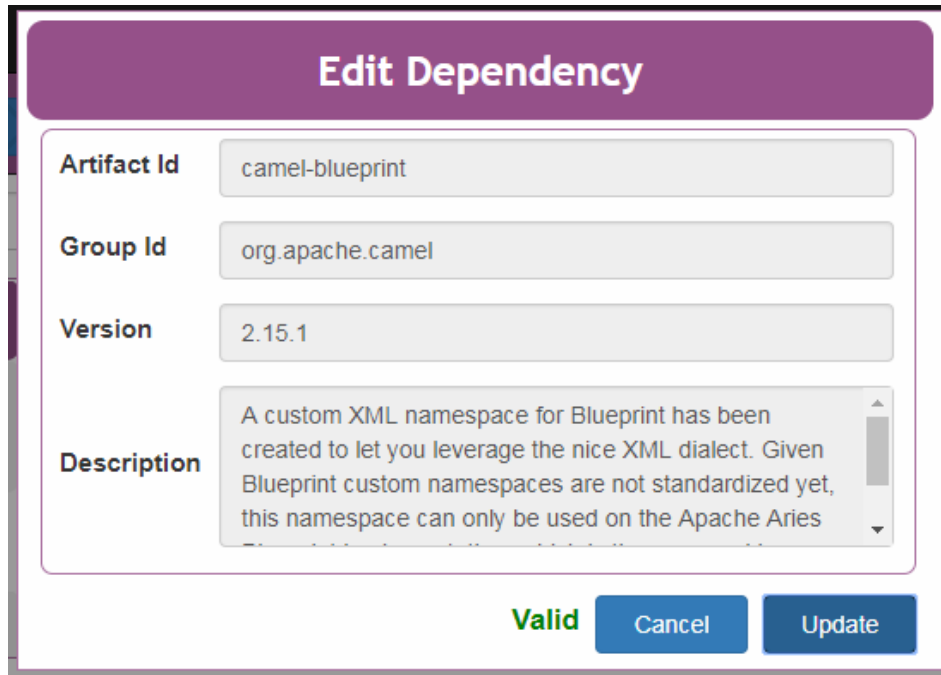
The 'Add Dependency' dialog box features a purple header with the title 'Add Dependency'. Below the header, there are four input fields: 'Artifact Id' with the value 'testID', 'Group Id' with the value '12', 'Version' with the value '1.1.1', and a 'Description' text area containing 'Test Description'. At the bottom right, there are two 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' text area contains a multi-line 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 buttons: 'Cancel' and 'Validate'.

2.2.2.1.3 Update Dependency



Edit Dependency

Artifact Id camel-blueprint

Group Id org.apache.camel

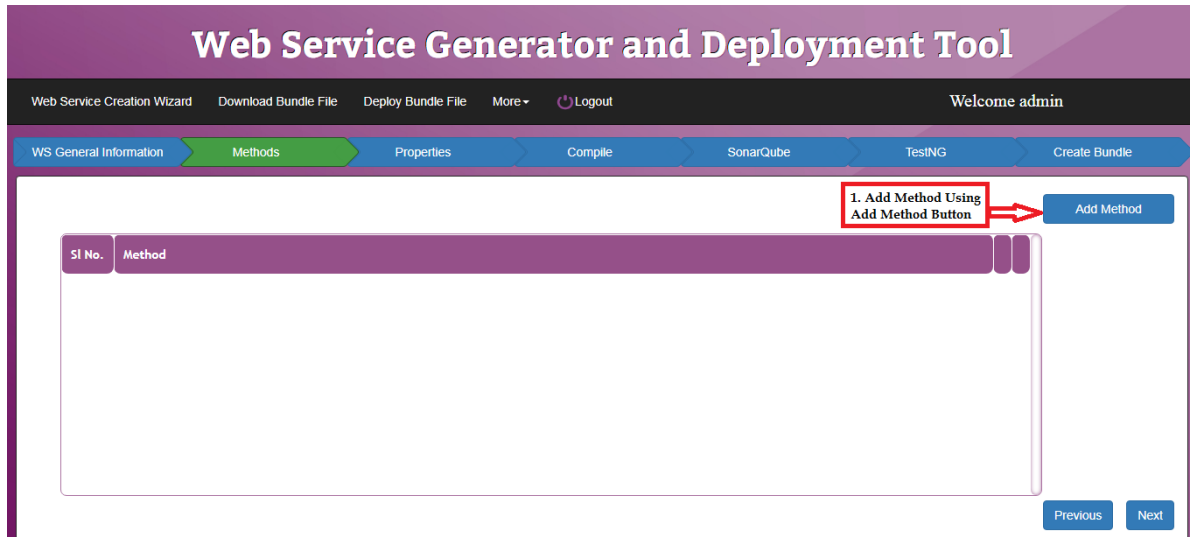
Version 2.15.1

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

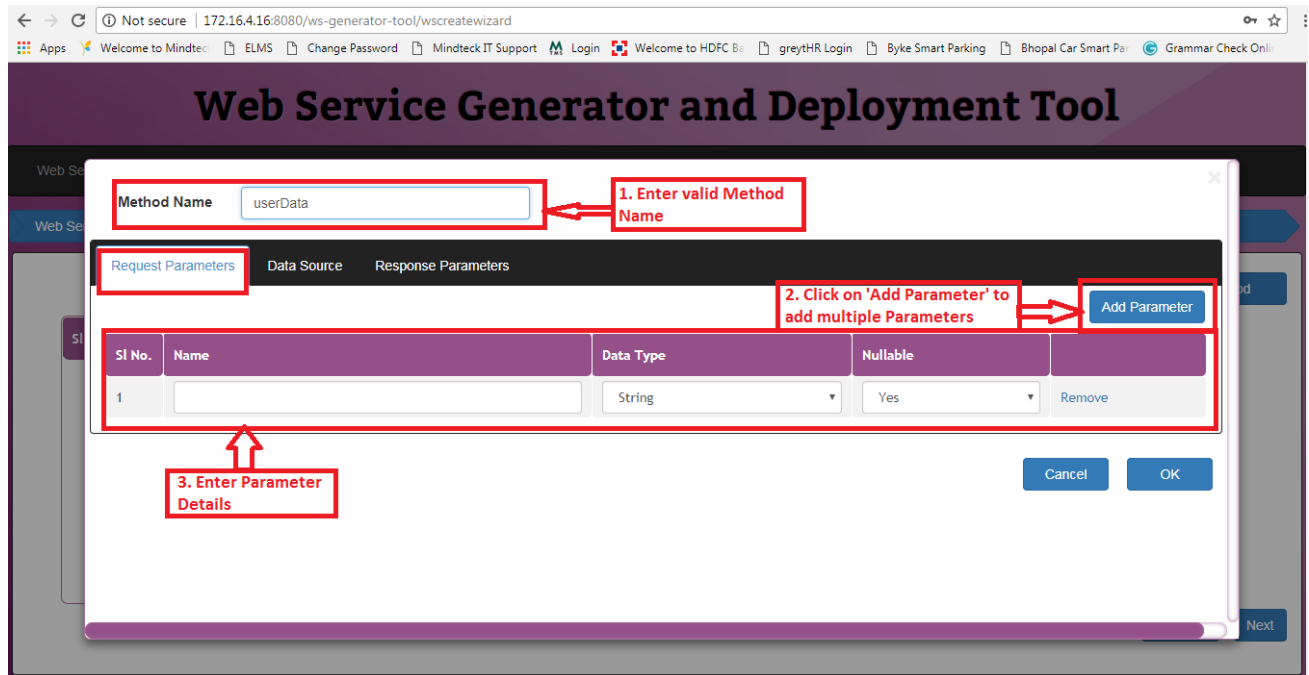
WS General Information **Methods** Properties Compile SonarQube TESTING Create Bundle

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

Request Parameters | Data Source | Response Parameters

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

Add Parameter

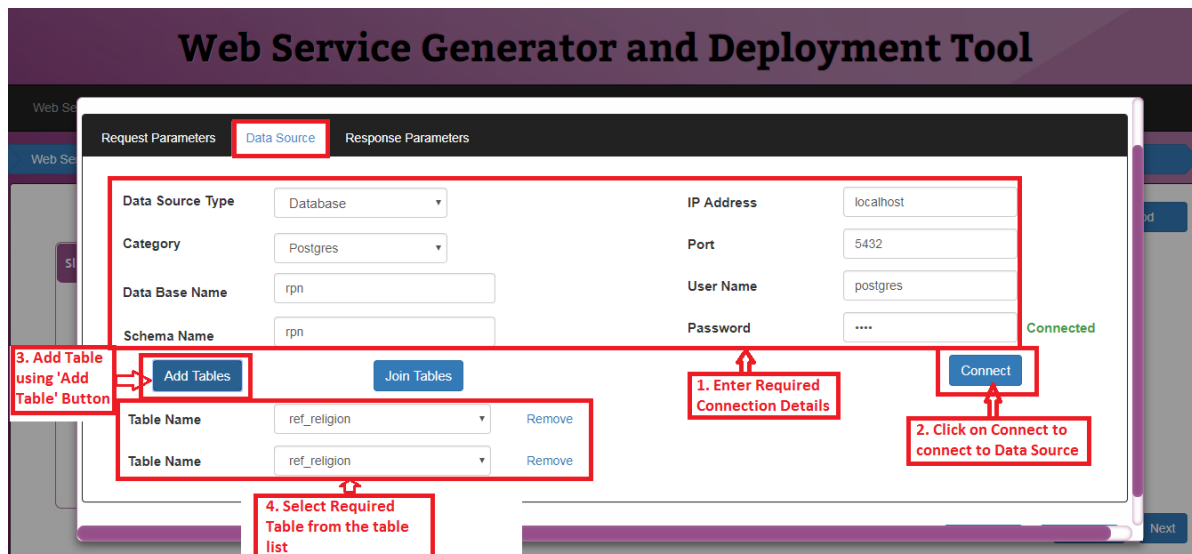
Sl No.	Name	Data Type	Nullable	
1	<input type="text"/>	String	Yes	Remove

3. Enter Parameter Details

Cancel OK

Next

2.2.2.3.2 Add Data Source



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

Add Tables | Join Tables

1. Enter Required Connection Details

2. Click on Connect to connect to Data Source

Connect

4. Select Required Table from the table list

Table Name: ref_religion | Remove

Table Name: ref_religion | Remove

Next

1. Postgres:

Request Parameters

Data Source

Response Parameters

Data Source Type

Database

Category

Postgres

Data Base Name

wstool

Schema Name

public

Add Tables

Join Tables

Table Name

ws_user

Remove

Table Name

project_details

Remove

IP Address

localhost

Port

5432

User Name

root

Password

....

Connected

Connect

2. 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

3. MSSQL:

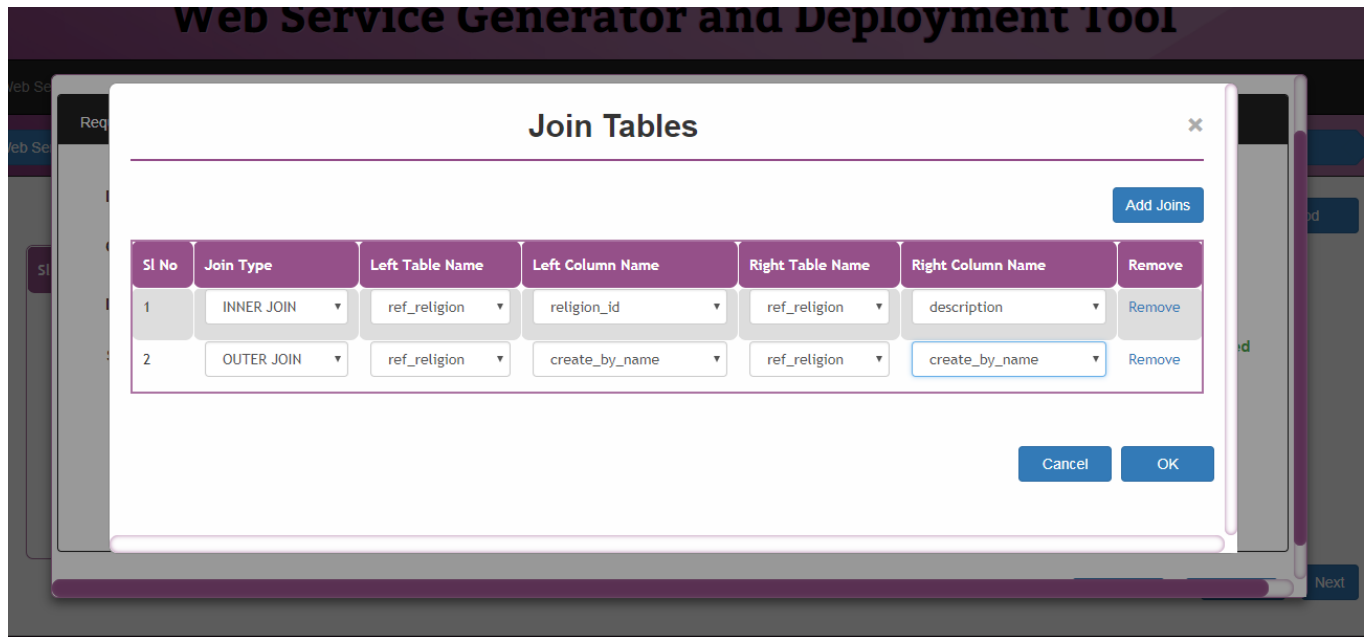
Request Parameters	Data Source	Response Parameters	
Data Source Type	Database ▼	IP Address	localhost
Category	MSSQL ▼	User Name	sa
Data Base Name	TestDB	Password Connected
Add Tables	Join Tables	Connect	
Table Name	Persons ▼ Remove		
Table Name	Inventory ▼ Remove		

4. Oracle:

Request Parameters	Data Source	Response Parameters	
Data Source Type	Database ▼	IP Address	localhost
Category	Oracle ▼	Port	1630
Service Name	TestDB	User Name	SA
Add Tables	Join Tables	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

Tables With Common Column :- ws_user, role, project_details

Step1 – Joining the tables using common column/field.

1st JOIN -Inner join between the tables ws_user and project_details which have the common column user_id.

2nd 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

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 Joins

Cancel

OK

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

☒

ws_user

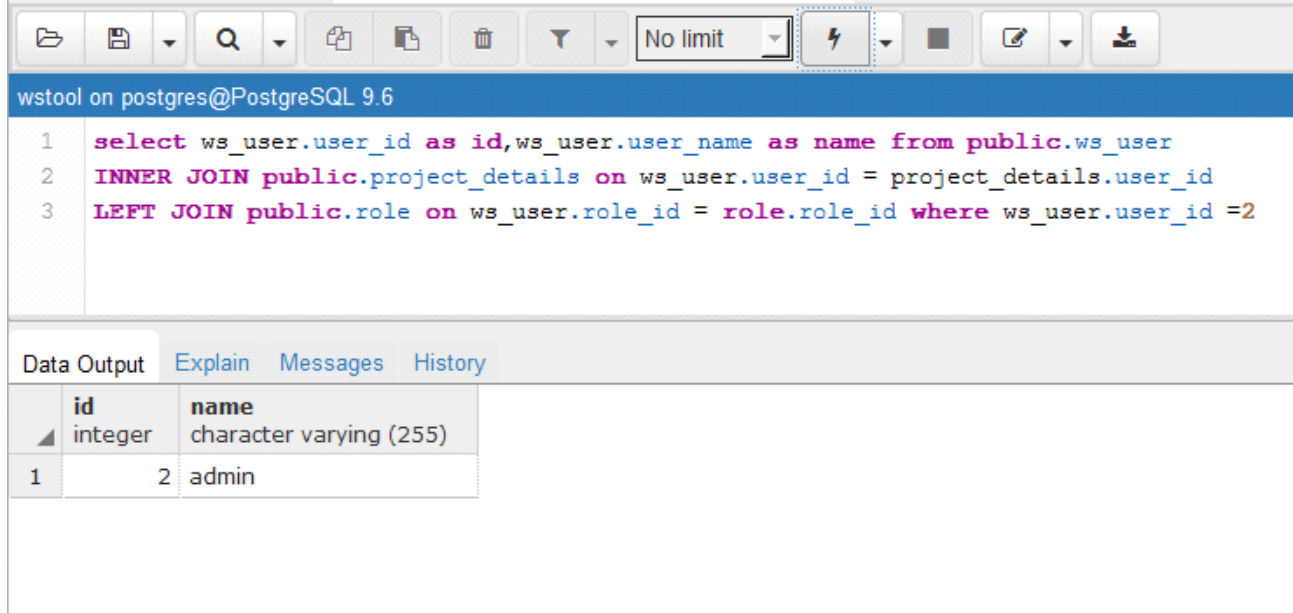
user_name

☒

=[id]

View Query

Step3 – Executing query in postgres.



wstool on postgres@PostgreSQL 9.6

```

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

```

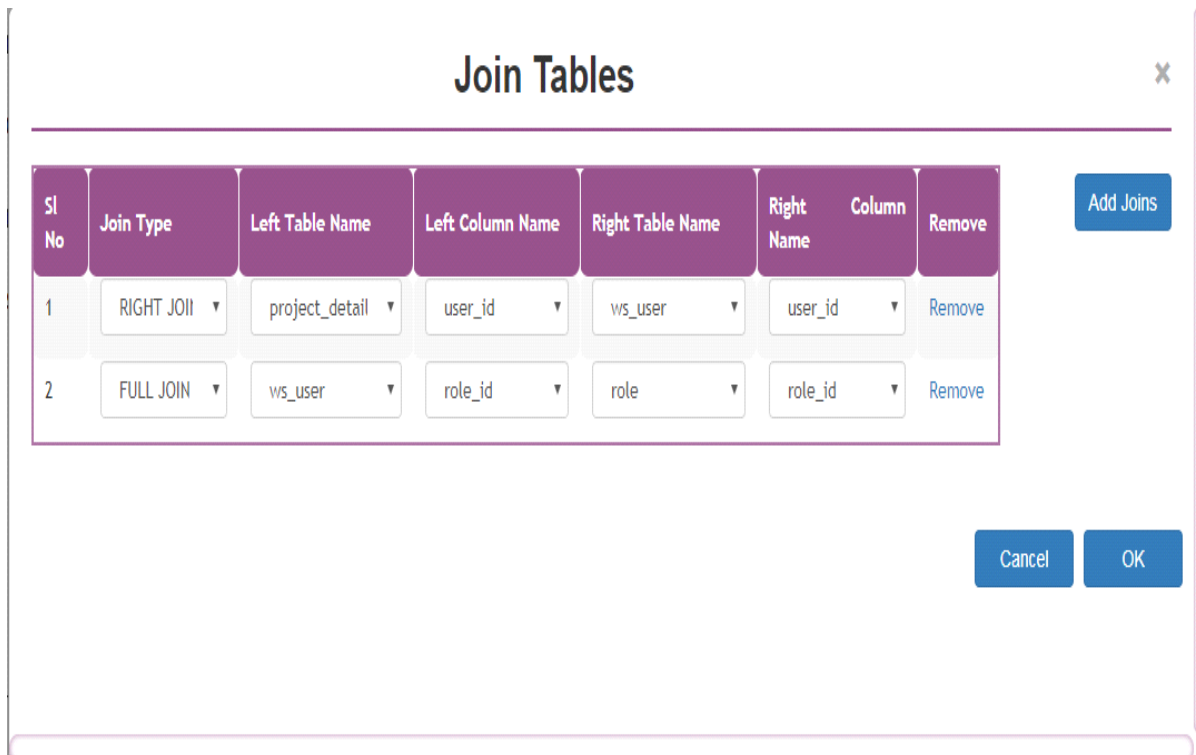
Data Output Explain Messages History

	id	name
	integer	character varying (255)
1	2	admin

Example2 :-

1st Join – Right join between the tables project_details and ws_user

2nd 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 Joins

Cancel OK

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.

No limit

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

Example 3.

1st 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

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.

wstool 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.

1st Join – Left join between the tables project_details and ws_user.

2nd Join- Right join between the tables role and ws_user.

Join Tables

1

LEFT JOIN

project_detail

user_id

ws_user

user_id

Remove

2

RIGHT JOIN

ws_user

role_id

role

role_id

Remove

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]
```

X

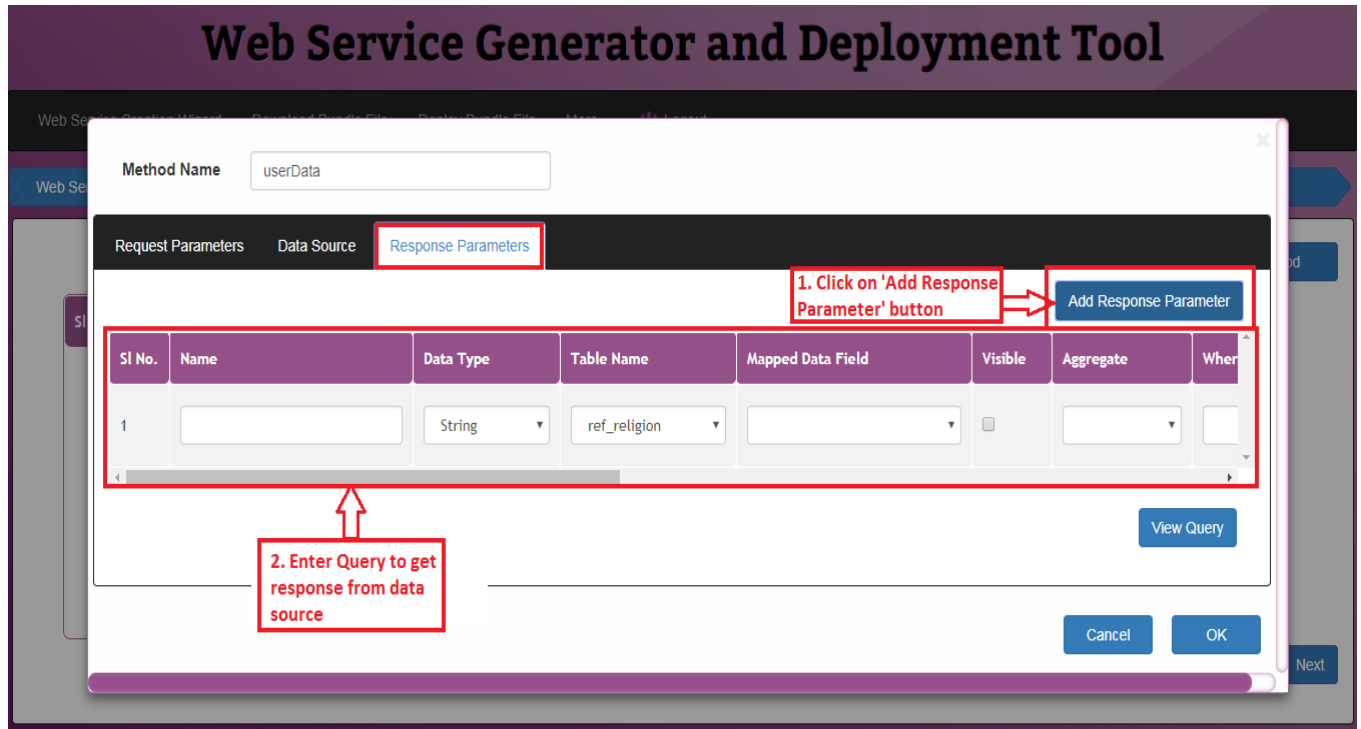
Result for the above.

```
wstool on postgres@PostgreSQL 9.6

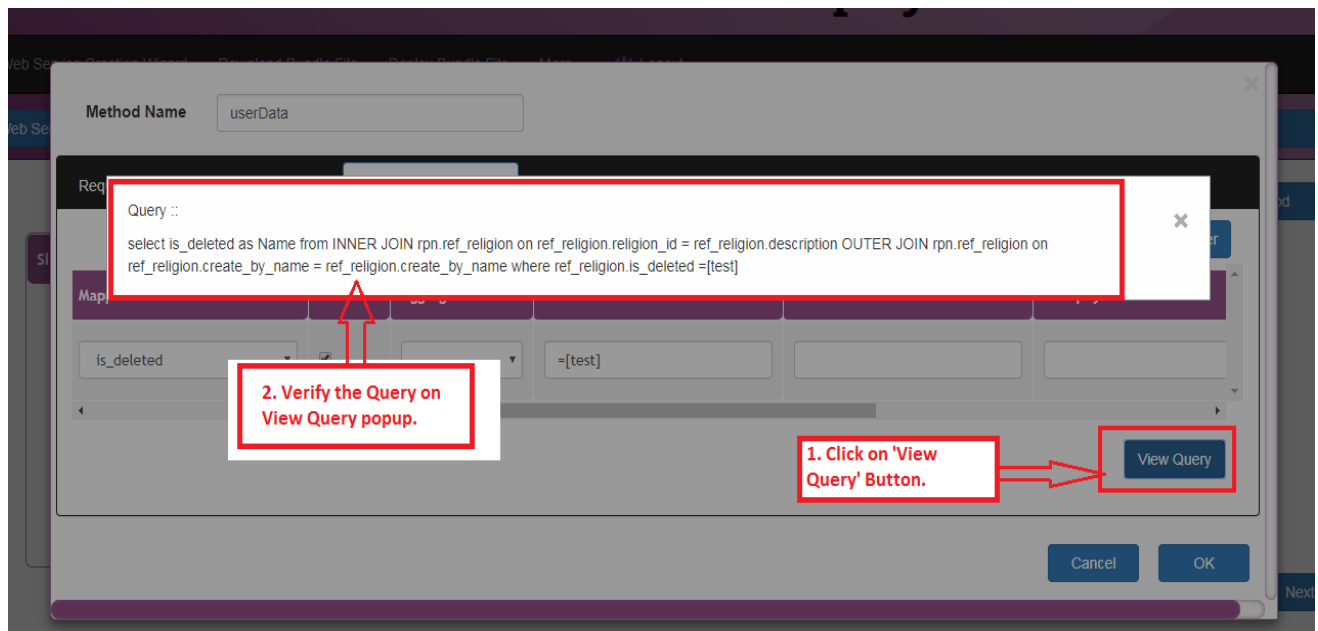
1 select project_details.project_id as ProjId,ws_user.is_active as UserActive,role.role_name as RoleName from public.project_details
2 LEFT JOIN public.ws_user on project_details.user_id = ws_user.user_id
3 RIGHT JOIN public.role on ws_user.role_id = role.role_id where project_details.project_id =824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

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

2.2.2.3.4 Add Response Parameter



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.

Web Service Generator and Deployment Tool

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

Web Service General Information **Methods** Properties Compile Create Bundle

Sl No.	Method		
1	userData(test)	Edit	Remove

[Add Method](#)

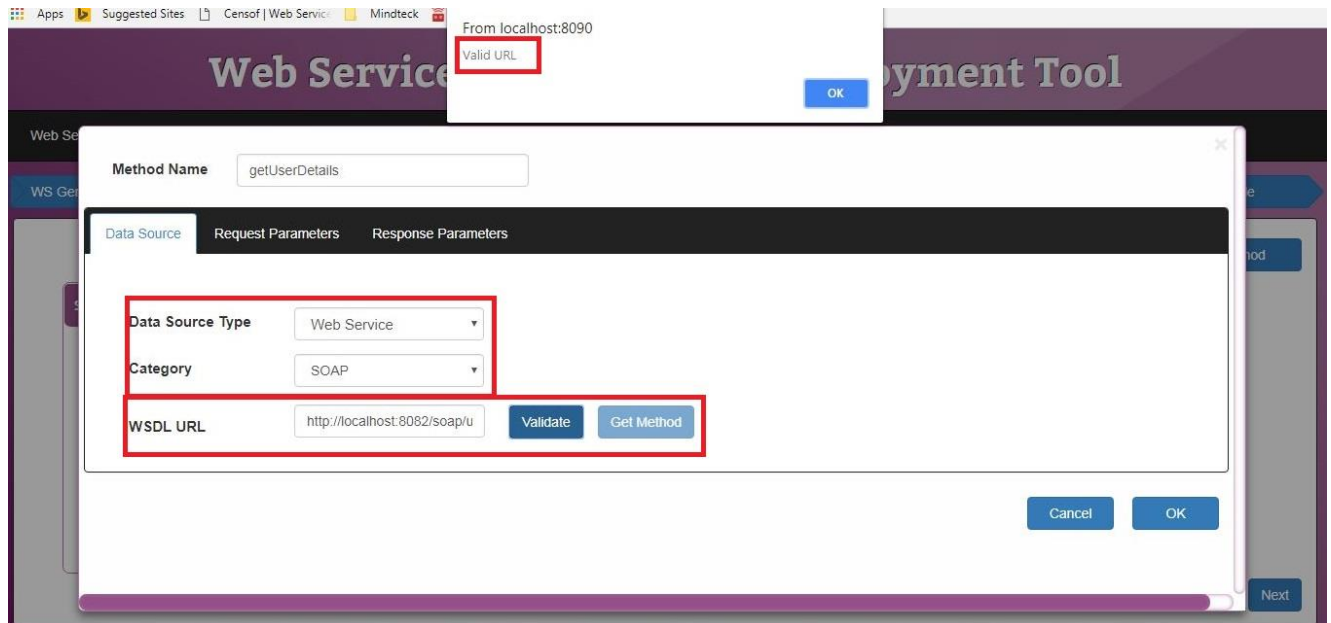
1. Methos will be displayed on the section

2. Click on Next after method is added

[Previous](#) [Next](#)

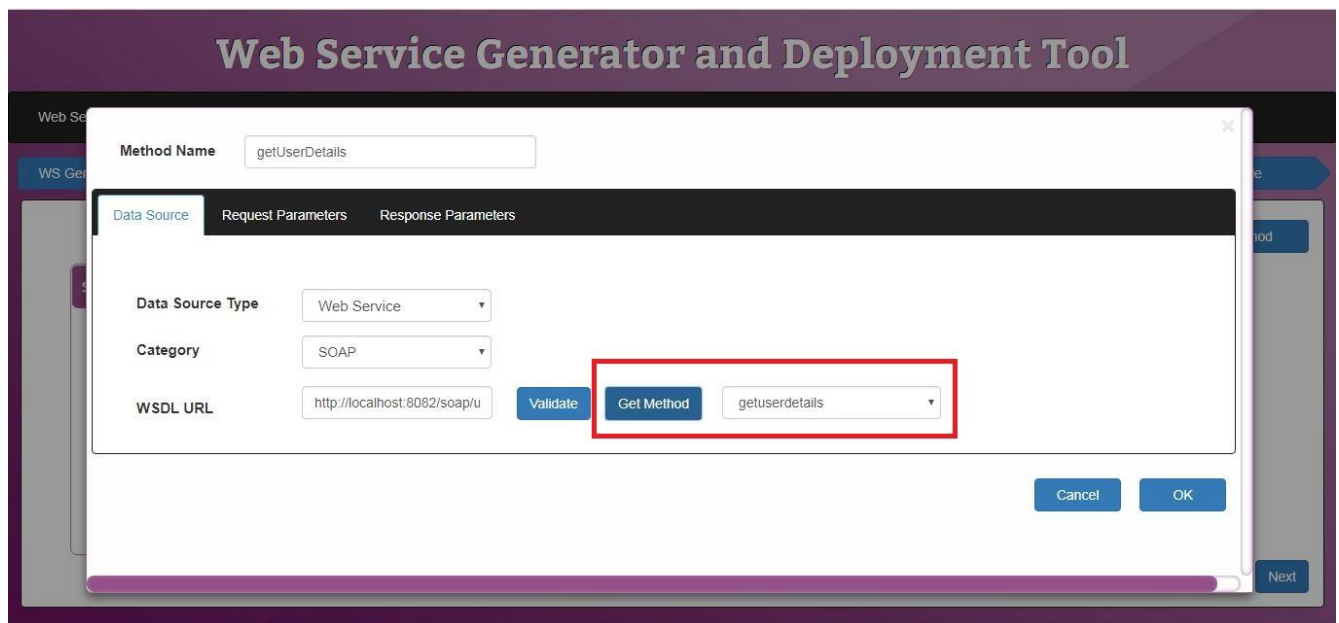
2.2.2.4.1 Add Web Service as Data source

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



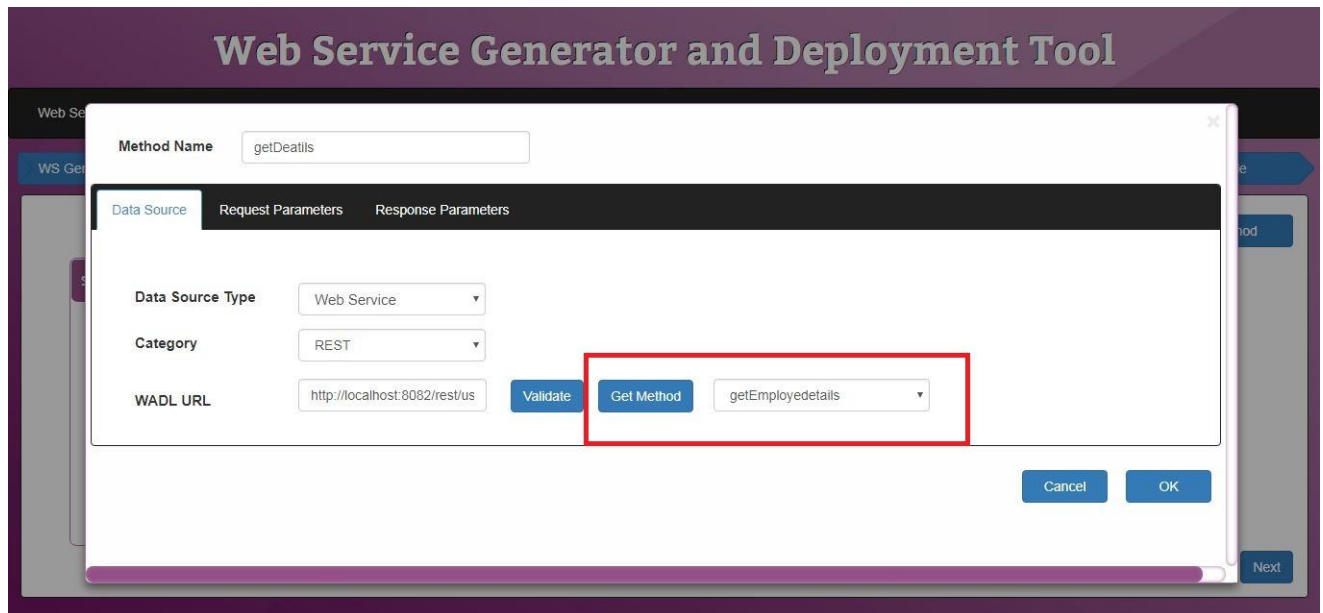
2.2.2.4.2 Add Web Service as Data source

- 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

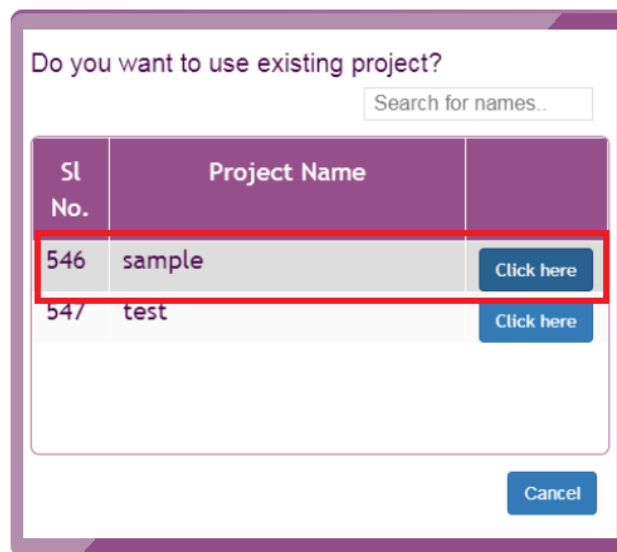
- Select Data Source as REST service and add Methods from the service.



The screenshot shows the 'Web Service Generator and Deployment Tool' interface. A modal window is open with the 'Data Source' tab selected. The 'Method Name' field contains 'getDeatils'. The 'Data Source Type' is set to 'Web Service' and the 'Category' is 'REST'. The 'WADL URL' is 'http://localhost:8082/rest/us'. A red box highlights the 'Get Method' button and the dropdown menu showing 'getEmployedetails'. The 'Validate' button is also visible. At the bottom right of the modal are 'Cancel' and 'OK' buttons. A 'Next' button is visible on the right side of the main interface.

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.

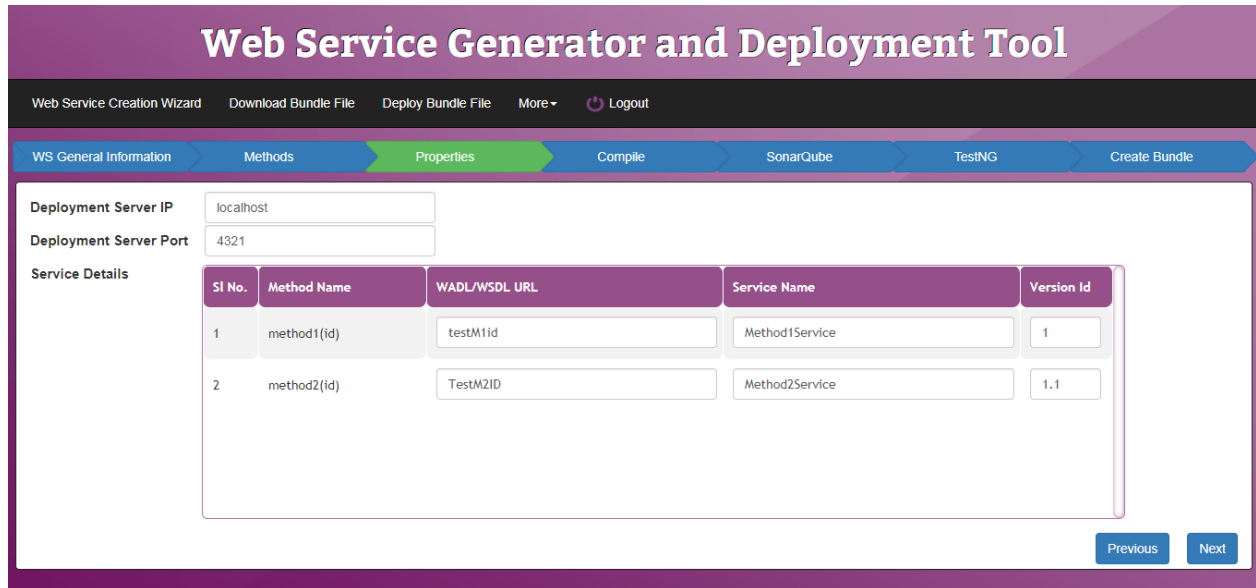


The screenshot shows a dialog box titled 'Do you want to use existing project?'. It contains a search bar labeled 'Search for names..'. Below the search bar is a table with two columns: 'Sl No.' and 'Project Name'. The table has two rows: one with '546' and 'sample', and another with '547' and 'test'. A red box highlights the row with '546' and 'sample', and the 'Click here' button next to it. A 'Cancel' button is at the bottom right of the dialog.

Sl No.	Project Name	
546	sample	Click here
547	test	Click here

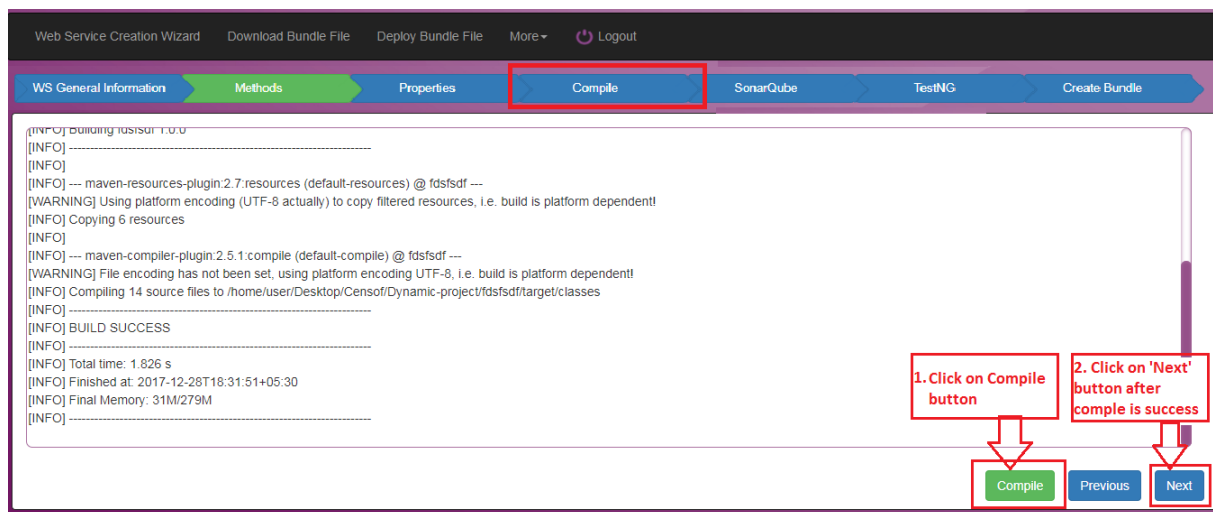
2.2.2.6 Properties page

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



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

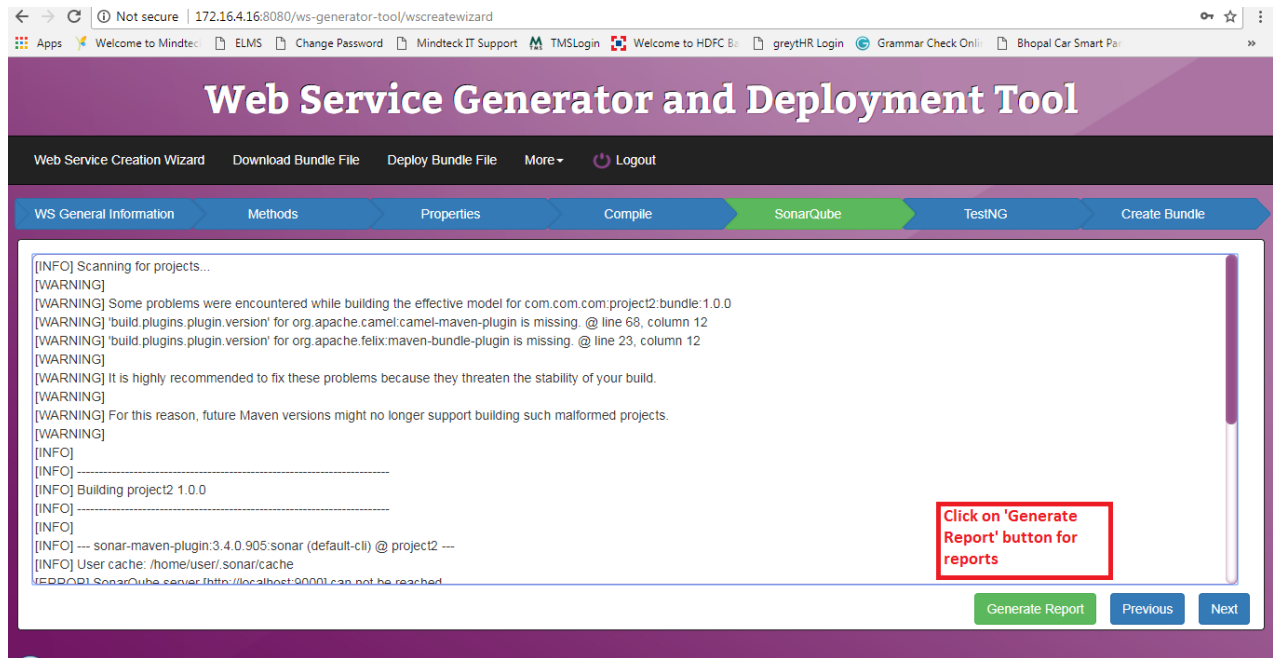
2.2.2.7 Compile page



[INFO] Building project 1.0.0
[INFO] ---
[INFO] --- maven-resources-plugin:2.7:resources (default-resources) @ fdfsdf ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 6 resources
[INFO] ---
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ fdfsdf ---
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 14 source files to /home/user/Desktop/Censoft/Dynamic-project/fdfsdf/target/classes
[INFO] ---
[INFO] BUILD SUCCESS
[INFO] ---
[INFO] Total time: 1.826 s
[INFO] Finished at: 2017-12-28T18:31:51+05:30
[INFO] Final Memory: 31M/279M
[INFO] ---

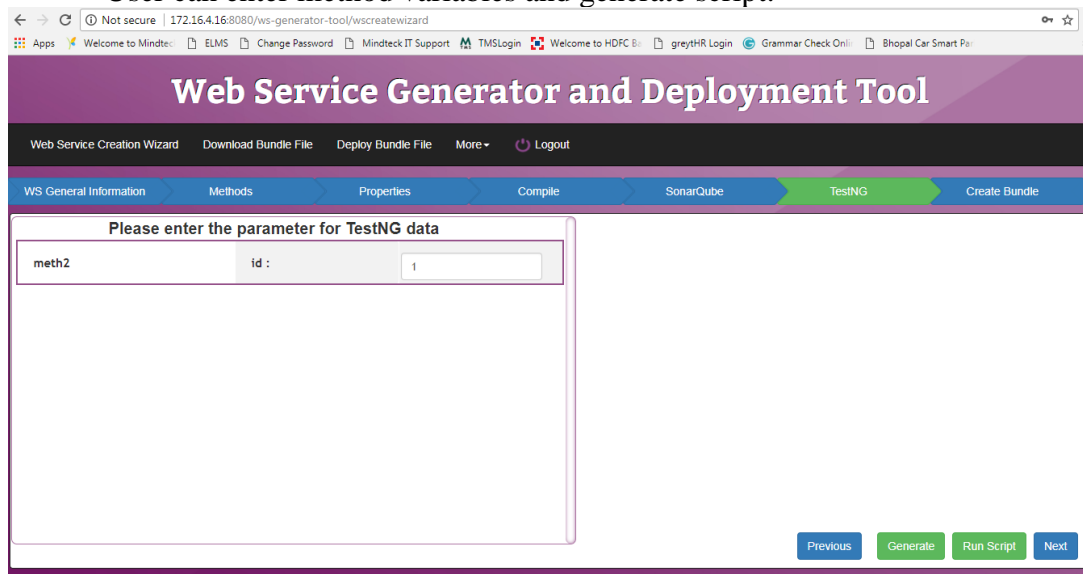
1. Click on Compile button
2. Click on 'Next' button after compile is success

2.2.2.8 SonarQube



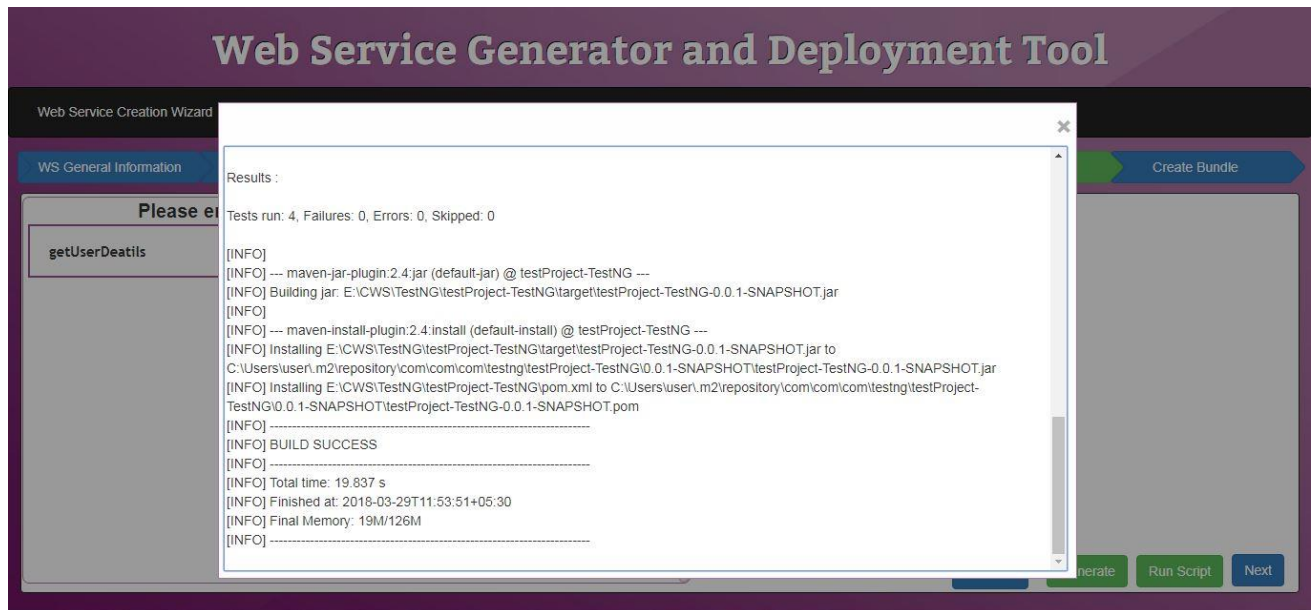
2.2.2.9 TestNG Screen

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



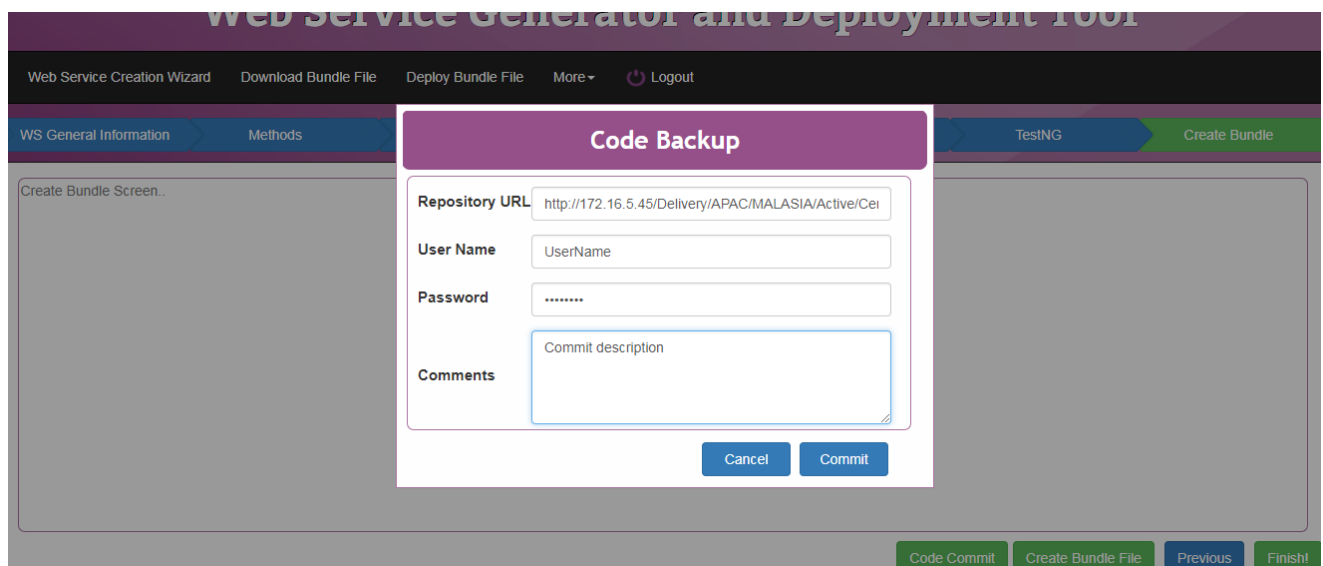
2.2.2.9.1 TestNG result

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



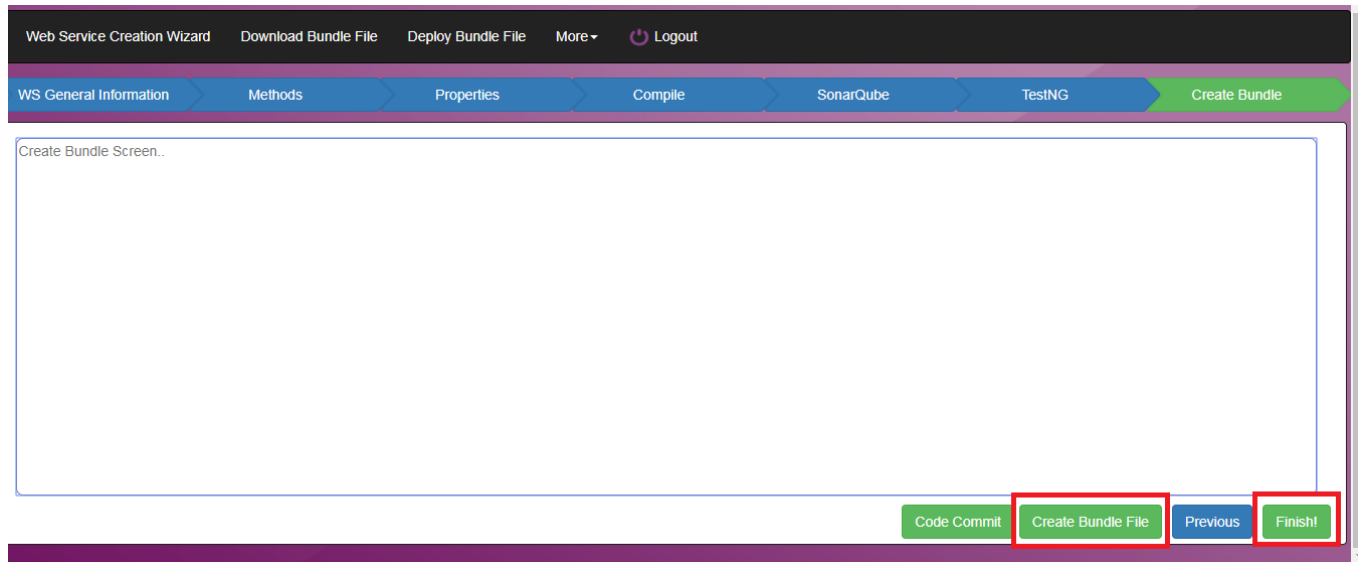
2.2.2.10 Commit create code to Repository

- 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.



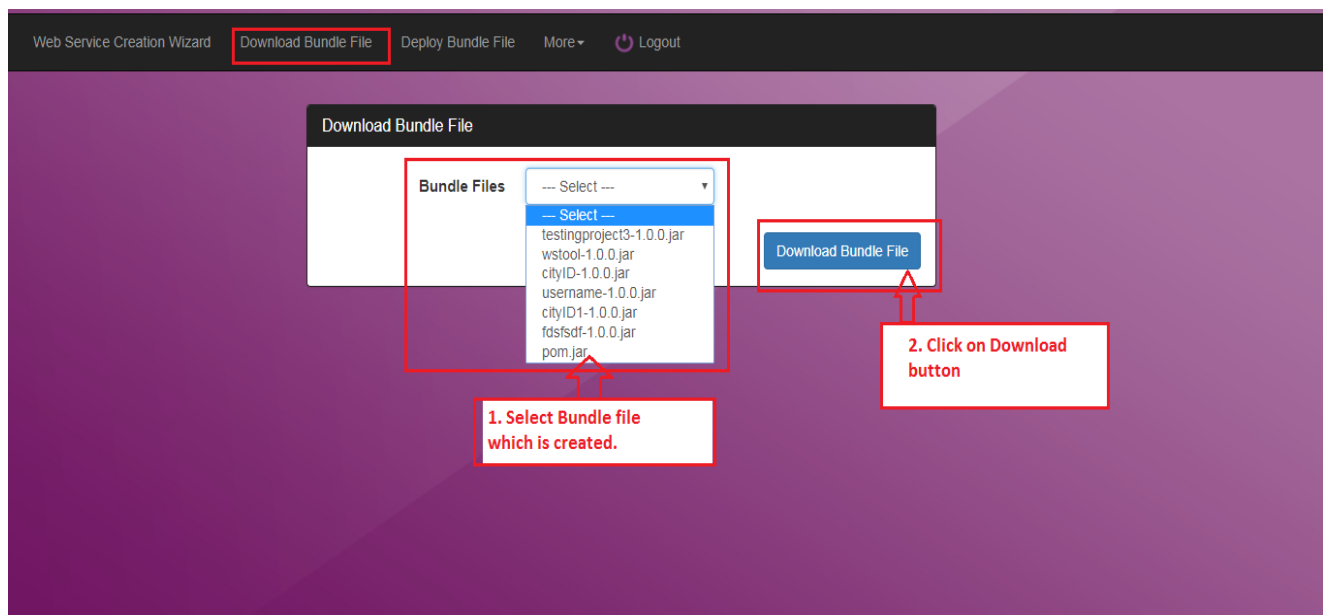
2.2.2.11 Create Bundle page

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



2.2.3 Download Bundle File

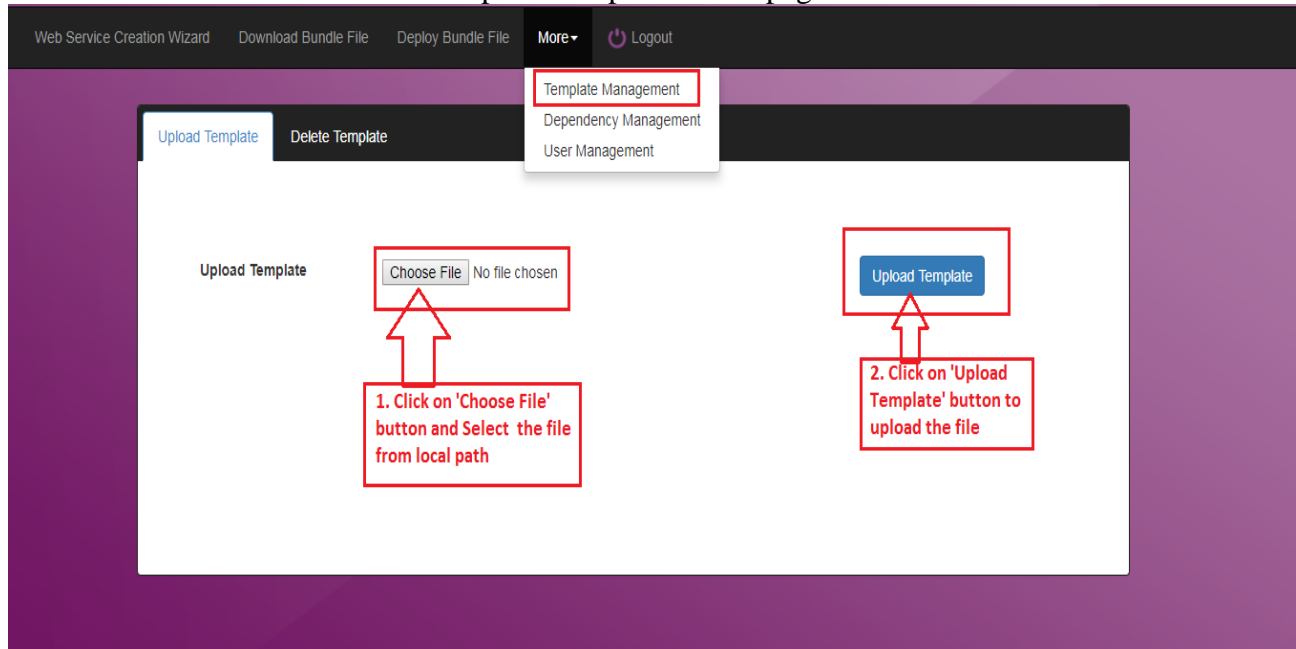
Created bundle file should be displayed on Bundle file Dropdown on 'Download bundle file section as shown below.



2.2.4 Template Management

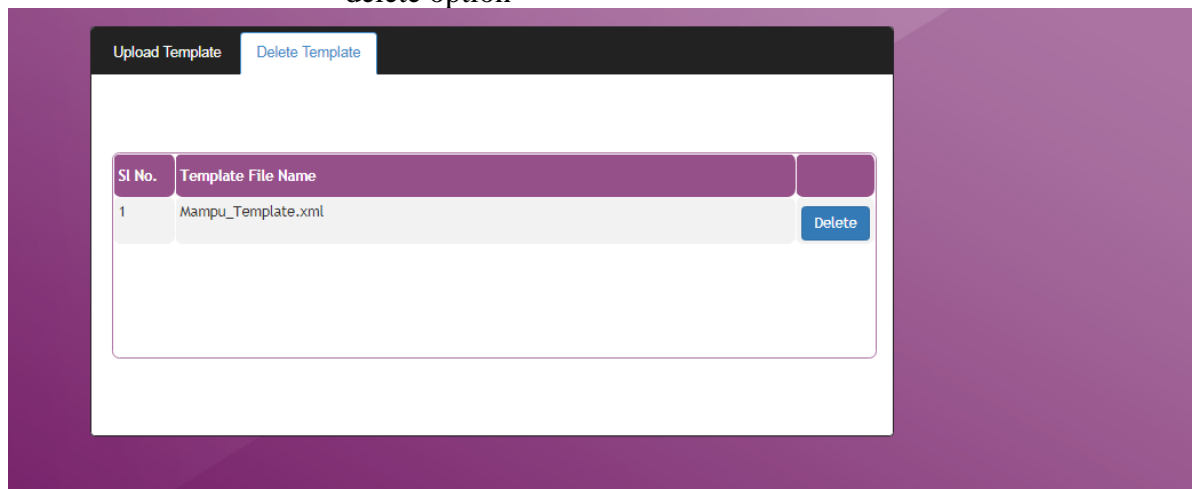
2.2.4.1 Upload Template

- 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

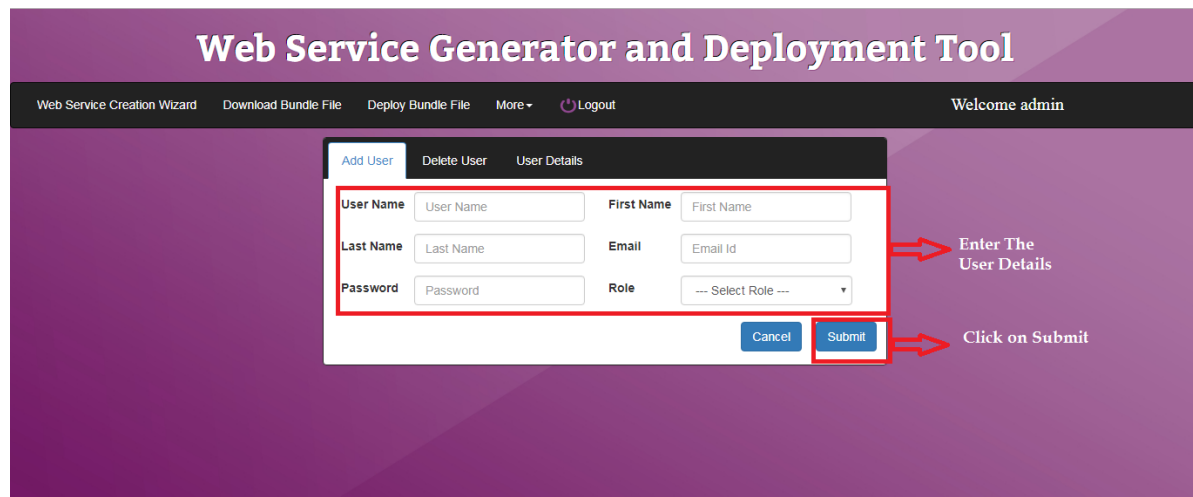
User roles and their responsibilities.

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.



Web Service Generator and Deployment Tool

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

Add User Delete User User Details

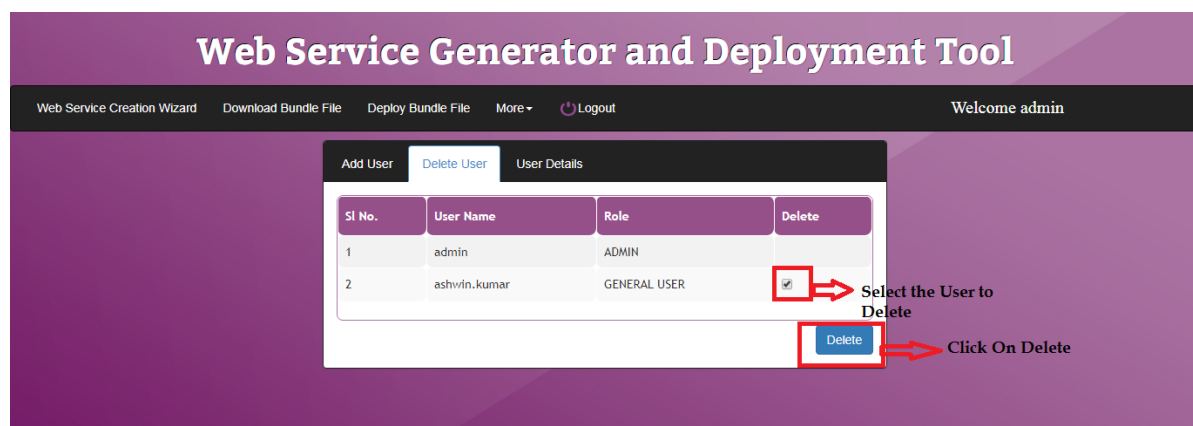
User Name First Name Last Name Email Password Role

Cancel Submit

Enter The User Details

Click on Submit

2.2.5.2 Delete User



Web Service Generator and Deployment Tool

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

Add User **Delete User** User Details

Sl No.	User Name	Role	Delete
1	admin	ADMIN	<input type="checkbox"/>
2	ashwin.kumar	GENERAL USER	<input checked="" type="checkbox"/>

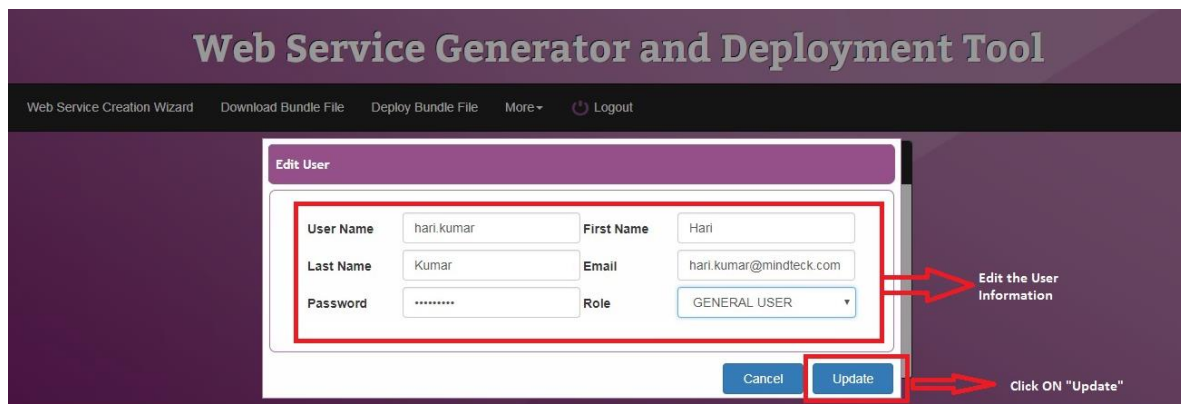
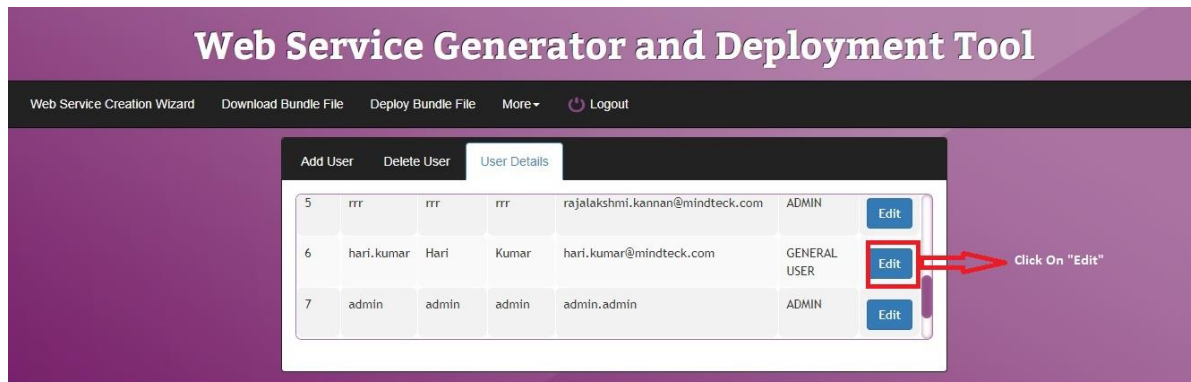
Delete

Select the User to Delete

Click On Delete

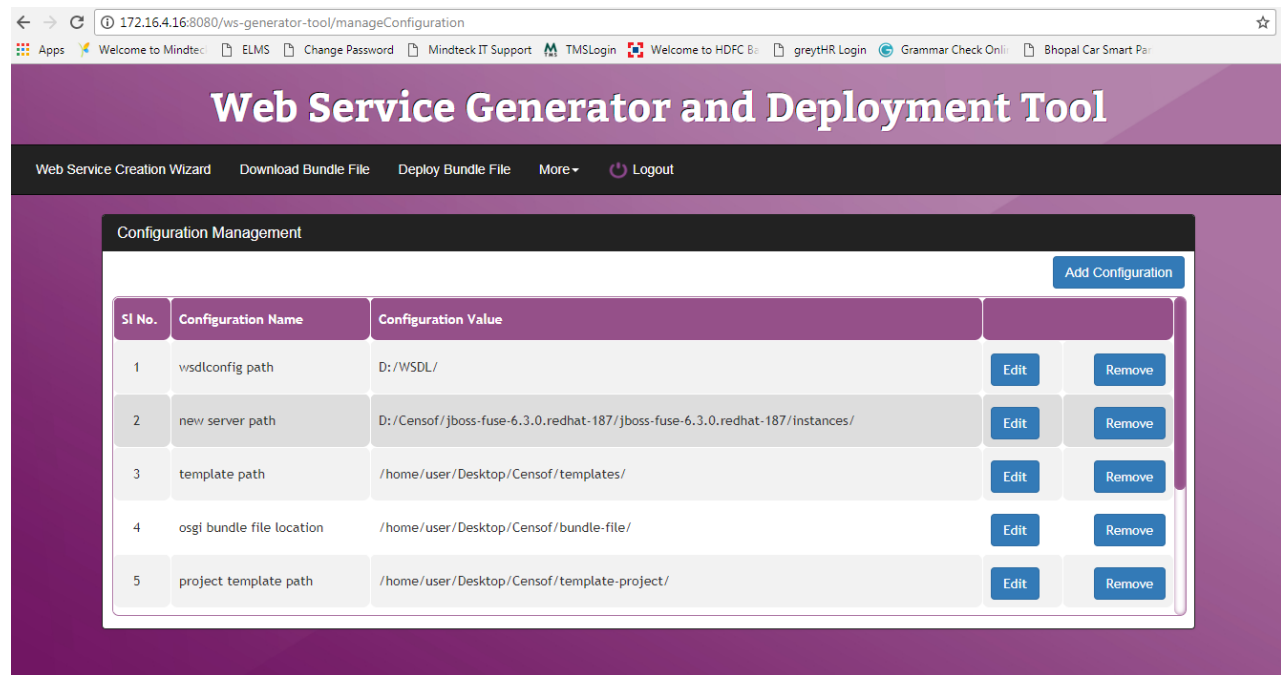
2.2.5.3 Edit User Details

- Admin User can edit the User Information.



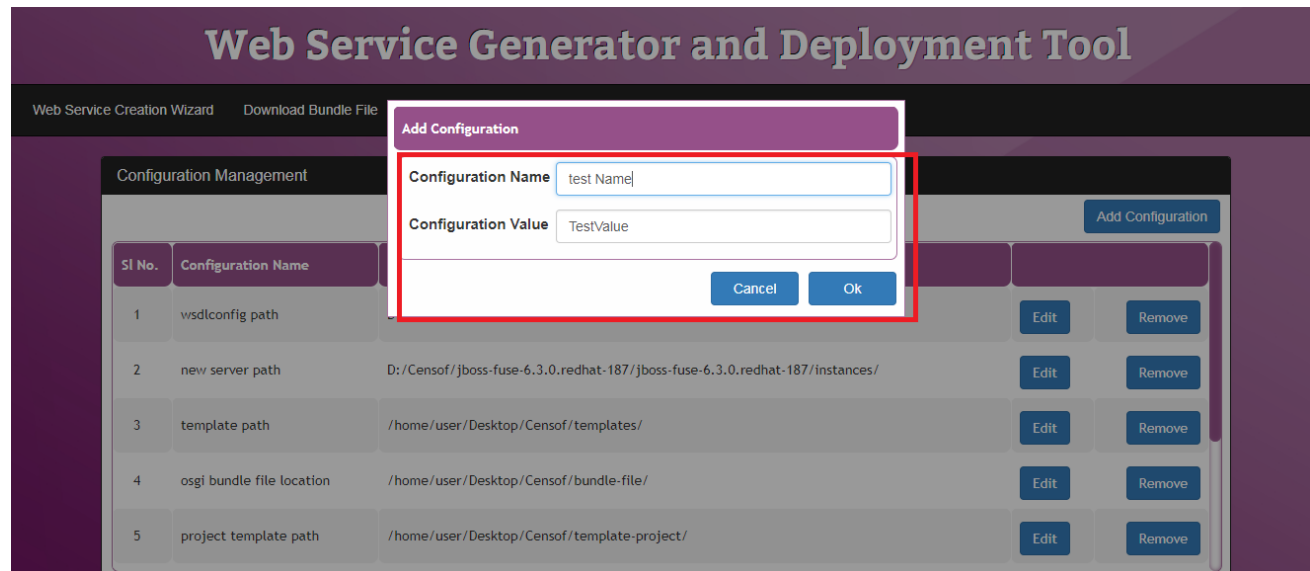
2.2.6 Configuration Management:

- User can add configuration details in Configuration management page.



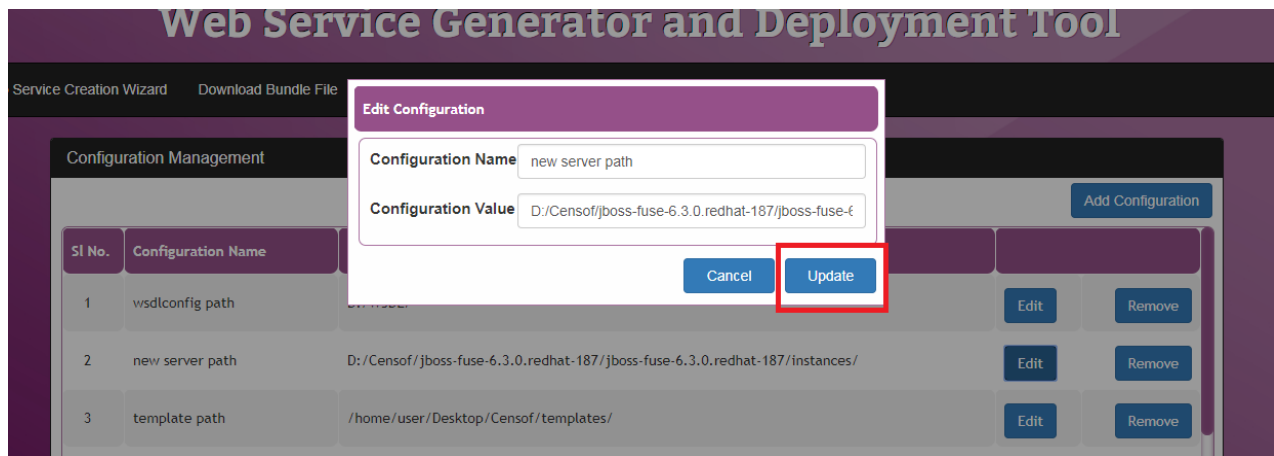
2.2.6.2 Add Configuration details

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



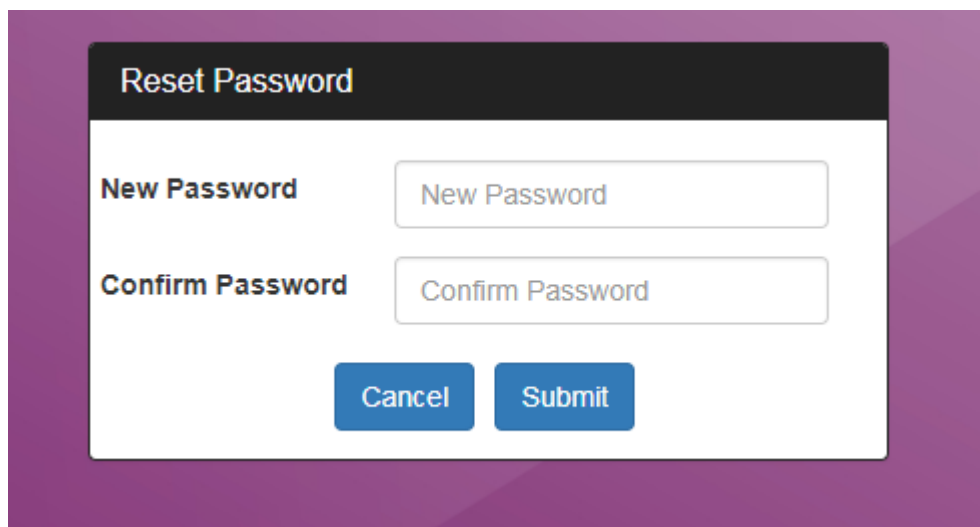
2.2.6.3 Update Configuration details:

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



2.2.7 Change Password

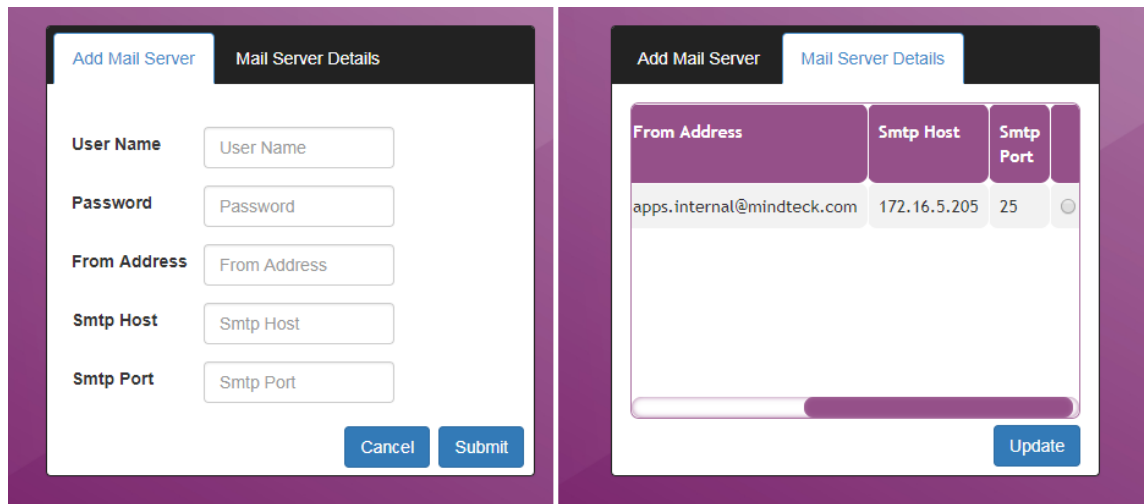
- Users can change the Password using the feature Change Password under More options shown below.



The screenshot shows the 'Reset Password' form. It includes two input fields: 'New Password' and 'Confirm Password'. Below the input fields are 'Cancel' and 'Submit' buttons.

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

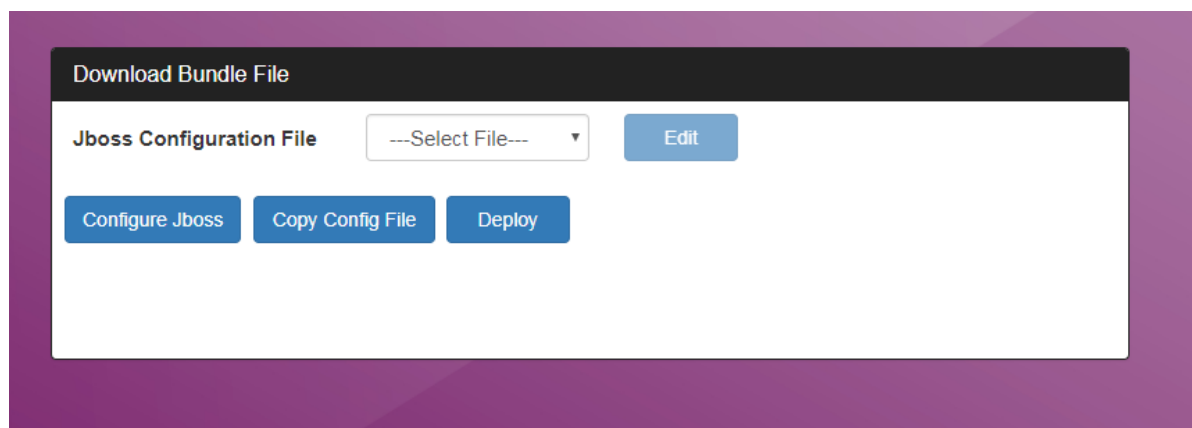


The image shows two side-by-side screenshots of a web application interface for configuring mail servers. The left screenshot shows the 'Add Mail Server' tab with input fields for User Name, Password, From Address, Smtip Host, and Smtip Port, along with Cancel and Submit buttons. The right screenshot shows the 'Mail Server Details' tab with a table for managing mail servers. The table has columns for From Address, Smtip Host, Smtip Port, and an activation toggle. One server is listed with the email address apps.internal@mindteck.com, host 172.16.5.205, and port 25. An Update button is at the bottom right.

From Address	Smtip Host	Smtip 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.



The image shows a web application interface for deploying a bundle file. It features a 'Download Bundle File' header, a 'Jboss Configuration File' dropdown menu with a '---Select File---' option, and an 'Edit' button. Below these are three buttons: 'Configure Jboss', 'Copy Config File', and 'Deploy'.

- JBoss Fuse Configuration file

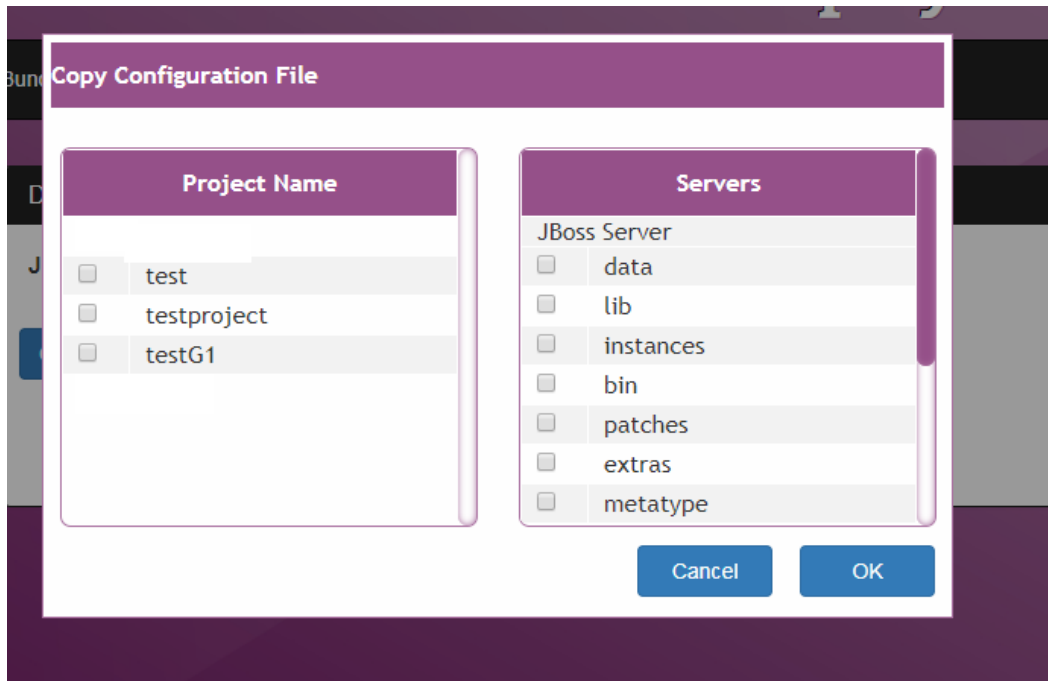


- Fuse Commands

- 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

