AuraPlayer Server Manager User Guide

Contents

Introduction

Service Manager

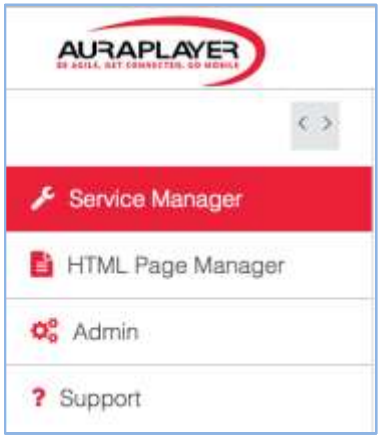
HTML Page Manager

File Manager (.orp)

Introduction

AuraPlayer's Service Manager is the tool used to create services and HTML pages as well as store, share and manage them. It also makes it possible to access the files from anywhere.

Main Menu

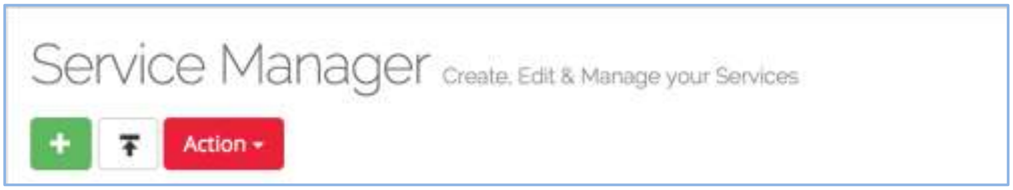


The Service Manager Menu has the following components:

Service Manager:	Area to create and manage Services - Services list, Service Creation, Service editing options, etc.
HTML Page Manager:	Area to create and manage HTML Pages - HTML pages list and HTML page editing options
Admin:	Area to manage administrative tasks and preferences - ORP files, Backup data, restore data from a backup file, and system information.
Support:	Support FAQ, videos, articles etc.

Service Manager

Service Manager Toolbar

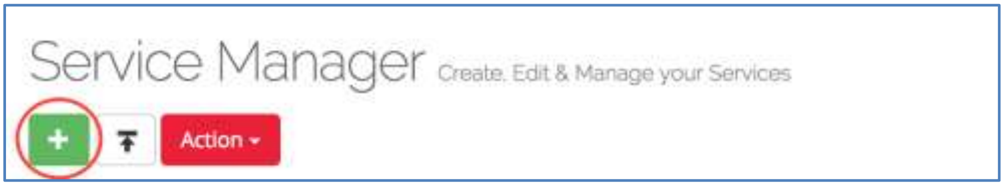


The Service Manager Toolbar has the following components:

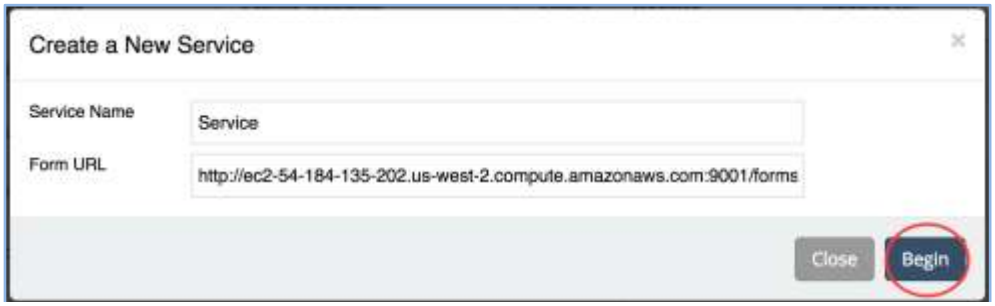
Record/Create:	To record a new service
Upload:	Upload a new service from an existing ORP file
Actions:	All actions can be done in batch by by checking the box next to all relevant Services.
Activate/disable:	Activate or Disable a service.
Export:	Export the list of services to a csv file.
Print:	Print the list of services
Delete:	Delete a service. Can be used in batch by checking the box next to all relevant services.

Recording/Creating a Service

1. Click “Record/Create” on the Service Manager Toolbar, the 'create a new service' window will appear.



2. Fill out the service information:



Create a New Service

Service Name: Service

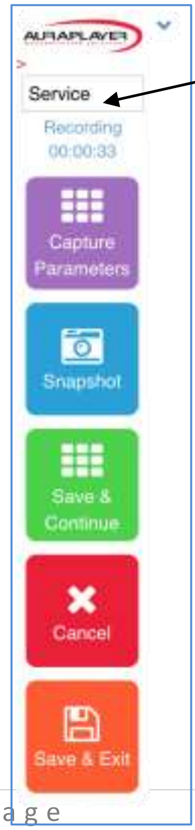
Form URL: http://ec2-54-184-135-202.us-west-2.compute.amazonaws.com:9001/forms

Service Name:	The unique name you want to call your service. It should reflect the unique name for your business process. For example: CreateOrder
Form URL:	The URL of the Forms system you wish to connect and record from. You can only record from a Forms server where AuraPlayer has been installed. See install guide for details.

3. Click “Begin” to begin creating your service.
4. The "Recording Toolbar" and the "Oracle Forms" windows will appear.

Recording Toolbar

The Recording Toolbar has the following components:



Service

Recording
00:00:33

File name:	Service name goes here
Status:	Indicates if the Service Manager is recording or on standby
Timer:	Shows the duration of the recording
Capture Parameters:	Capture all the Form's existing fields and assign them as output parameters for the service
Snapshot:	Create a replica of the existing Oracle Forms canvas screen as an HTML page
Save & Continue:	Saves the recorded service so far and allows continuation of recording later. Used to capture list of values mid-way through recordings.
Cancel:	Cancel the current recording
Save & Exit:	Save the recorded service and stop recording

5. **Creating Input Parameters –** To create an input parameter simply input text into a text box, setting a value in a field, clicking on check box, selecting option, any entering of values in the Forms session will create the value as an input field.
6. **Creating Output Parameters**
    a. To capture specific output parameters **-** Click on the desired text fields in the Form, they will be captured as output parameters.
    b. To capture ALL of the output parameters on a specific canvas **-** Click the "Capture Parameters" button on the toolbar. All the fields in the Form will be captured as output parameters

7. To stop recording without saving as a webservice, click on the “Cancel” button.
8. To save and end the recording, click the “Save & Exit” button. The service is now saved and can be found in the 'Service Manager'. Once pressing the “Save & Exit” button, you will be routed to the 'Edit Service' page to finalize the creation of the Service.

Recording Feature: Parameters

The "Capture Parameters" feature allows the user to capture output parameters for all fields in the Oracle Form that is now being recorded.

1. Once you reach the Form that has the fields you would like as output fields on your mobilized application, on the Recording Toolbar, click the “Capture Parameters” button
2. Upon success, your recording toolbar will have a success message on the top right corner.

    These parameters will then be added to the output parameters list of your Service. For more details see the "Edit a Service" Section.

Recording Feature: Snapshot

The "Snapshot" feature captures the Oracle Form (showing during the recording process) and recreates it as an HTML page (without the Oracle Forms functionality).

1. To create a snapshot, begin the recording process. Navigate to the canvas you wish to convert to an HTML page.

2. Once you have reached the canvas you want to convert, click the “Snapshot” icon.

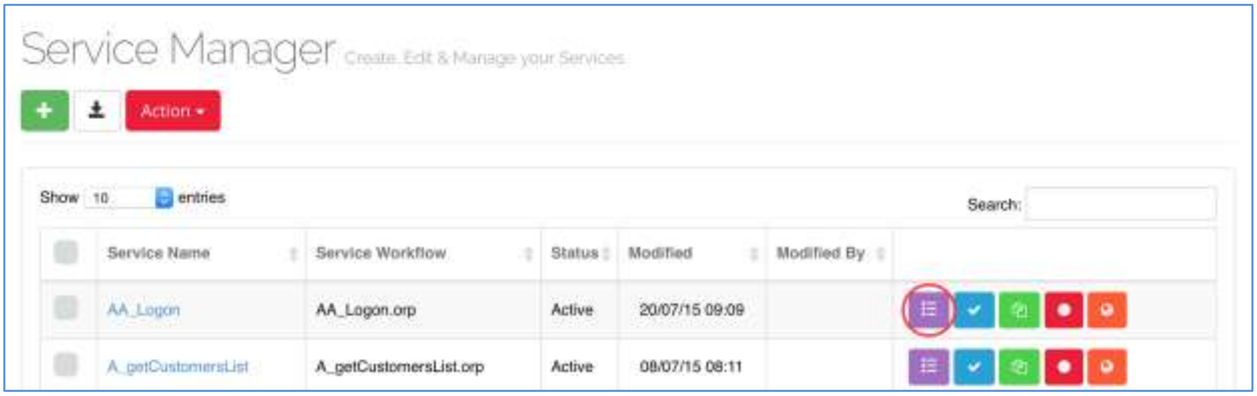
Note: Make sure the status on the recording toolbar reads “Recording”.

After clicking “Snapshot”, an HTML replica of the Forms canvas is created and can be found in the "HTML Page Manager”.

The HTML file created by the snapshot feature is pure HTML and has no functionality. To add functionality and edit the file, see the section “Editing an HTML Page”.

Service Details

In order to see the service details, click on the Details icon next to the service name in the "Service Manager".



Service Manager Create, Edit & Manage your Services

Show 10 entries

Search:

□	Service Name	Service Workflow	Status	Modified	Modified By	
□	AA_Logon	AA_Logon.orp	Active	20/07/15 09:09		
□	A_getCustomersList	A_getCustomersList.orp	Active	08/07/15 08:11		

The Service "Details" section contains the following data:

Service Name:	The service name
Description:	Description of service set by the user when creating the Service
Service URL:	The RESTful service URL. This URL will be used to create RESTful APi's
WSDL URL:	SOAP URL for the service description file (WSDL)
Form URL:	The URL of the Oracle Forms server that the service is running against.
Status:	Service status - active or disabled

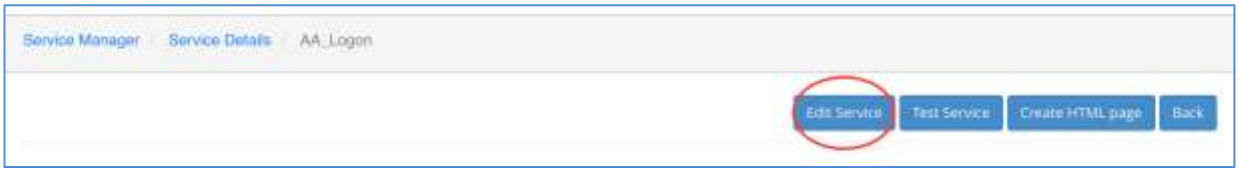
Running AuraPlayer Externally

To run the AuraPlayer Service externally, you can copy the relevant URL from the details page and create the Service client in any tool of your choice. It is also possible to copy the REST URL to your choice of browser.

To see an example of consuming the SOAP Service in ADF Mobile in the Oracle Jdeveloper development tool see this video http://www.youtube.com/watch?v=6PtsaeJUAps&hd=1.

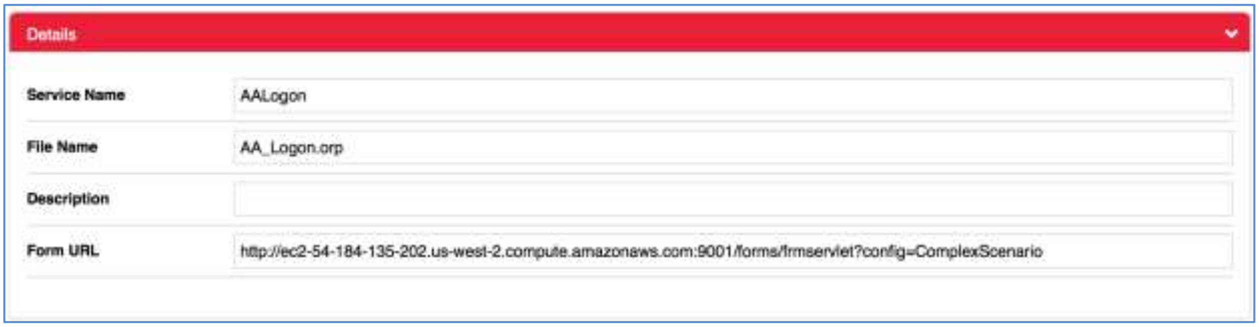
Editing the Service

The 'Edit Service' option is nested within the 'Service Details' page.
Service Manager -> Service Details -> Edit Service



To edit the Parameters, click the Edit Service button on either the top or bottom right of the page.

Details

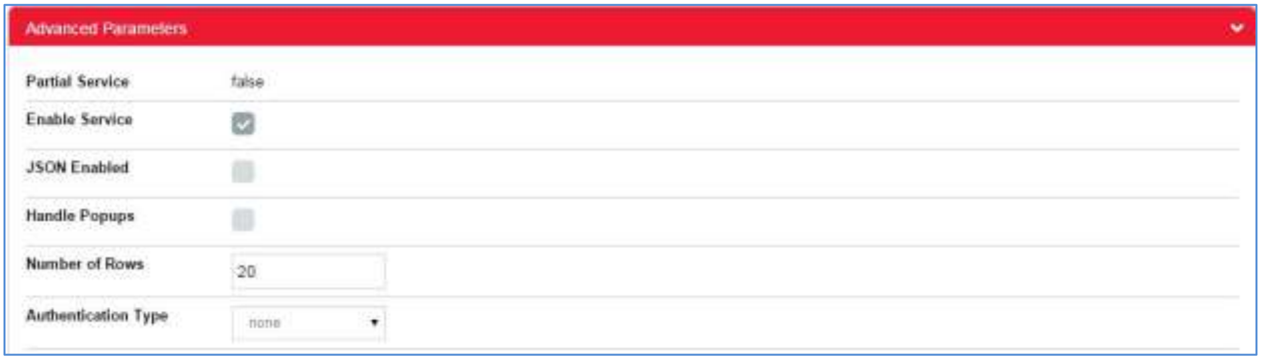


Details

Service Name	AALogon
File Name	AA_Logon.orp
Description	
Form URL	http://ec2-54-184-135-202.us-west-2.compute.amazonaws.com:9001/forms/frmservlet?config=ComplexScenario

Service Name:	Name given to the service at time of recording
File Name:	ORP file name
Description:	What the service does
Form URL:	The URL of the Form server where the Service will be run. It does not have to be the same URL that was used for recording.

Advanced Parameters

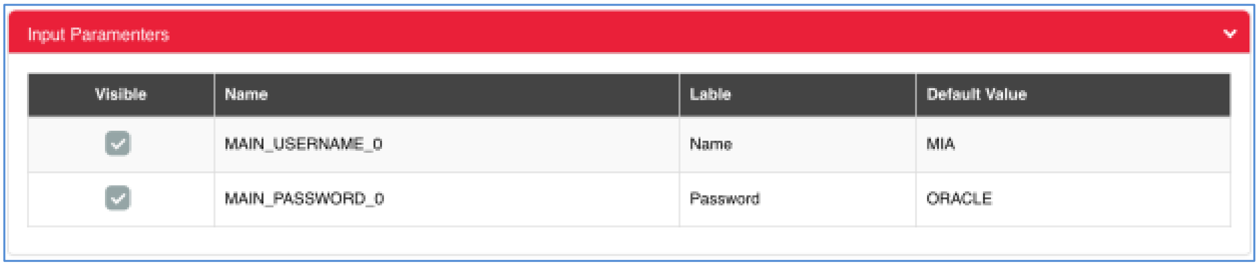


Advanced Parameters

Partial Service	false
Enable Service	☑
JSON Enabled	□
Handle Popups	□
Number of Rows	20
Authentication Type	none

Partial Service:	(Advance Feature) – Partial Service is a recording that doesn't start from the beginning of the scenario and isn't independent. It relies on a previous service to run in order to succeed.
Enabled Service:	By checking this box, you are enabling the Service to be accessible.
JSON enabled:	By checking this box, your REST Service would return a JSON response.
Handle Pop-ups:	Handle Pop-ups deal with unexpected pop-ups that appear in the form while calling a service (during playback). Check this box if you would like to Click on the default button in the popup & continue with the playback.
Number of Rows:	The total number of rows you wish to display in the result set
Authentication:	(Advanced Feature) – Authentication allows you to add an additional security authentication to specific service. Username & Password can be defined in weblogic.

Input Parameters

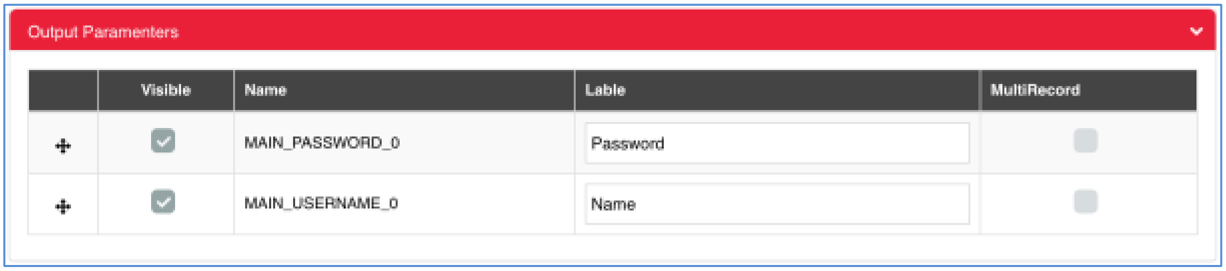


Input Paramenters

Visible	Name	Lable	Default Value
☑	MAIN_USERNAME_0	Name	MIA
☑	MAIN_PASSWORD_0	Password	ORACLE

Output Parameters

The output parameters are the parameters returned by the service.



Output Paramenters

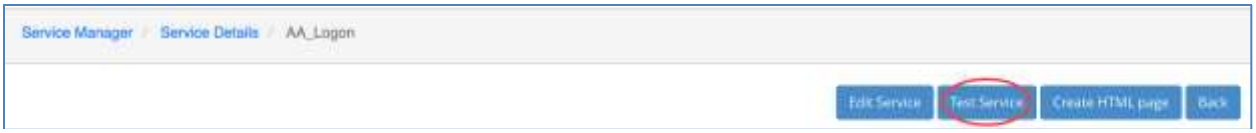
	Visible	Name	Lable	MultiRecord
✥	☑	MAIN_PASSWORD_0	Password	☐
✥	☑	MAIN_USERNAME_0	Name	☐

Once saved, you will be redirected back to the Service Manager.

Testing the Service

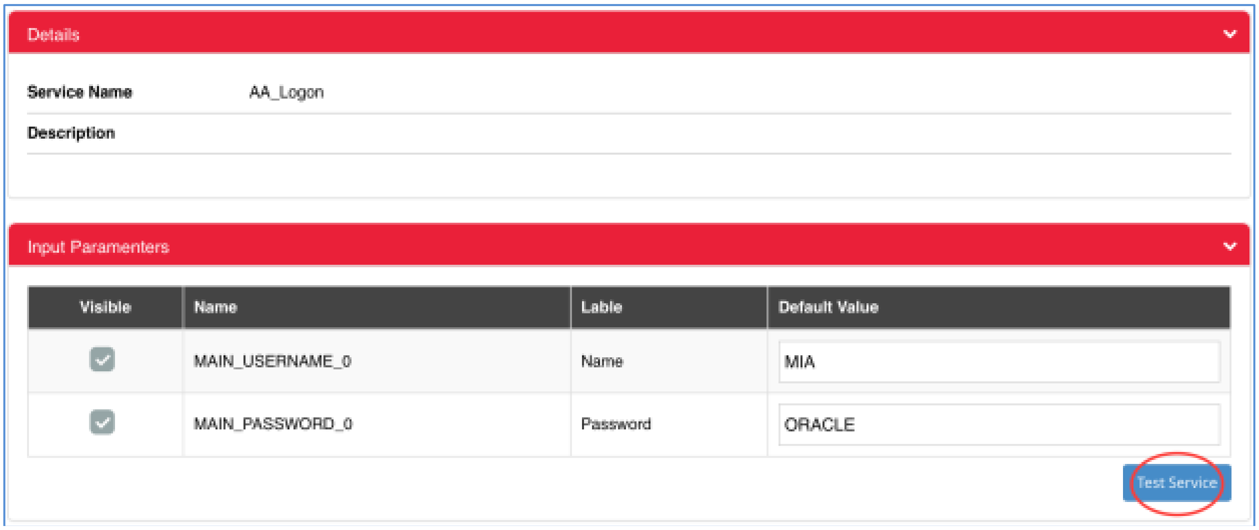
The 'Test Service' option is nested within the 'Service Details' page.
Service Manager -> Service Details -> Test Service

1. In order to test the service created, click on the 'Test Service' button.



2. The "Test Service" page appears with the default input values set during service creation.

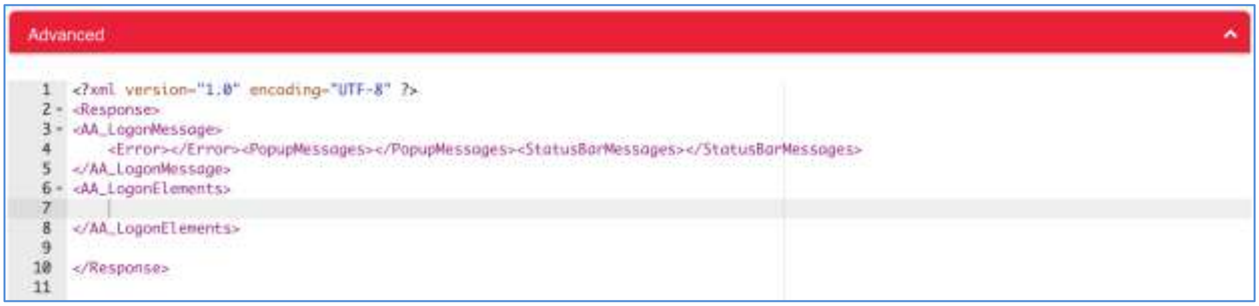
3. To change the input parameters default values, simply change the values in the "Default Value" column.



Service Name	AA_Logon
Description	

Visible	Name	Lable	Default Value
☑	MAIN_USERNAME_0	Name	MIA
☑	MAIN_PASSWORD_0	Password	ORACLE

4. Once all values are set, click the "Test Service" button. The XML response will appear in the text box below



```
<?xml version="1.0" encoding="UTF-8" ?>
<Response>
<AA_LogonMessage>
    <Error></Error><PopupMessages></PopupMessages><StatusBarMessages></StatusBarMessages>
</AA_LogonMessage>
<AA_LogonElements>

</AA_LogonElements>

</Response>

```

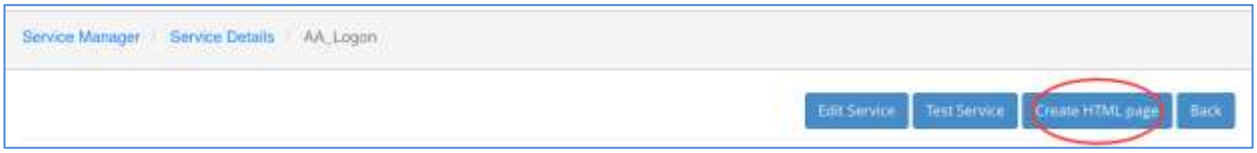
The Oracle Forms error messages, status bar messages and popup messages can be found at the top of the response.

HTML Page Manager

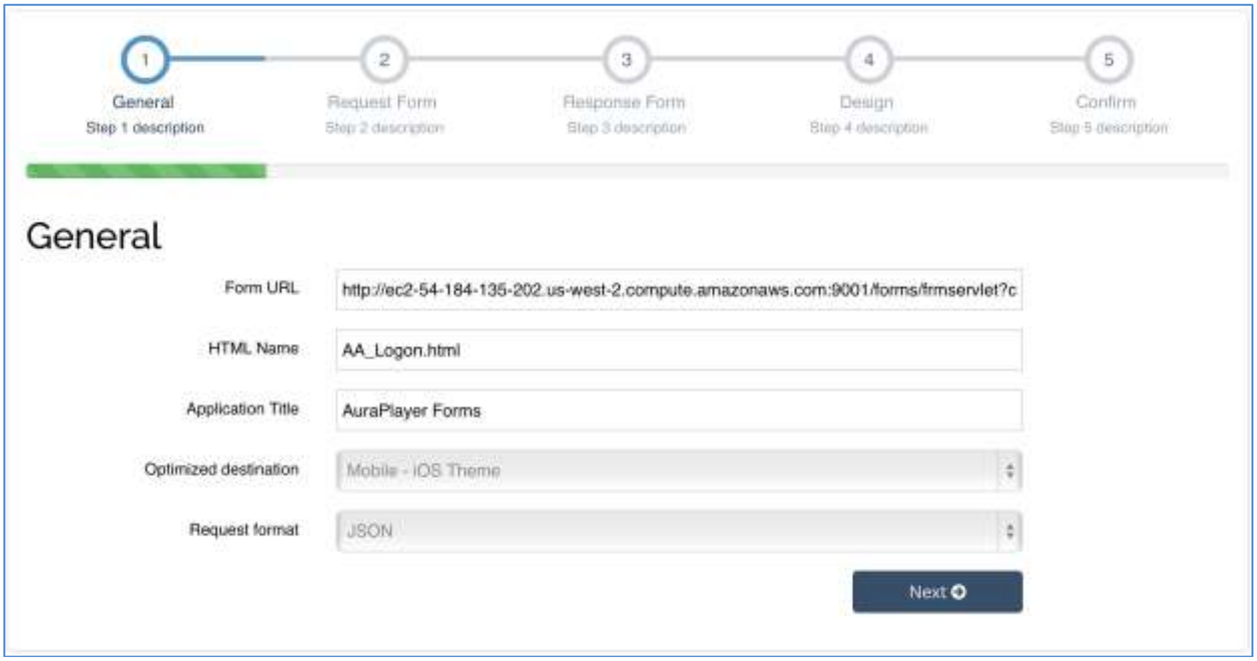
Creating an HTML Page - Visualize

The 'Create HTML Page' option is nested within the 'Service Details' page.
Service Manager -> Service Details -> Create HTML Page

1. To create an HTML page, click on the "Create HTML Page" button.



You will be redirected to the "**General**" page of the HTML generator.


1 General Step 1 description
2 Request Form Step 2 description
3 Response Form Step 3 description
4 Design Step 4 description
5 Confirm Step 5 description
Next

General

Form URL: http://ec2-54-184-135-202.us-west-2.compute.amazonaws.com:9001/forms/frmservlet?c

HTML Name: AA_Logon.html

Application Title: AuraPlayer Forms

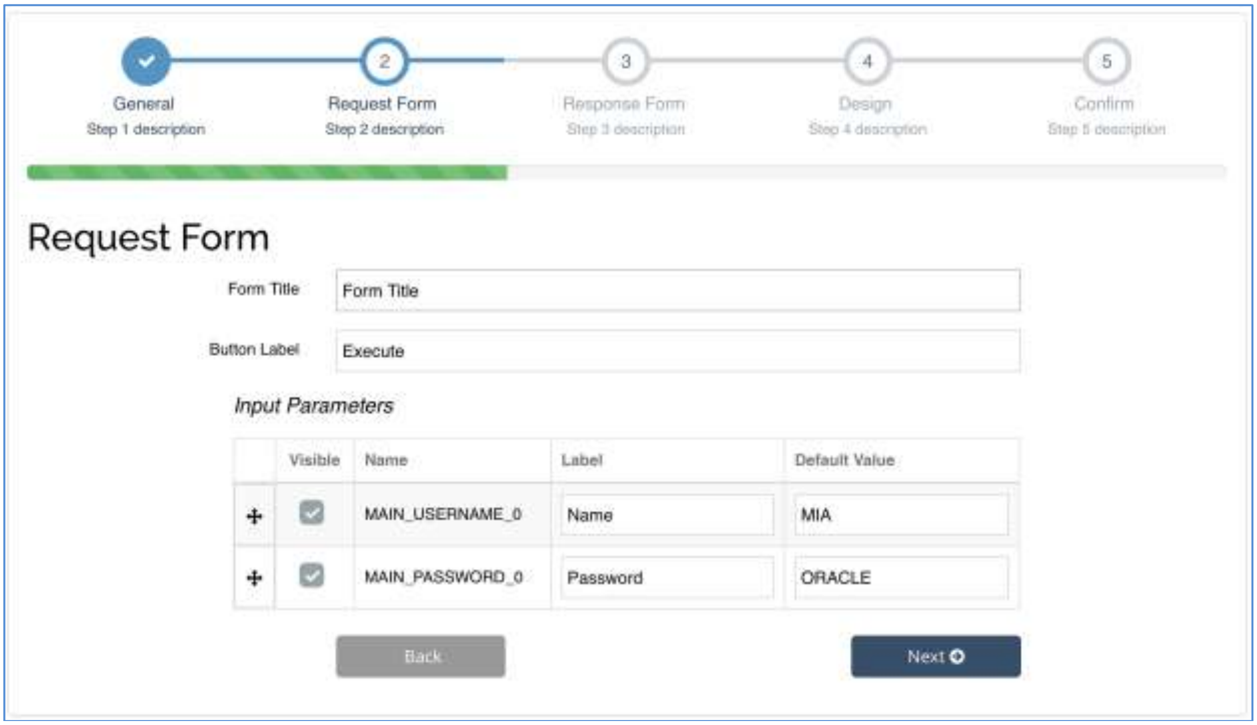
Optimized destination: Mobile - iOS Theme

Request format: JSON

2. Fill in the:

Form URL:	The URL of the Form on which the service will run
HTML Name:	**The name you wish to give the HTML page.**
Application Title:	**The Application Title appears at the top of the browser. Selecting the check box "Add service name" will change the Application Title to that of the service name**
Optimized destination:	Choose the desired platform styling for your HTML page. Options include iPhone, Android, Desktop etc..
Request format: (Advanced)	Choose the format for the response either JSON or XML

3. Click "Next" to move the “Request Form” page



Request Form

Form Title: Form Title

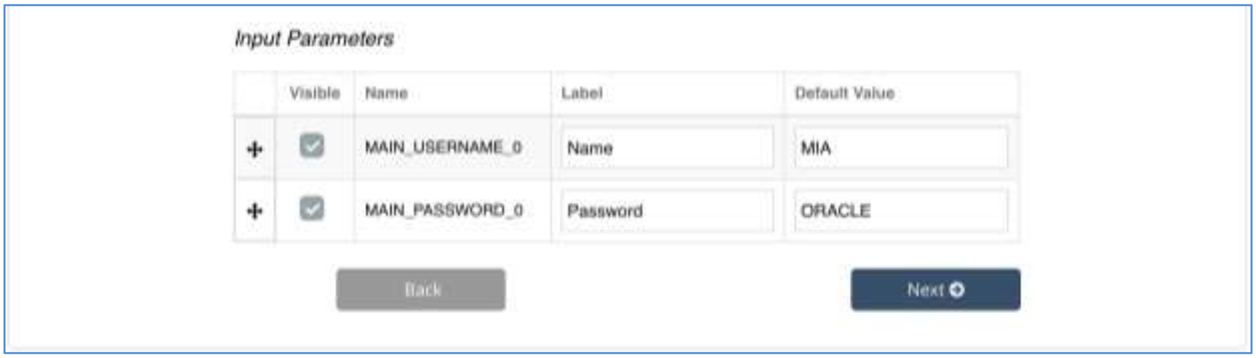
Button Label: Execute

Input Parameters

	Visible	Name	Label	Default Value
✢	☑	MAIN_USERNAME_0	Name	MIA
✢	☑	MAIN_PASSWORD_0	Password	ORACLE

4. Fill in the:

Form Title:	The Form Title appears at the top of the HTML page.
Button Label:	The label you wish to have showing on the button.



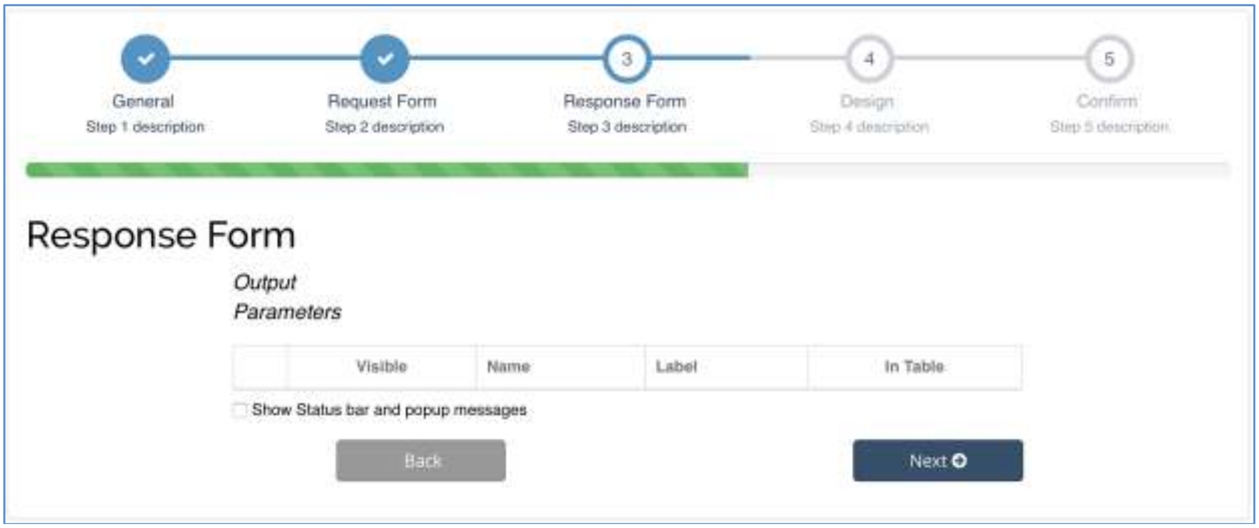
Input Parameters

	Visible	Name	Label	Default Value
✥	☑	MAIN_USERNAME_0	Name	MIA
✥	☑	MAIN_PASSWORD_0	Password	ORACLE

5. The input parameters inserted into the service:

✥	You can use the arrows icon to change the order of the fields.
Visible:	By checking this box, the field becomes visible to the end user
Name:	The name as it appears in the Oracle Forms system FMB
Label:	The title you wish to appear for this field on the HTML page
Default Value:	The default value that will appear in that field if it is set to visible

6. Click "Next" to move to the “Response Form” page



Response Form

Output Parameters

	Visible	Name	Label	In Table

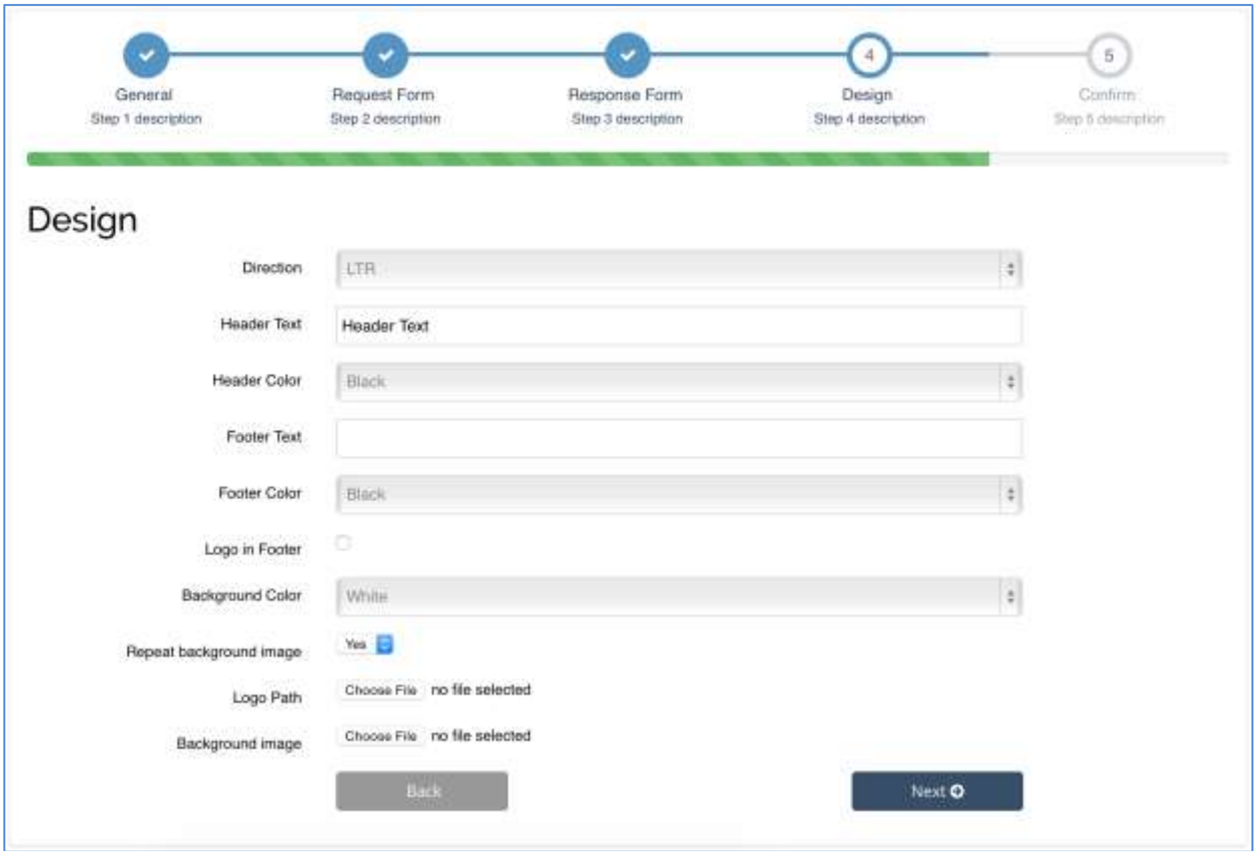
□ Show Status bar and popup messages

The output parameters are the values returned from the Forms system by the service

✥	You can use the arrows icon to change the order of the fields.
Visible:	By checking this box, the field becomes visible to the end user
Name:	The name as it appears in the Oracle Forms system FMB

Label:	The title you wish to appear for this field on the HTML page
In Table:	By checking this box, you will receive multiple results for this Field. In the HTML page this will be set as a table.
Show Status bar and popup messages:	Checking this box will add an additional fields to the response page, containing the messages from the Oracle Form. By default, any popup messages, error messages, or status bar messages will be found in the results page.

7. Click "Next" to move to the “Design” page here you can edit the layout of the HTML page.



Design

Direction: LTR

Header Text: Header Text

Header Color: Black

Footer Text:

Footer Color: Black

Logo in Footer: □

Background Color: White

Repeat background image: Yes

Logo Path: Choose File no file selected

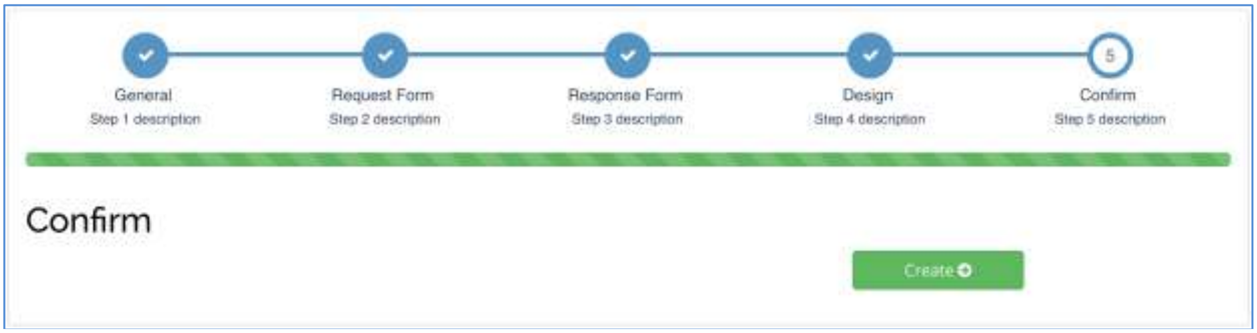
Background image: Choose File no file selected

Here, you are able to use your creativity to design the basic look of your HTML page. You can change the color of the text and the background.

Direction:	Choose the direction of the page right-to-left or left-to-right
Header Text:	This is free text that will appear at the top of your page
Header Color:	Set the color of the header text defined above
Footer Text:	This is free text that will appear at the bottom of your page
Footer Color:	Set the color of the footer text defined below

Logo in Footer:	Choose if the logo image should appear in the page footer
Background Color:	Define the background color of the page
Repeat Background Image:	If a background image is set. This will define if it is repeated
Logo Path:	Choose the logo image file to show at the top of the page
Background Image:	Choose an image to appear as the background image. This is optional. You may choose simply a background color.

8. Once you designed the HTML page, click “Create”.

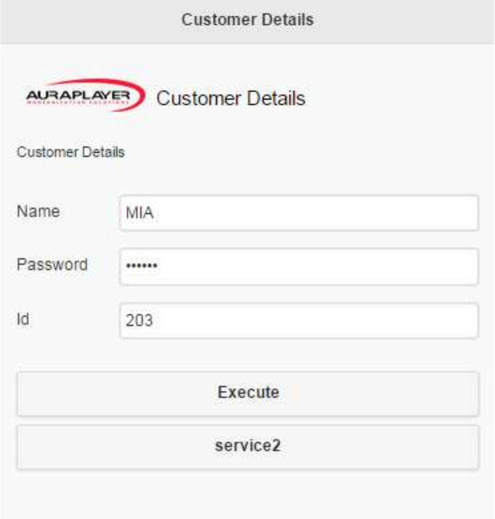


9. Go back to the 'HTML Page Manager' to see your newly designed HTML page.

Running an HTML Page with the created Business Process

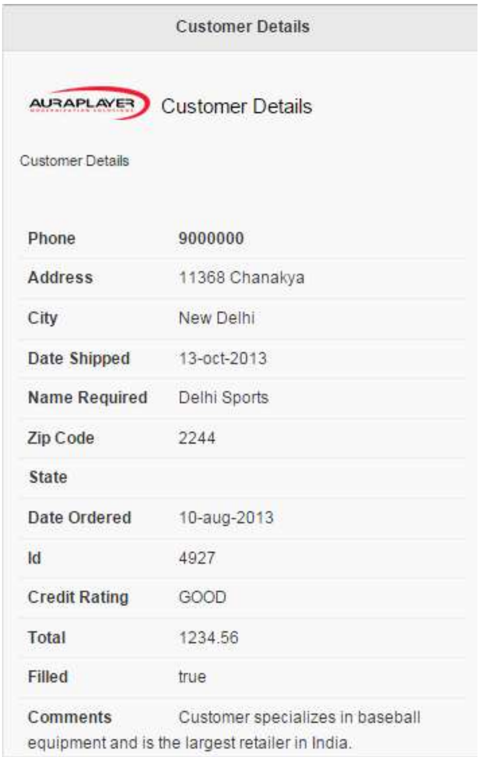
After creating your HTML page, you can now visually test your Service.

1. Enter your input parameters then click on the "Execute" button



Name	MIA
Password	••••••
Id	203

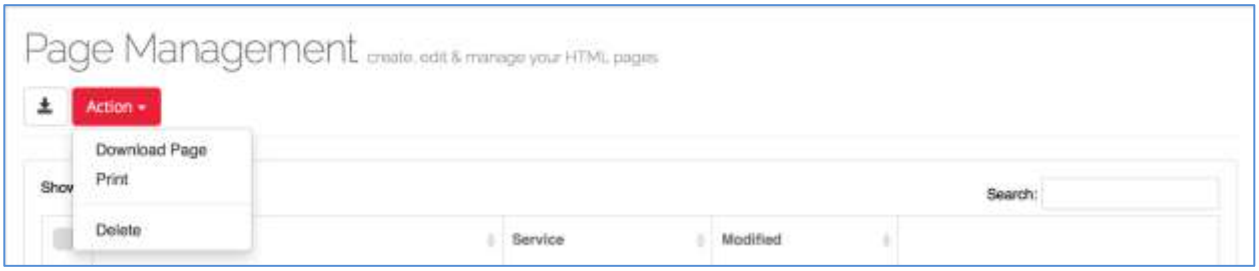
2. You will then see the response page with your service output values.



Phone	9000000
Address	11368 Chanakya
City	New Delhi
Date Shipped	13-oct-2013
Name Required	Delhi Sports
Zip Code	2244
State	
Date Ordered	10-aug-2013
Id	4927
Credit Rating	GOOD
Total	1234.56
Filled	true
Comments	Customer specializes in baseball equipment and is the largest retailer in India.

HTML Page Manager

On the "Page Manager" page, you can upload, download and delete HTML files. Created either using the visualization wizard or creating a snapshot during the recording process. The Page Manager allows the user to manage and edit the created HTML pages.



The components of the Pages Management toolbar are:

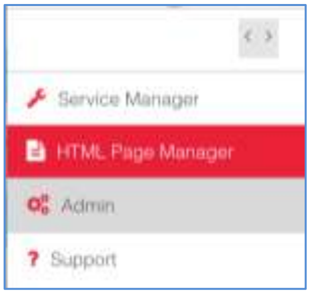
Upload:	Upload a newly selected HTML file(s) to the server
Action:	All actions will be performed in batch by checking the check box next to the HTML page.
Download Page:	Download all selected HTML files as a zip file
Print:	Print selected page
Delete:	Deletes selected HTML files. The .orp file (business logic file) that is attached to service will not be deleted.

Next to the page name there is a separate menu. This menu gives you all the toolbar actions seen above as well as:

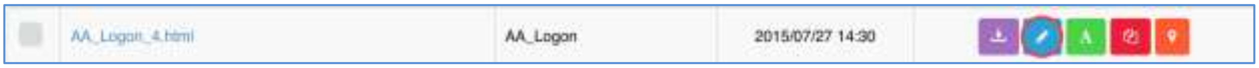
Rename	Allows you to rename an existing HTML file
Edit	Edit an existing HTML file. See “Editing HTML Feature” section below.
Duplicate:	Duplicate the file
Show Map (In Beta)	This allows you to see a flowchart mapping of the connections between HTML pages

Editing an HTML Page

1. To edit an HTML file, select the “HTML Page Manager” from the Main Menu.



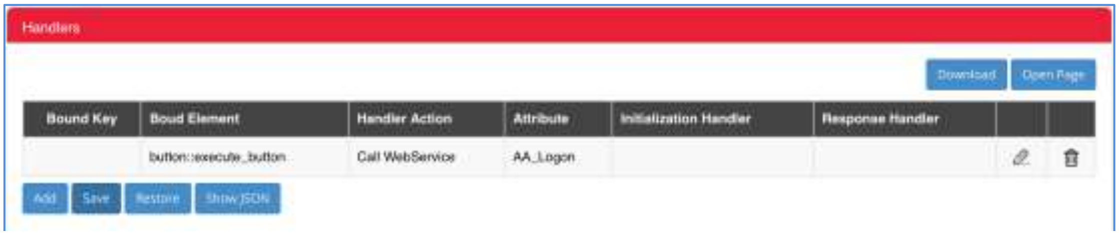
2. Click on the “Edit” icon on the left of the HTML page



□	AA_Logon_4.html	AA_Logon	2015/07/27 14:30	

Note: If you can't find the file you wish to edit, it might have not been uploaded to the server, use the "Upload" icon to find the HTML file and upload it manually.

3. On the Edit page, the first section is “Handlers”. They are the actions of the page.



Handlers

Download | Open Page

Bound Key	Boud Element	Handler Action	Attribute	Initialization Handler	Response Handler		
	button::execute_button	Call WebService	AA_Logon				

Add | Save | Restore | Show JSON

The Handler’s table has the following components:

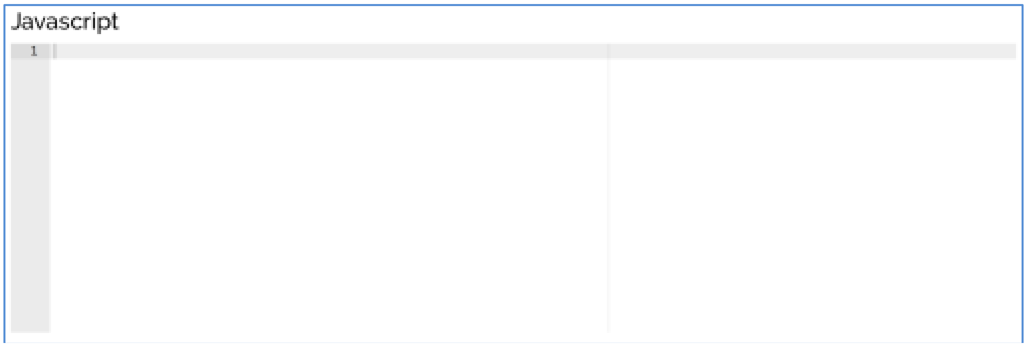
Bound Key:	The key on the keyboard that will trigger the action to occur
Bound Element:	The action will only occur on this specifically designated element
Handler Action:	The Handler's action type. For example: calling a service, calling a function, navigating to a designated page, etc...
Attribute:	The handler's action name. The name of the action that will be triggered.
Initialization Handler:	Initialization handler is a callback being called prior to calling the web service. In this callback you can set parameters, run actions (for example start spinner), write business logic in it, or decide if you want to run the web service or NOT.
Response Handler:	The handler responsible for the response returned from the service. This can be used to analyze the data, perform error handling or other functions on the Service results.

Delete:	Deletes the action

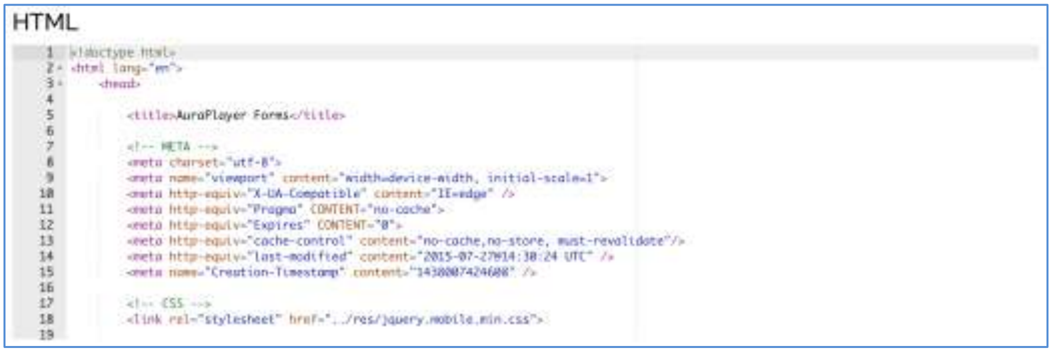
Button menu

Add:	Add a new handler by clicking the "Add" button
Save:	To save all changes, click the “Save” button
Restore:	To go back to the default settings, click “Restore”
Show JSON:	To view the JSON code/response generated

The JavaScript section allows you to create customized JavaScript functions that can be called using action handlers. See the adding handlers section below.



The HTML section allows you to custom edit the HTML code of the page



```
<!doctype html>
<html lang="en">
    <head>

        <title>AuraPlayer Forms</title>

        <!-- META -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv="Pragma" CONTENT="no-cache">
        <meta http-equiv="Expires" CONTENT="0">
        <meta http-equiv="cache-control" content="no-cache,no-store, must-revalidate"/>
        <meta http-equiv="last-modified" content="2015-07-27T14:30:24 UTC" />
        <meta name="Creation-Timestamp" content="1438007424608" />

        <!-- CSS -->
        <link rel="stylesheet" href="../res/jquery.mobile.min.css">

```

Note: Always remember to Click “Save” to save the changes made to your HTML page

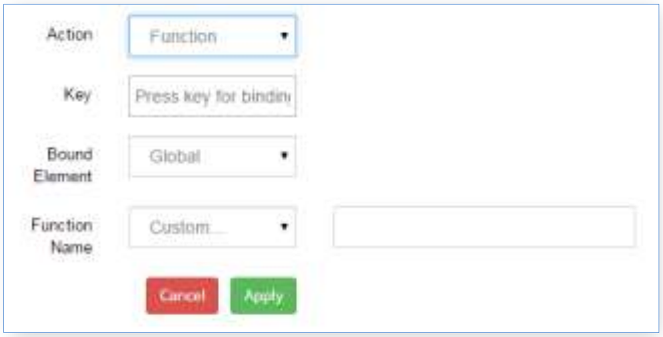
Adding Page Handlers to an HTML Page

Actions can be created to be triggered by an HTML page using the "Create Handlers Wizard". The wizard automatically generates the JavaScipt / HTML code.

1. Click "Add" to create a new handler
2. Choose the selected action

The actions that can be added as a handler are: “Function”, “Service”, “Navigate”, and “Service (On Load)”.

1. **The "Function" Action**
   The function action are JavaScript functions that can be triggered by a keyboard key or a button to perform the action.



Action: Function ▾

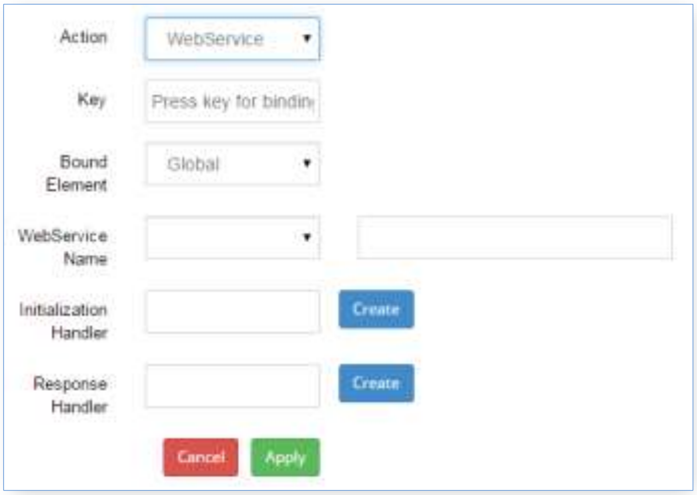
Key: Press key for bindin

Bound Element: Global ▾

Function Name: Custom... ▾

Key:	To set the key click in the "key" field and type on the desired keyboard key. Note: If the bound element is a button on the HTML page, leave the key field blank
Bound Element:	The Bound Element is the element on the HTML page that will be attached to the action. 1. **Global:** Global will cause the action to be triggered when the key is pressed anywhere on the page. 2. **Specific field/element:** Specifying a specific field/element on the page will cause the action to be triggered when the cursor is in a specific field. **Example:** If the bound element is set to “Global”, and the key is "Tab", pressing the Tab key anywhere on the HTML page will perform the function. Otherwise if the bound element is the "Customer ID" only typing Tab when the cursor is in the Customer ID will trigger the action.
Function Name:	The function name can already exist in the service manager or be customized in the Javascript editor

2. **The "WebService" action:**
   The action calls a designated Webservice.



Action: WebService

Key: Press key for bindin

Bound Element: Global

WebService Name

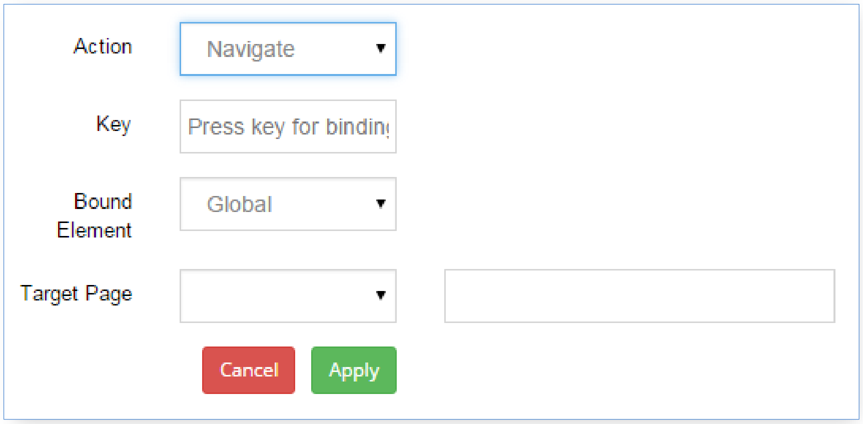
Initialization Handler

Response Handler

Key:	To set the key, click in the "key" field and type on the desired keyboard key. Note: If the bound element is a button on the HTML page, leave the key field blank
Bound Element:	The Bound Element is the element on the HTML page that will be attached to the action. 1. **Global:** Global will cause the action to be triggered when the key is pressed anywhere on the page. 2. **Specific field/element:** Specifying a specific field/element on the page will cause the action to be triggered when the cursor is in a specific field. **Example:** If the bound element is set to "Global", and the key is "Tab", pressing the Tab key anywhere on the HTML page will perform the function. Otherwise if the bound element is the "Customer ID" only typing Tab when the cursor is in the Customer ID will trigger the action.
Service Name:	The name of the service you wish to call.
Response Handler:	This can be used to analyze the Service response, perform error handling or extra functions. When you click "create", a template handler is written in the JavaScript editor below, where you can customize your response handler

3. The "Navigate" action:

Navigate Directs the user to a new web page upon action.



Action: Navigate

Key: Press key for bindin

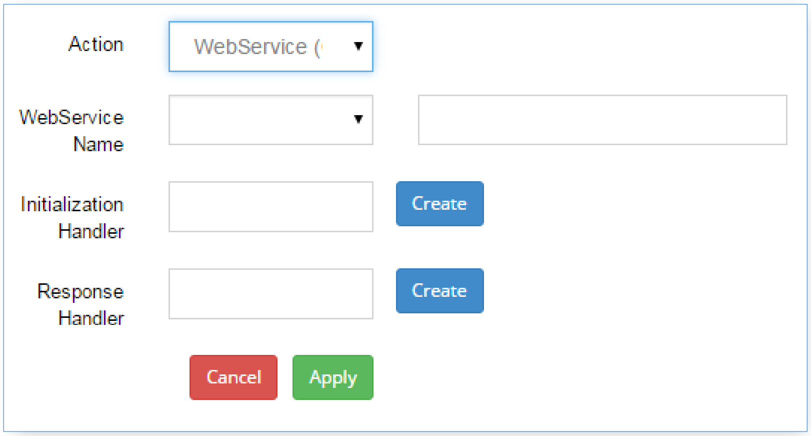
Bound Element: Global

Target Page

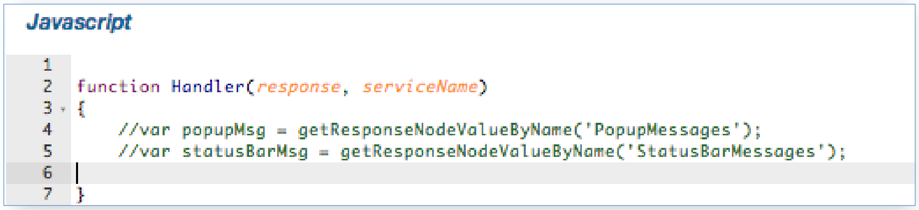
Key:	To set the key, click in the "key" field and type on the desired keyboard key. Note: If the bound element is a button on the HTML page, leave the key field blank
Bound Element:	The Bound Element is the element on the HTML page that will be attached to the action. 1. **Global:** Global will cause the action to be triggered when the key is pressed anywhere on the page. 2. **Specific field/element:** Specifying a specific field/element on the page will cause the action to be triggered when the cursor is in a specific field. **Example:** If the bound element is set to "Global", and the key is "Tab", pressing the Tab key anywhere on the HTML page will perform the function. Otherwise if the bound element is the "Customer ID" only typing Tab when the cursor is in the Customer ID will trigger the action.
Target Page:	The name of the HTML page that will be called on performing the action. This target page can be a service on the Service Manager server or any other desired web page.

4. The Service (On Load):

Service (On Load) triggers the calling of a Service upon loading of the page


Action: WebService (

WebService Name:	The name of the service you wish to call.
Initialization Handler:	Initialization handler is a callback being called prior to calling the webservice. In this callback you can set parameters, run actions (for example start spinner), write business logic in it, or decide if you want to run the web service or NOT.
Response Handler:	This can be used to analyze the response received from the Service, perform error handling or other JavaScript functions. When you click "create", a template handler is given in the JavaScript editor. Here, you can customize your response handler.



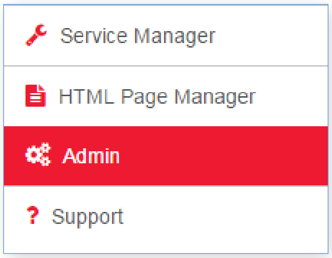
```

function Handler(response, serviceName)
{
    //var popupMsg = getResponseNodeValueByName('PopupMessages');
    //var statusBarMsg = getResponseNodeValueByName('StatusBarMessages');

}
```

ORP File Manager (Advanced)

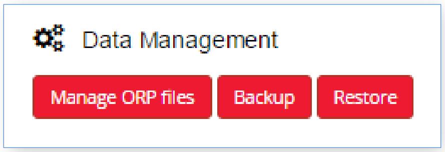
File Manager Toolbar

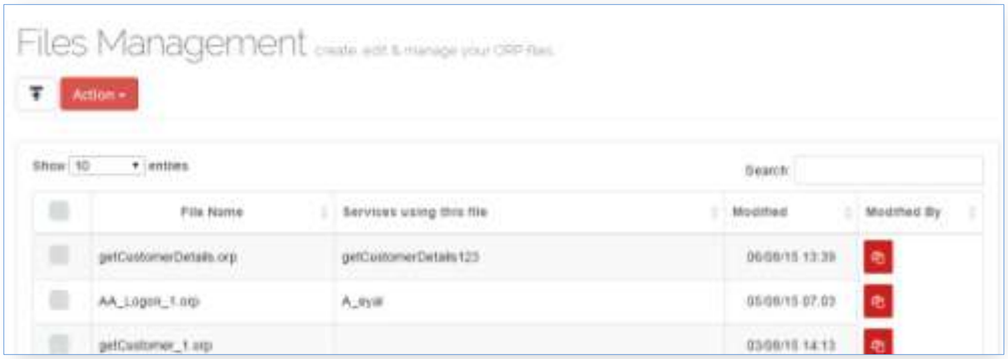
Admin -> Manage ORP Files

To manage the .orp files (AuraPlayer's Oracle Forms business process files) saved on the server, click "Admin" on the Main Menu.



In the Admin page, click on the "Manage ORP files" button.





	File Name	Services using this file	Modified	Modified By
□	getCustomerDetails.orp	getCustomerDetails123	06/08/15 13:39	
□	AA_Logon_1.orp	A_eyal	05/08/15 07:03	
□	getCustomer_1.orp		03/08/15 14:13	

The Files toolbar has the following icons:

Upload:	Enables you to upload a new .orp file(s) to the server
Actions:	Allows you to manipulate all ORP files in bulk by checking the check box
Download:	Downloads all selected .orp files to a zip file
Delete:	Deletes the selected .orp files. Any .orp file that is attached to an existing service can't be deleted.
Service:	Allows you to create a new service from the selected .orp file. You can create multiple services from a single .orp file

When uploading the .orp files to the Service Manager, the files are saved on the server. When deleting a service from the Service Manager the .orp file that is was created with, will not be deleted.

The "Files Management" list has the following columns:

File Name:	This is the name of the ORP file as it can be found on the AuraPlayer server machine.
Services using this file:	Here you will see the Services that use the specific ORP file. This will also allow you to see the unused files. Meaning, if a service was deleted, you can view the orphaned file here (an .orp file without any services).
Modified:	This is the modified date of the file.

Create a new service from an ORP file

There are 2 ways to create a new service from the File Manager.

1. Click on the "Create Service" button next to any of ORP files in the list.

2. Create a new service using the "Upload" button.

   This will import an ORP file that was generated from another AuraPlayer service manager or from the AuraPlayer client server toolbar.

   a. Click on the “Upload” icon and choose the file you wish to upload.
   b. Once uploaded, click the "Create Service" button.
   c. The Edit Service page will appear where you can do the following:
      i. Edit Page Data
      ii. Edit Request Page Input Parameters
      iii. Edit Response Page Output Parameters

   Please see the "Edit the Service" section above for step by step instructions.