

The Educational Conference For Oracle Technology Users

ODTUG
Kscope23
aurora, co june 25-29



Interested in volunteering?
Scan the code to sign up



Don't Forget To
Fill Out Your Evals

Building Professional APEX Plug-ins Explained

ODTUG KScope 2023

Dimitri Gielis - dgielis@united-codes.com - @dgielis - dgielis.com

United Codes - @united_codes - united-codes.com



Dimitri Gielis

- Founder & CEO of **United Codes**
- 23+ years of Oracle Experience (OCP & APEX Certified)
- Oracle ACE Director
- “APEX Developer of the year 2009” by Oracle Magazine
- “Oracle Developer Choice award (ORDS)” in 2015
- Author Expert **Oracle APEX**
- Presenter at Conferences



www.united-codes.com



APEX Office Print



APEX Media Extension



APEX Project Eye



Plug-ins Pro



APEX Office Edit

How do you start?

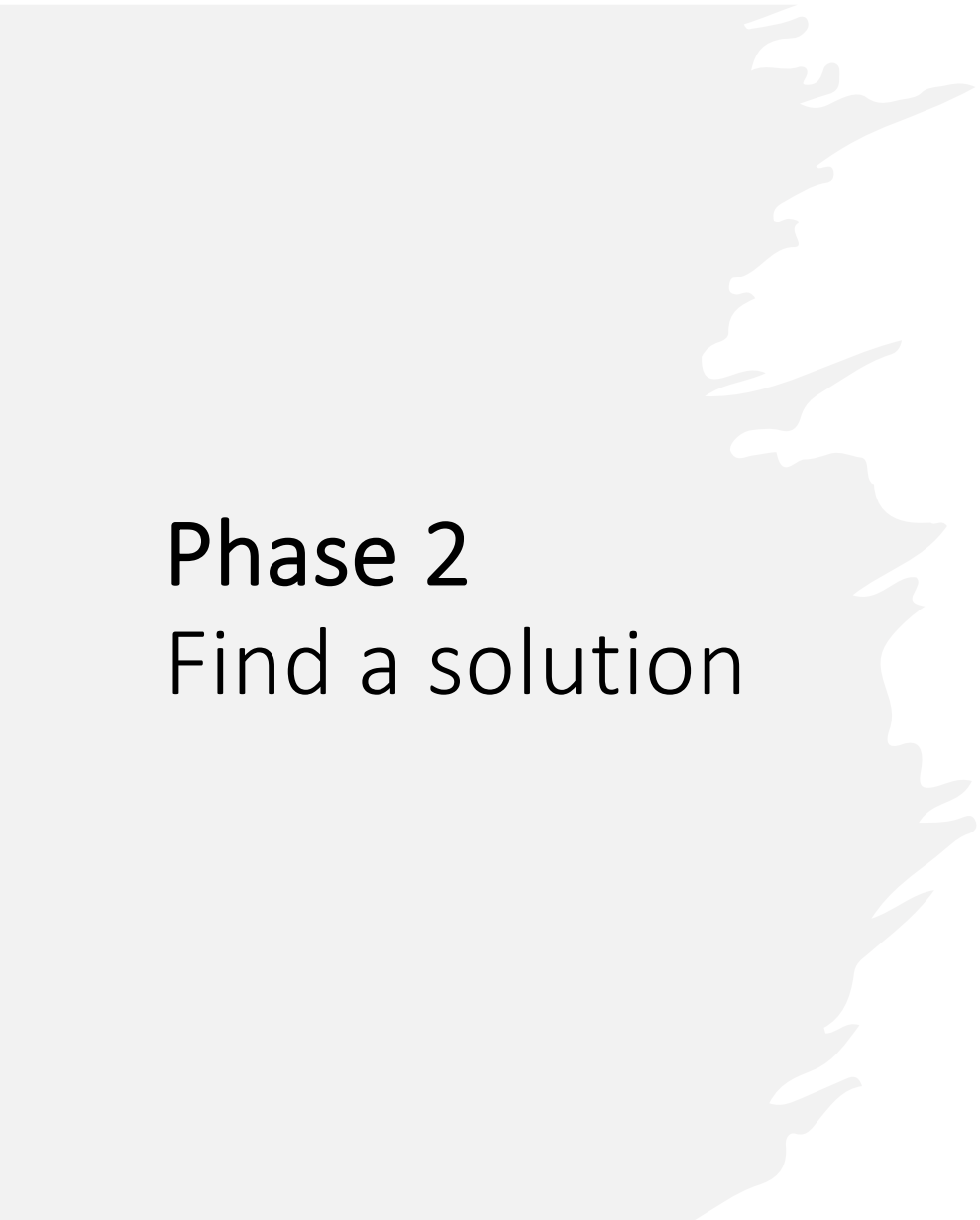
- Phase 1: Identify a missing feature
- Phase 2: Find a solution
- Phase 3: Initial draft
- Phase 4: Integrate into APEX
- Phase 5: APEX Plug-in
UC Standards & Best Practices
- Phase 6: Sample app & Documentation



Phase 1
Identify a
missing feature

PDF manipulation in APEX

- Annotate
- Insert
- Edit
- Measure
- Forms
- Layers
- Measurements
- Digital Signatures
- Safe back to DB



Phase 2

Find a solution

JavaScript libraries

- PDF-LIB
- PDFKit
- PSPDFKIT
- PDFjs
- PDFTron (Apyrse)



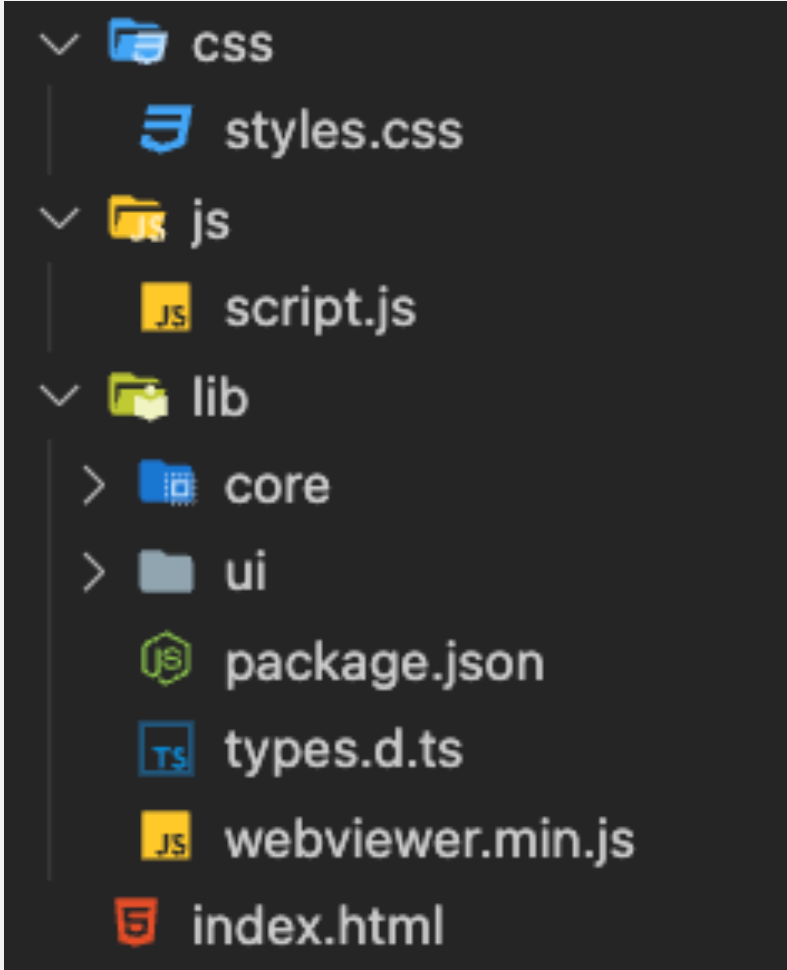
Phase 3

Initial draft

Validate your solution

- Works in plain HTML
 - Test base functionality
 - Does it work as expected?
 - Additional functionality
 - Need REST API
-
- Supported & Maintained
 - License Fee

Plain HTML Files



```

└─ css
   └─ styles.css
└─ js
   └─ script.js
└─ lib
   ├── core
   ├── ui
   ├── package.json
   ├── types.d.ts
   ├── webviewer.min.js
   └─ index.html

```

A file explorer view showing a project structure. The root directory contains three subdirectories: `css`, `js`, and `lib`. The `css` directory contains `styles.css`. The `js` directory contains `script.js`. The `lib` directory contains `core`, `ui`, `package.json`, `types.d.ts`, `webviewer.min.js`, and `index.html`. Each item is preceded by a collapse/expand icon (a downward arrow for folders, a rightward arrow for files).

Plain HTML

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="lib/webviewer.min.js"></script>
  <link rel="stylesheet" href="css/styles.css">
</head>

<body>
  <div id="viewer"></div>
  <script src="js/script.js"></script>
</body>

</html>
```

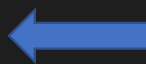
Most important parts in index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="lib/webviewer.min.js"></script>
  <link rel="stylesheet" href="css/styles.css">
</head>

<body>
  <div id="viewer"></div>
  <script src="js/script.js"></script>
</body>

</html>
```



JS library code

Most important parts in index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="lib/webviewer.min.js"></script>
  <link rel="stylesheet" href="css/styles.css">
</head>

<body>
  <div id="viewer"></div>
  <script src="js/script.js"></script>
</body>

</html>
```



Styling

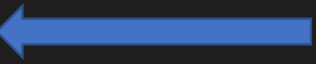
Most important parts in index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="lib/webviewer.min.js"></script>
  <link rel="stylesheet" href="css/styles.css">
</head>

<body>
  <div id="viewer"></div>
  <script src="js/script.js"></script>
</body>

</html>
```



Necessary HTML

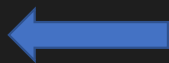
Most important parts in index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="lib/webviewer.min.js"></script>
  <link rel="stylesheet" href="css/styles.css">
</head>

<body>
  <div id="viewer"></div>
  <script src="js/script.js"></script>
</body>

</html>
```



Custom code

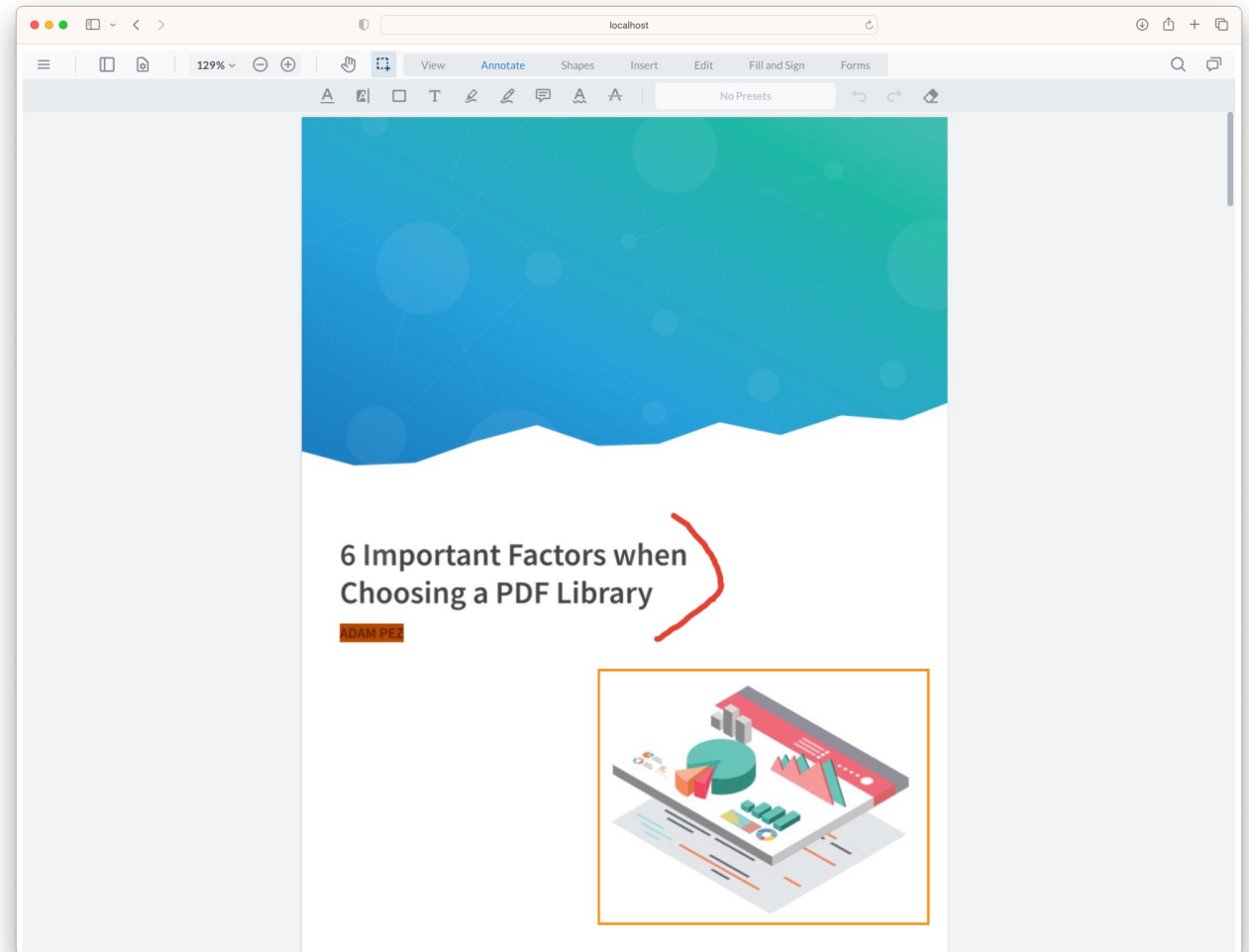
styles.css

```
#viewer {  
  height: 100%;  
  width: 100%;  
}  
  
body {  
  height: 100vh;  
}
```

script.js

```
WebView(  
  {  
    // Path below needs to be dynamic  
    path: "lib",  
    initialDoc: "https://pdftron.s3.amazonaws.com/downloads/pl/demo-annotated.pdf",  
  },  
  document.getElementById("viewer")  
)  
.then((instance) => {  
  if (navigator.appVersion.indexOf("MSIE") > -1) {  
    return alert("This WebView version is not supported in your browser. ");  
  }  
});
```

Plain HTML Result





Phase 4

Integrate into APEX

Backwards compatibility

- Develop in oldest APEX version – we chose 19.2

HTML → APEX → Plug-in

Why create an APEX plug-in?

- Extend an APEX application with custom functionality
- Helper functions for plug-in
- Highly customizable
- Feels like native APEX
- Reusable

Types of plug-ins

Item

Dynamic
Action

Region

Process

Authorization
Scheme type

Authentication
Scheme type

REST Data
Source

Template
Component (*)

(*) New in APEX 23.1

Integrate into
APEX

HTML → APEX → Plug-in

1. Create Application
2. Create Plug-in
3. Upload files
4. Implement on page

Create APEX Application

↑ Application 165

Application 165 - PDF Region Pro

Edit Application Properties



Run Application



Supporting Objects



Shared Components



Utilities



Export / Import

Q

Go



Actions

Create Page



0 - Global Page - Desktop



1 - Home



9999 - Login Page

1 - 3

About

Use the Application home page to run, edit, import, export, copy, or delete an application. Select a page to edit the page properties, or click **Create Page** to add a page to the application.
[Learn More ...](#)

Tasks

Delete this Application

Copy this Application

Recently Edited Pages

0. Global Page - Desktop

9999. Login Page

1. Home

Create Plug-in



Application 165

1

Application 165 - PDF Region Pro

Edit Application Properties

About

Use the Application home page to run, edit, import, export, copy, or delete an application. Select a page to edit the page properties, or click **Create Page** to add a page to the application.
[Learn More ...](#)

Tasks

Delete this Application >

Copy this Application >

Recently Edited Pages

0. Global Page - Desktop

9999. Login Page

1. Home

1 - 3

Run Application

Supporting Objects

Shared Components

Utilities

Export / Import

0 - Global Page - Desktop

1 - Home

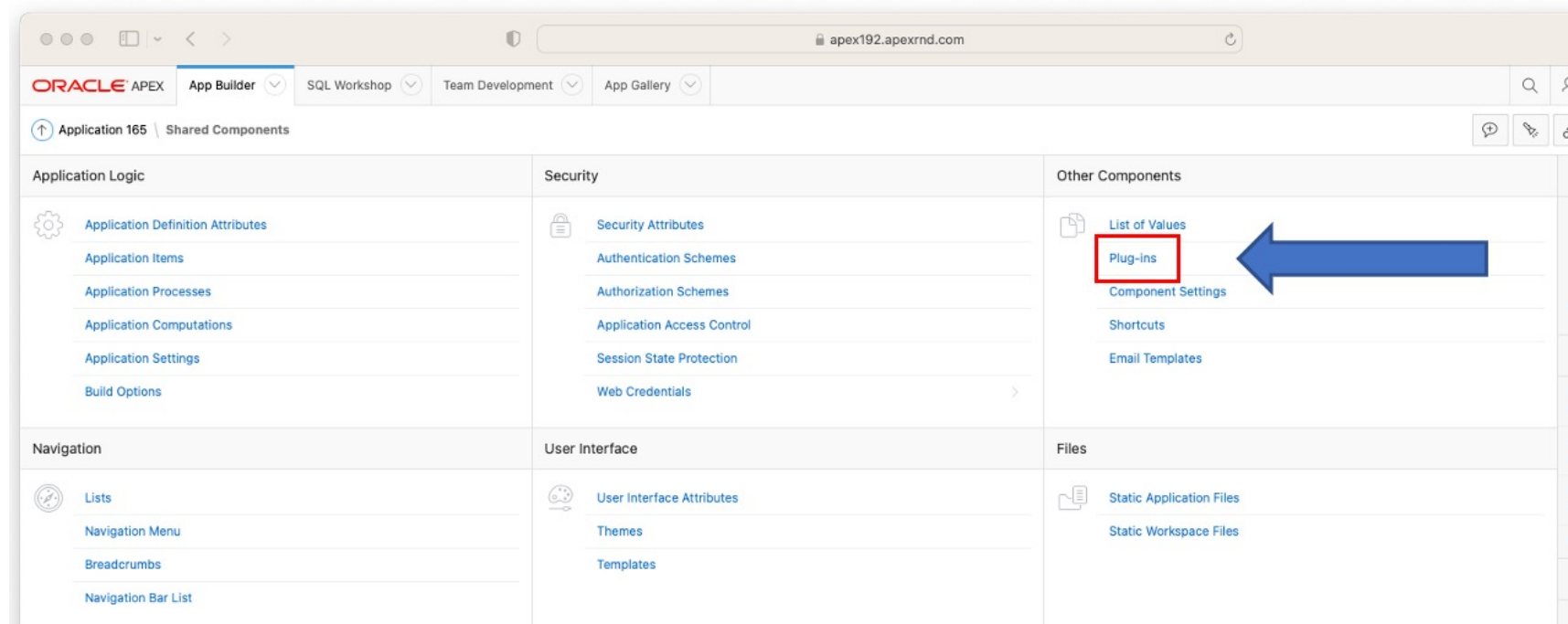
9999 - Login Page

Go

Actions

Create Page >

Create Plug-in



Create Plug-in

The screenshot displays the Oracle APEX web interface. At the top, the browser address bar shows 'apex192.apexrnd.com'. The Oracle APEX navigation bar includes tabs for 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. The main breadcrumb trail reads 'Application 165 \ Shared Components \ Plug-ins'. Below this, there are tabs for 'Plug-ins', 'Utilization', and 'History'. A search bar is present with a 'Go' button and an 'Actions' dropdown. To the right of the search bar are buttons for 'Reset', 'View Plug-in Repository', 'Import >', and a 'Create >' button which is highlighted with a red rectangular box. A large blue arrow points upwards towards the 'Create >' button. The main content area shows a magnifying glass icon and the text 'No plug-ins found.'. On the right side, a sidebar contains a 'Plug-ins' section with a description: 'A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, dynamic action process, authentication scheme and authorization scheme types in your application.' Below this are sections for 'Tasks' (with an 'Export Plug-in' link) and 'Recently Edited' (showing 'No plug-ins found').

ORACLE APEX App Builder SQL Workshop Team Development App Gallery

Application 165 \ Shared Components \ Plug-ins

Plug-ins Utilization History

Search Go Actions

Reset View Plug-in Repository Import > **Create >**

No plug-ins found.

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, dynamic action process, authentication scheme and authorization scheme types in your application.

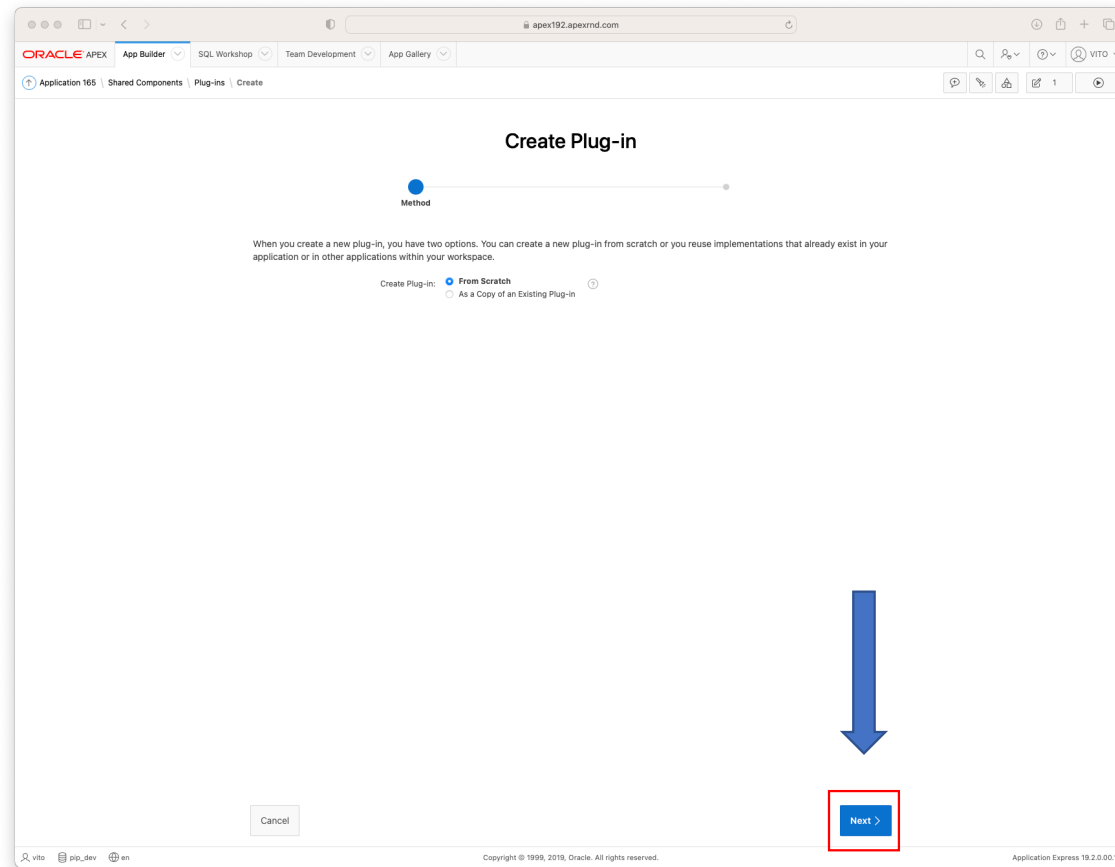
Tasks

[Export Plug-in](#)

Recently Edited

No plug-ins found

Create Plug-in



Create Plug-in: Give your plug-in a name and type

The screenshot shows the Oracle APEX App Builder interface for creating a plug-in. The browser address bar shows `apex192.apexrnd.com`. The top navigation bar includes tabs for **ORACLE APEX**, **App Builder**, **SQL Workshop**, **Team Development**, and **App Gallery**. The breadcrumb trail indicates the current location: **Application 165** > **Shared Components** > **Plug-ins** > **Create / Edit**. The main heading is **Plug-in:**, with **Cancel** and **Create Plug-in** buttons. Below this is a table with columns: **Show All**, **Name**, **Source**, **Callbacks**, **Supported for**, **Standard Attributes**, **Information**, **Help Text**, and **Comments**. The **Name** section is highlighted with a red box and contains three fields: *** Name** (text input), *** Internal Name** (text input), and *** Type** (dropdown menu set to **Item**). A blue arrow points to the **Name** section. The **Source** section is currently empty. The **PL/SQL Code** section at the bottom is also empty. On the right side, there is a **Plug-ins** sidebar with a description: "A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application."

Create Plug-in: Give your plug-in a name and type

The screenshot shows the Oracle APEX App Builder interface. The browser address bar displays `apex192.apexrnd.com`. The top navigation bar includes tabs for **App Builder**, **SQL Workshop**, **Team Development**, and **App Gallery**. The breadcrumb trail indicates the current location: **Application 165 > Shared Components > Plug-ins > Create / Edit**. The main heading is **Plug-in:**, with **Cancel** and **Create Plug-in** buttons. Below the heading is a table with columns: **Show All**, **Name**, **Source**, **Callbacks**, **Supported for**, **Standard Attributes**, **Information**, **Help Text**, and **Comments**. The **Name** section is highlighted with a red box, containing three fields: **Name** (text input with value `PDF Region Pro`), **Internal Name** (text input with value `UC PDF Region Pro`), and **Type** (dropdown menu with value `Region`). A blue arrow points to the **Name** section. The **Source** section is visible below the **Name** section, showing a **PL/SQL Code** editor with a line number `1`. On the right side, there is a **Plug-ins** sidebar with a description: "A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application."

Create Plug-in

Oracle APEX App Builder | SQL Workshop | Team Development | App Gallery

Application 165 | Shared Components | Plug-ins | Create / Edit

Plug-in: Cancel Create Plug-in

Name

Name: PDF Region Pro ⓘ

Internal Name: UC PDF Region Pro ⓘ

Type: Region ⓘ

Source

PL/SQL Code ⓘ

1

☐ Do not validate PL/SQL code (parse PL/SQL code at runtime only).

Callbacks

Render Procedure/Function Name: ⓘ

AJAX Procedure/Function Name: ⓘ

Supported for

User Interface: ☒ Desktop ⓘ ☐ Mobile

Standard Attributes

Attributes: ☐ Region Source supports different Data Sources ⓘ ☐ Region Source is SQL Statement [Deprecated] ⓘ ☐ Region Source is Plain Text ⓘ

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

Create Plug-in: upload necessary files

Plug-in: PDF Region Pro

Cancel

Delete

Apply Changes

Show All	Name	Subscription	Source	Callbacks	Supported for	Standard Attri...	Standard Attri...	Custom Attrib...	Files	File URLs to L...	Events	Information	Help Text	Comments
Files														
File Prefix No Files found.														
Created: 60 minutes ago - VITO Updated: 60 minutes ago - VITO														

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

☐ Return to page

Tasks

Export Plug-in

Utilization



Create Plug-in: upload necessary files

Upload File(s)

Use this page to associate files like images, CSS- or Javascript files with your plug-in. To associate a file with your plug-in, click Browse, select the file(s), and click Upload.

Use **p_plugin.file_prefix** in your plug-in code to reference a plug-in file.

Directory

* File 3 files

* File Character Set

* Unzip File ☒

Choose Files to Upload

pdf-test

css

css.zip

index.html

js

js.zip

lib

lib.zip

ZIP

3 items

3 documents - 97,5 MB

Information

Create Plug-in: Reference files



Plug-in: PDF Region Pro

Cancel

Delete

Apply Changes

Show All

Name

Subscription

Source

Callbacks

Supported for

Standard Attri...

Standard Attri...

Custom Attrib...

Files

File URLs to L...

Events

Information

Help Text

Comments

Files

Upload File

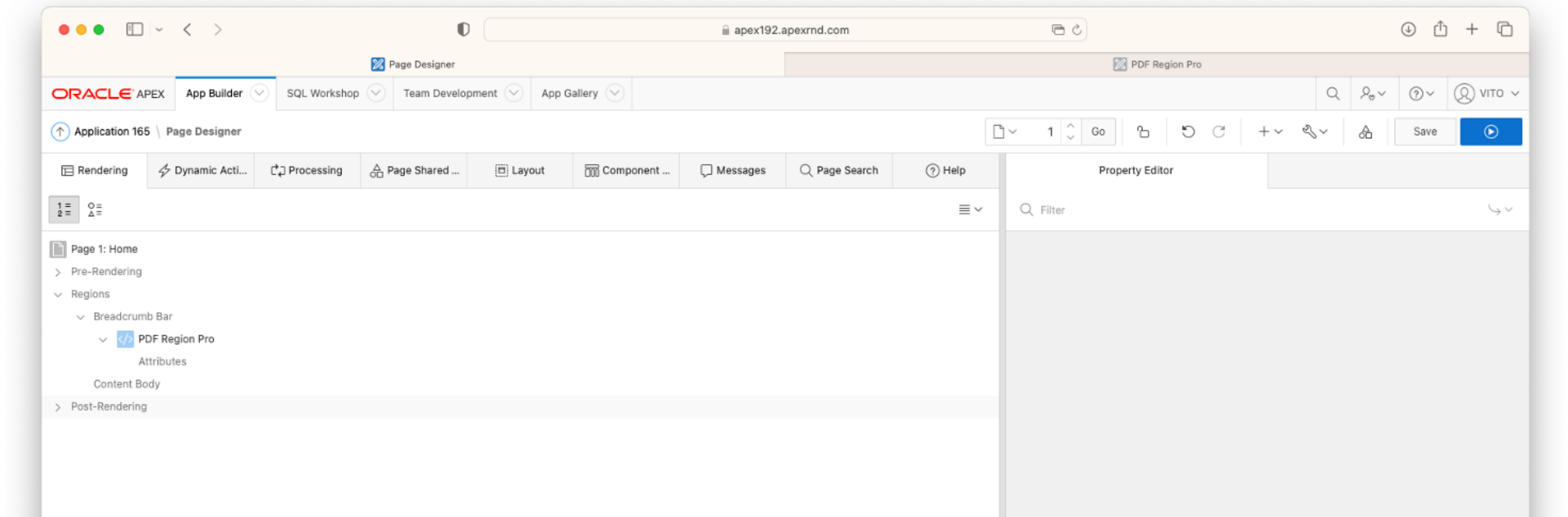
File Prefix

?

File Name ↑	Mime Type	File Size	File
css/styles.css	text/css	70	Download
js/script.js	application/x-javascript	356	Download
lib/core/20.chunk.js	application/x-javascript	3KB	Download
lib/core/21.chunk.js	application/x-javascript	4KB	Download
lib/core/AndroidContentPartRetriever.chunk.js	application/x-javascript	948	Download
lib/core/ArrayBufferPdfPartRetriever.chunk.js	application/x-javascript	623	Download
lib/core/AzurePartRetriever.chunk.js	application/x-javascript	776	Download
lib/core/CORSWorker.js	application/x-javascript	1KB	Download
lib/core/ContentEdit.chunk.js	application/x-javascript	27KB	Download
lib/core/DecryptWorker.js	application/x-javascript	14KB	Download
lib/core/ExternalAnnotationMerger.chunk.js	application/x-javascript	7KB	Download
lib/core/ExternalPdfPartRetriever.chunk.js	application/x-javascript	3KB	Download
lib/core/HttpPartRetriever.chunk.js	application/x-javascript	6KB	Download
lib/core/IOSPartRetriever.chunk.js	application/x-javascript	979	Download
lib/core/LocalPartRetriever.chunk.js	application/x-javascript	2KB	Download

1 - 15 [Next](#)

Implement on Page



Implement on Page

The screenshot displays the Oracle APEX Page Designer interface. The top navigation bar includes tabs for ORACLE APEX, App Builder, SQL Workshop, Team Development, and App Gallery. The main toolbar contains icons for file operations, undo, redo, and a blue 'Run' button highlighted with a red box and a blue arrow. The left pane shows the page structure for 'Page 1: Home', with 'PDF Region Pro' selected under the 'Regions' section. The right pane shows the configuration for the 'PDF Region Pro' component, including its title, type, sequence, and parent region.

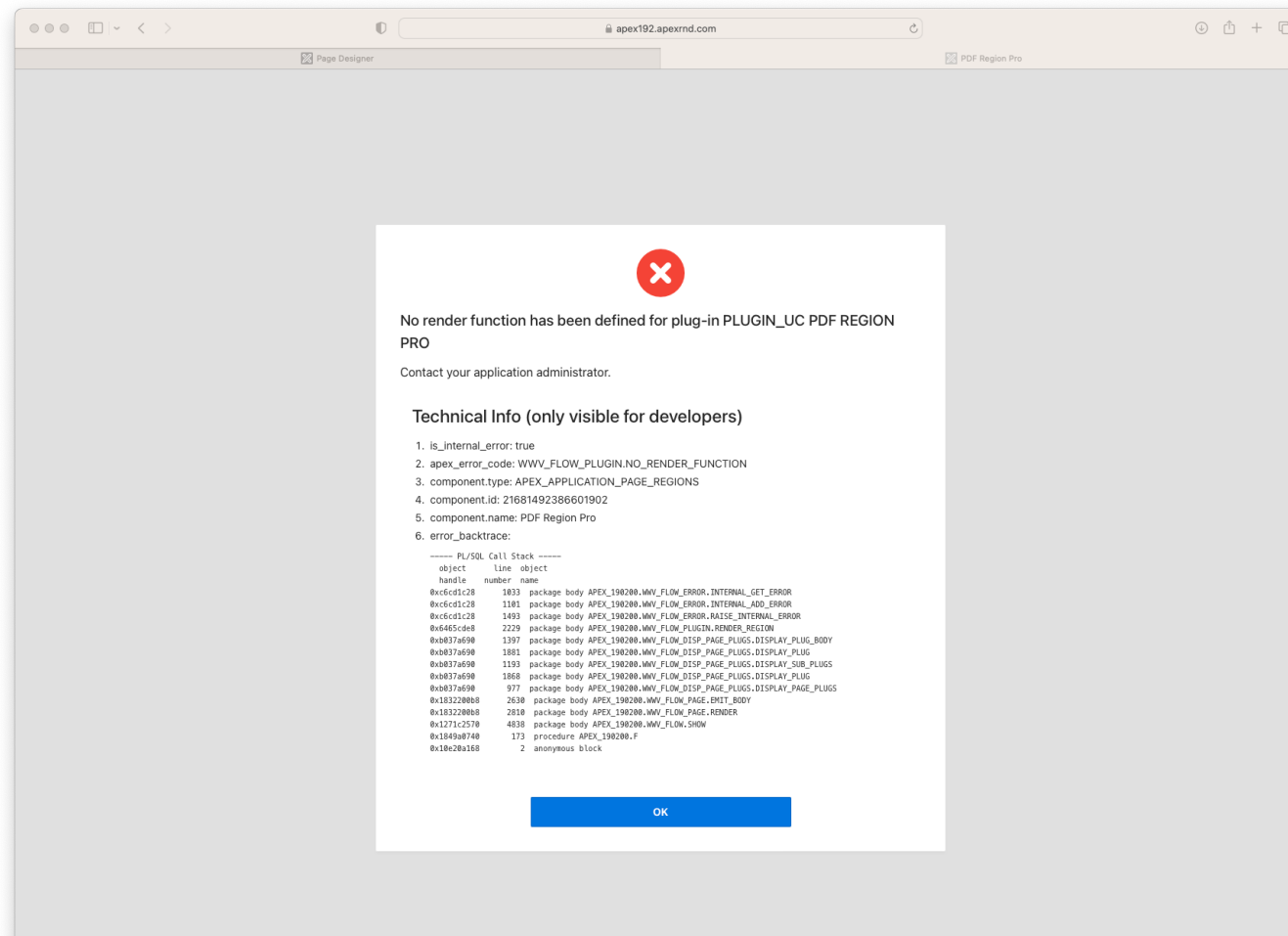
Page Structure (Left Pane):

- Page 1: Home
 - Pre-Rendering
 - Regions
 - Breadcrumb Bar
 - PDF Region Pro (Selected)
 - Attributes
 - Sub Regions
 - Content Body
 - Post-Rendering

Component Configuration (Right Pane):

Region	
Filter	
Identification	
Title	PDF Region Pro
Type	PDF Region Pro [Plug-In]
Layout	
Sequence	10
Parent Region	PDF Region Pro

Implement on Page



Core Concepts: Region Plug-in

- Attributes
 - Standard & Custom
- Callbacks
 - Render
 - Ajax
- Events

Core Concepts: Region Plug-in

- **Attributes**
 - **Standard & Custom**
- Callbacks
 - Render
 - Ajax
- Events

Attributes: Standard

- Source
- Read only
- ...

Standard Attributes

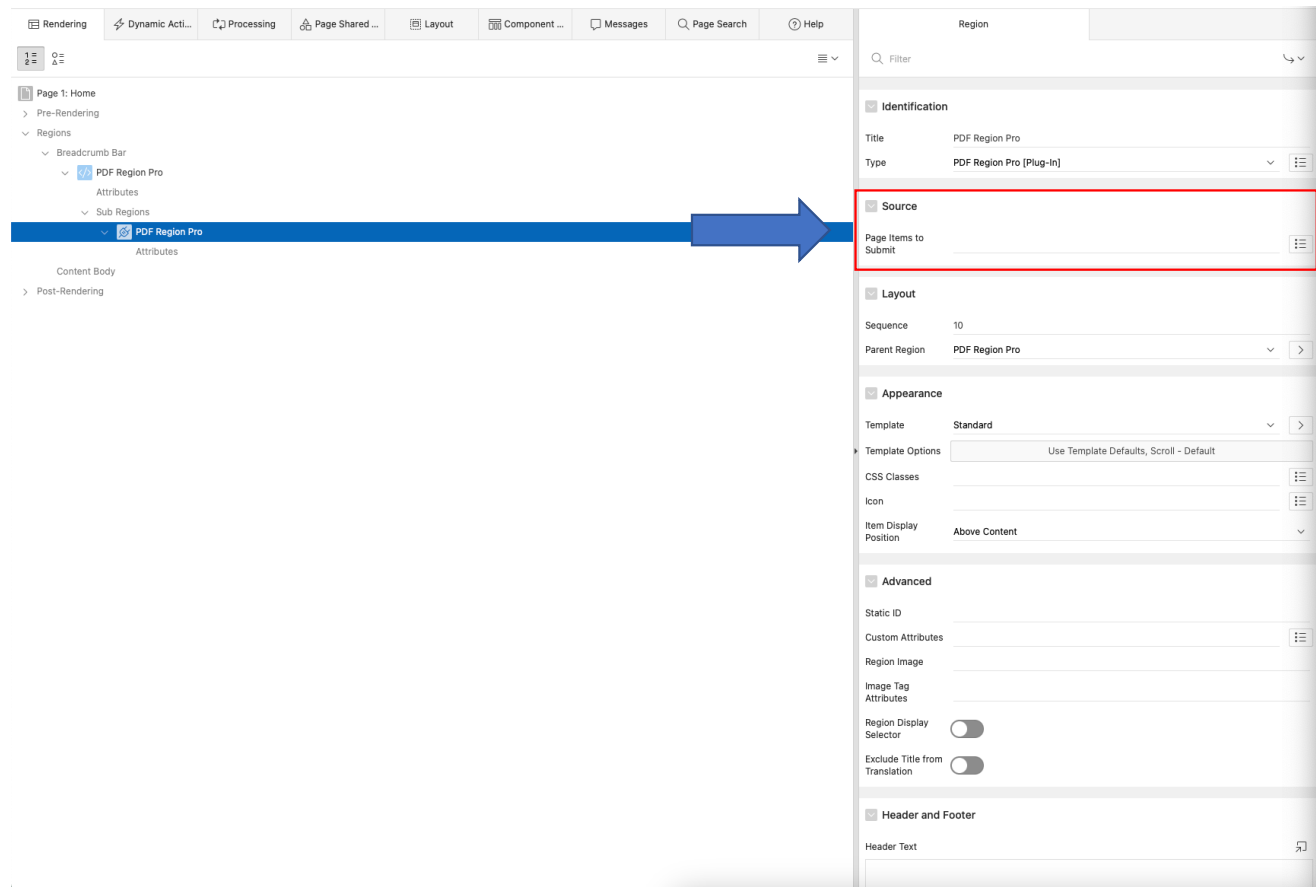
Attributes: ☐ Region Source supports different Data Sources ?
☐ Region Source is SQL Statement [Deprecated]
☐ Region Source is Plain Text
☒ Has "Page Items to Submit" Attribute
☐ Has "Number of Fetched Rows" Attribute
☐ Has "No Data Found Message" Attribute
☐ Has "Escape Special Characters" Region Attribute
☒ Has "Initialization JavaScript Code" Attribute
☐ Has "Region Columns"
☐ Has "Heading" Column Attribute
☐ Has "Heading Alignment" Column Attribute
☐ Has "Alignment" Column Attribute
☐ Has "CSS Class" Column Attribute
☒ Has "Custom Attributes" Column Attribute
☐ Has "Escape Special Characters" Column Attribute

Standard Attribute additional Meta Data

Standard Attribute	Required	Default Value	Depending on Custom Attribute
Has "Initialization JavaScript Code" Attribute	No		

Attributes: Standard

- Source
- Read only
- ...



Attributes: Custom

Custom Attributes							Add Attribute
Substitute Attribute Values ☒ ?							
Label	Scope ↑≡	Attribute	Sequence	Type	Required	Default Value	Depending on
Editor Settings	Component	2	20	Checkboxes	No		
Table Name	Component	6	2110	Text	Yes		Source
Content Column	Component	7	2120	Text	Yes		Source
Primary Key(s) Column(s)	Component	8	2115	Text	Yes		Source
Item(s) Containing Primary Key(s) Value(s)	Component	9	2000	Page Items	Yes		
Last Update Column	Component	10	2400	Text	Yes		Source
Owner Column	Component	11	2100	Text	No		Source
Filename Column	Component	12	2200	Text	Yes		Source
MIME Type Column	Component	13	2300	Text	Yes		Source
Custom PL/SQL Package Name	Component	14	2990	Text	No		
Version Column	Component	15	2500	Text	No		Source
Document Permissions	Component	21	2570	Select List	Yes	EVERYONE_CAN_READ_AND_EDIT	
Initialization PL/SQL Code	Component	24	3000	PL/SQL Code	No		
Source	Component	25	2100	Select List	Yes	DEFAULT	

Attributes: Custom

The screenshot displays the Oracle APEX editor interface. On the left, the 'Page 100: Online demo' tree shows the 'Attributes' section highlighted with a red arrow. The main workspace shows the 'Attributes' panel for the 'P100_DOCUMENT_ID' attribute. The panel is divided into 'Settings' and 'Advanced' sections.

Settings

- ☐ Disable Annotations
- ☐ Disable Shapes
- ☐ Disable Insert
- ☐ Disable Edit
- ☐ Disable Forms
- ☐ Disable Fill and Sign

Editor Settings

Item(s) Containing Primary Key(s) Value(s): P100_DOCUMENT_ID

Source: Default table

Document Permissions: Everyone can read and edit

Custom PL/SQL Package Name

Initialization PL/SQL Code

Advanced

JavaScript Initialization Code

```
function( pOptions ) {  
  pOptions.darkMode      = false;  
  pOptions.language      = "EN";  
  pOptions.zoomLevel     = 90;  
  pOptions.showSidebar   = true;  
  pOptions.activeTool    = "VIEW"  
  
  return pOptions;  
}
```

Core Concepts: Region Plug-in

- Attributes
 - Standard & Custom
- **Callbacks**
 - **Render**
 - Ajax
- Events

Callbacks

ORACLE APEX | App Builder | SQL Workshop | Team Development | App Gallery

Application 165 | Shared Components | Plug-ins | Create / Edit

Plug-in: PDF Region Pro

Show All	Name	Subscription	Source	Callbacks	Supported for	Standard Attri...	Standard Attri...	Custom Attrib...	Files	File URLs to L...	Events	Information	Help Text	Comments
Callbacks														
Render Procedure/Function Name	UC_DEMO.render ?													
AJAX Procedure/Function Name	UC_DEMO.ajax ?													
Created: 18 hours ago - VITO Updated: 17 hours ago - VITO														

Cancel Apply Changes

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

☐ Return to page ?

Tasks

- Export Plug-in >
- Utilization >

Callbacks

ORACLE APEX

App Builder

SQL Workshop

Team Development

App Gallery

Application 165

Shared Components

Plug-ins

+

🔍

🔗

📄

1

▶

✓

Action processed.

×

Plug-ins

Utilization

History

🔍

Go

🗃️

📄

Actions

Reset

View Plug-in Repository

Import

Create

Name	Type	Updated	Version	About URL	Subscribed From	Subscribers	References
PDF Region Pro	Region	Now	1.0				1

1 - 1

Plug-ins

A plug-in is an extension to the built-in types of Application Express. Use plug-ins for declarative use of new item, region, dynamic action process, authentication scheme and authorization scheme types in your application.

Tasks

Export Plug-in

Recently Edited

PDF Region Pro

Callbacks

ORACLE APEX

App Builder

SQL Workshop

Team Development

App Gallery

?

VITO

Object Browser

SQL Commands

SQL Scripts

Utilities

RESTful Services

Recently Created Tables

Recent SQL Commands

About

Schema

Create Object

Recently Created Tables	Recent SQL Commands
UC_PDF_REGION_PRO_FILES_UPLOADED 6 days ago	select VERSION, ... 4 days ago
UC_PDF_DEFAULT_FILES 12 days ago	select VERSION, ... 4 days ago
UC_PDF_DEFAULT_VERSIONS 12 days ago	CREATE OR REPLACE EDITIONABLE TRIGG... 6 days ago
UC_GRID_PROJECTS 13 days ago	CREATE OR REPLACE EDITIONABLE TRIGG... 6 days ago
UC_GRID_TASKS 13 days ago	drop TRIGGER "BI_UC_PDF_REGION_PRO_... 6 days ago
UC_PDF_REGION_PRO_EXT_FILES 2 weeks ago	CREATE OR REPLACE EDITIONABLE TRIGG... 6 days ago
UC_PDF_REGION_PRO_CUSTOM_TBL 2 weeks ago	CREATE OR REPLACE EDITIONABLE TRIGG... 6 days ago

About

Schema

Create Object

Object Browser

SQL Commands

SQL Scripts

Utilities

RESTful Services

Recently Created Tables

Recent SQL Commands

About

Schema

Create Object

Callbacks

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes tabs for 'App Builder', 'SQL Workshop' (which is active), 'Team Development', and 'App Gallery'. The 'SQL Workshop' tab is highlighted with a green underline. The 'Object Browser' is visible on the left side, showing a list of tables and views under the 'PIP_DEV' schema. A blue arrow points from the 'Object Browser' to a context menu that is open, showing options for creating database objects. The 'Package' option is highlighted with a red border.

Oracle APEX | App Builder | SQL Workshop | Team Development | App Gallery

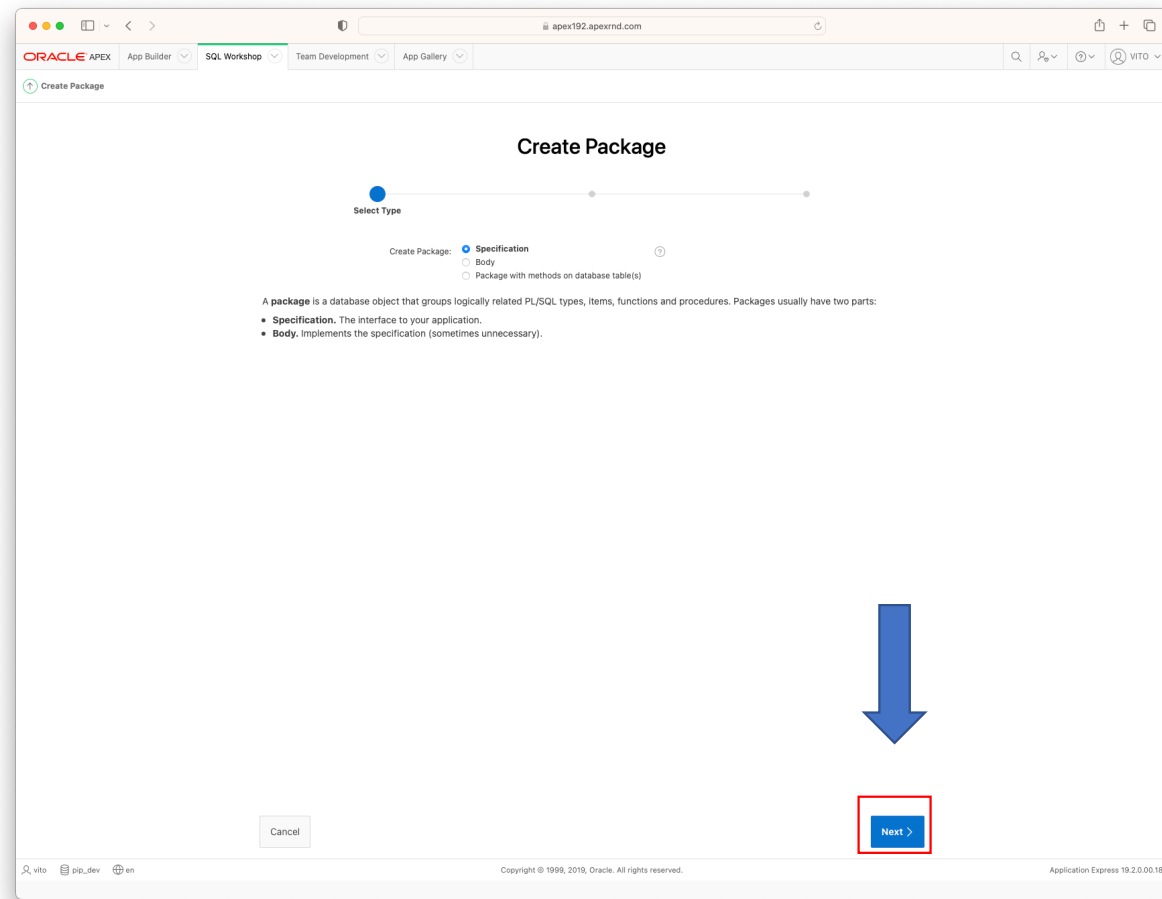
Object Browser | Schema: PIP_DEV

Tables

- AOE_API_STAGING
- AOE_BUCKET_CONFIG
- AOE_BUCKET_FILES
- AOE_CONFIG
- AOE_CUSTOM_TBL
- AOE_DEFAULT_FILES
- AOE_DEFAULT_VERSIONS
- AOE_EXT_FILES
- AOE_FILES_CUSTOM
- AOE_FILES_UPLOADED
- AOE_REST_LOGS
- DEPT
- DR\$AOE_DEFAULT_FILES_TEXT_IDX\$I
- DR\$AOE_DEFAULT_FILES_TEXT_IDX\$K
- DR\$AOE_DEFAULT_FILES_TEXT_IDX\$N
- DR\$AOE_DEFAULT_FILES_TEXT_IDX\$U
- DR\$AOE_DEFAULT_VERSIONS_TEXT_IDX\$I
- DR\$AOE_DEFAULT_VERSIONS_TEXT_IDX\$K

Table
View
Index
Sequence
Type
Package
Procedure
Function
Trigger
Database Link
Materialized View
Synonym

Callbacks



Callbacks

ORACLE APEX App Builder SQL Workshop Team Development App Gallery

Create Package

Create Package

Progress bar: [Green dot] [Blue dot] [Grey dot]

Name

* Package Name UC_DEMO ⓘ

☐ Preserve Case

< Cancel Next >

Copyright © 1999, 2019, Oracle. All rights reserved. Application Express 19.2.0.00.18

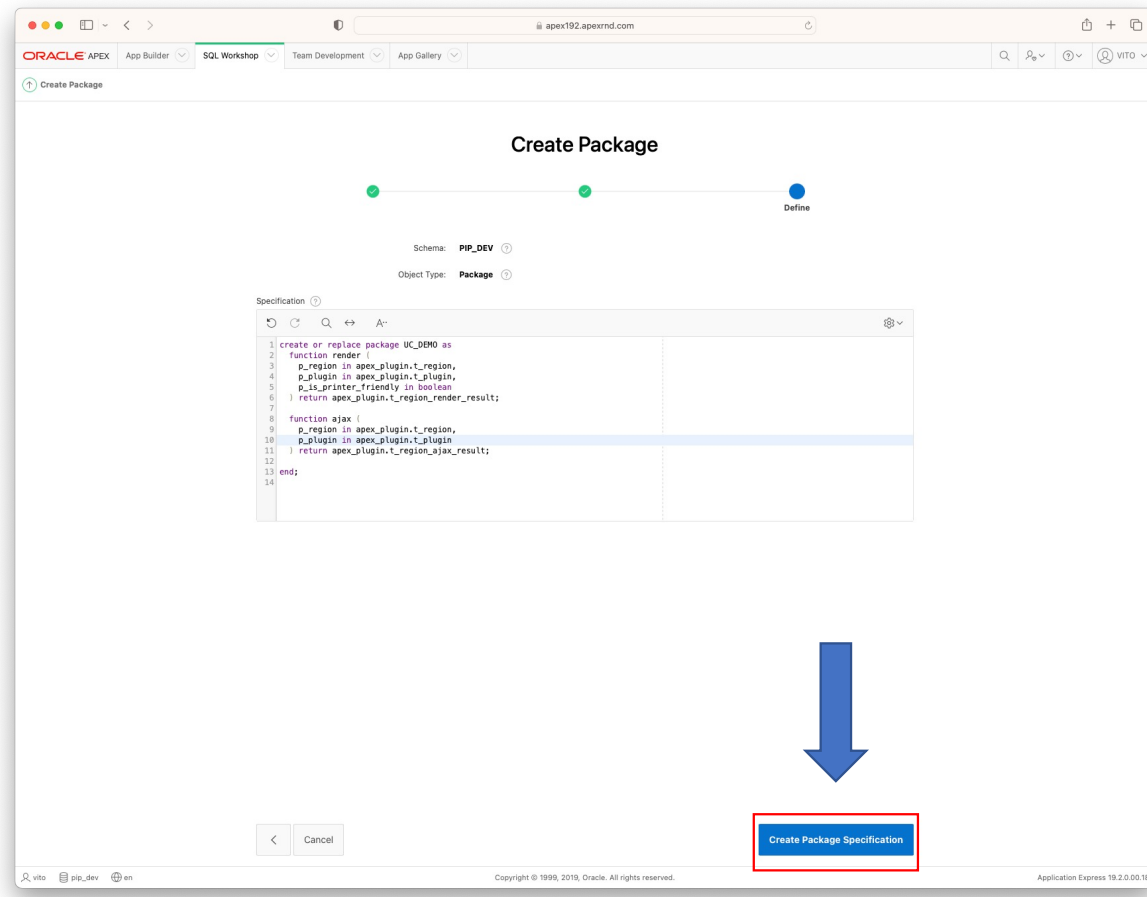
Callbacks

```
create or replace package UC_DEMO as
  function render (
    p_region in apex_plugin.t_region,
    p_plugin in apex_plugin.t_plugin,
    p_is_printer_friendly in boolean
  ) return apex_plugin.t_region_render_result;

  function ajax (
    p_region in apex_plugin.t_region,
    p_plugin in apex_plugin.t_plugin
  ) return apex_plugin.t_region_ajax_result;

end;
```

Callbacks



Callbacks

ORACLE APEX App Builder SQL Workshop Team Development App Gallery

Schema PIP_DEV

Object Browser

Packages

- AOE_APEX_CONTEXT_MENU
- AOE_CALLBACK_PKG
- AOE_CUSTOM_PKG
- AOE_EXT_AWS_PKG
- AOE_EXT_FILES_PKG
- AOE_EXT_OCI_PKG
- AOE_PERMISSION_PKG
- AOE_PLUGIN
- AOE_REST
- AOE_SAMPLE_APP_UTIL
- AOP_API22_PKG
- AOP_CONVERT22_PKG
- AOP_PLSQL22_PKG
- UC_CLIENT_SIDE_VALIDATION
- UC_CONSTRAINT_VALIDATION
- UC_CRYPTO
- UC_DEMO**
- UC_FROALA_3RD_PLUGINS

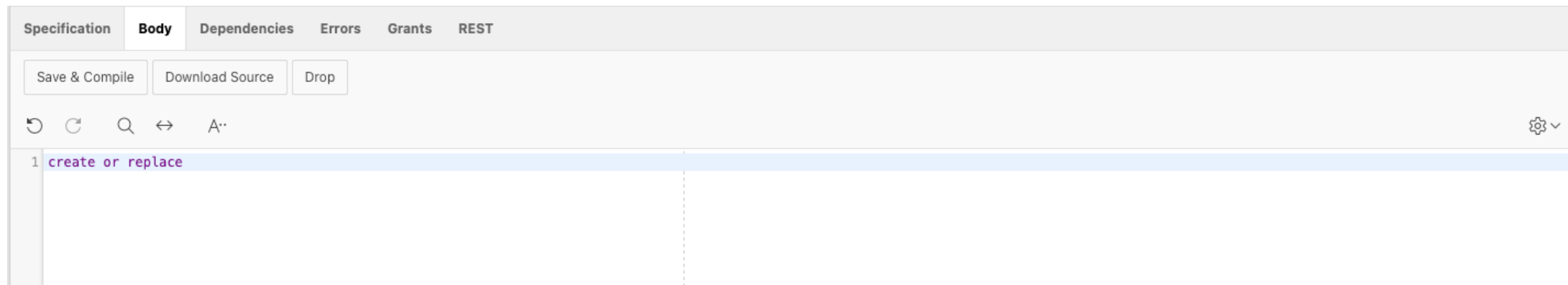
UC_DEMO

Specification **Body** Dependencies Errors Grants REST

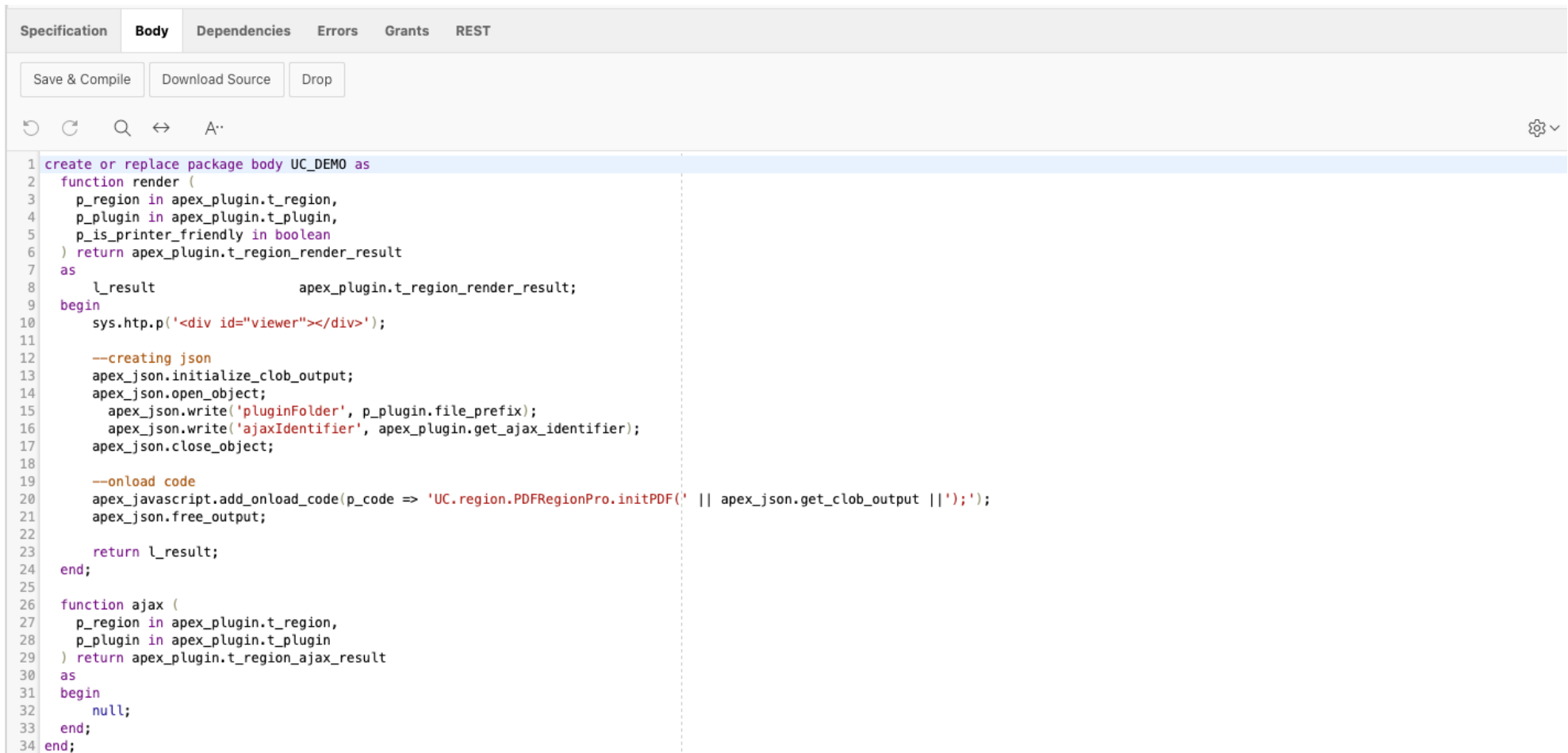
Save & Compile Download Source Drop

```
1 create or replace package UC_DEMO as
2   function render (
3     p_region in apex_plugin.t_region,
4     p_plugin in apex_plugin.t_plugin,
5     p_is_printer_friendly in boolean
6   ) return apex_plugin.t_region_render_result;
7
8   function ajax (
9     p_region in apex_plugin.t_region,
10    p_plugin in apex_plugin.t_plugin
11  ) return apex_plugin.t_region_ajax_result;
12
13 end;
14
```

Callback: Render function



Callback: Render function



```
1 create or replace package body UC_DEMO as
2   function render (
3     p_region in apex_plugin.t_region,
4     p_plugin in apex_plugin.t_plugin,
5     p_is_printer_friendly in boolean
6   ) return apex_plugin.t_region_render_result
7   as
8     l_result          apex_plugin.t_region_render_result;
9   begin
10    sys.http.p('<div id="viewer"></div>');
11
12    --creating json
13    apex_json.initialize_clob_output;
14    apex_json.open_object;
15    apex_json.write('pluginFolder', p_plugin.file_prefix);
16    apex_json.write('ajaxIdentifier', apex_plugin.get_ajax_identifier);
17    apex_json.close_object;
18
19    --onload code
20    apex_javascript.add_onload_code(p_code => 'UC.region.PDFRegionPro.initPDF(' || apex_json.get_clob_output || ');');
21    apex_json.free_output;
22
23    return l_result;
24  end;
25
26  function ajax (
27    p_region in apex_plugin.t_region,
28    p_plugin in apex_plugin.t_plugin
29  ) return apex_plugin.t_region_ajax_result
30  as
31  begin
32    null;
33  end;
34 end;
```

Render function

```
2  function render (  
3      p_region in apex_plugin.t_region,  
4      p_plugin in apex_plugin.t_plugin,  
5      p_is_printer_friendly in boolean  
6  ) return apex_plugin.t_region_render_result  
7  as  
8      l_result          apex_plugin.t_region_render_result;  
9  begin  
10     sys.http.p('<div id="viewer"></div>');  Send HTML to frontend  
11  
12     --creating json  
13     apex_json.initialize_clob_output;  
14     apex_json.open_object;  
15     apex_json.write('pluginFolder', p_plugin.file_prefix);  
16     apex_json.write('ajaxIdentifier', apex_plugin.get_ajax_identifier);  
17     apex_json.close_object;  
18  
19     --onload code  
20     apex_javascript.add_onload_code(p_code => 'UC.region.PDFRegionPro.initPDF(' || apex_json.get_clob_output || ');');  
21     apex_json.free_output;  
22  
23     return l_result;  
24 end;
```

Render function

```
2  function render (  
3      p_region in apex_plugin.t_region,  
4      p_plugin in apex_plugin.t_plugin,  
5      p_is_printer_friendly in boolean  
6  ) return apex_plugin.t_region_render_result  
7  as  
8      l_result          apex_plugin.t_region_render_result;  
9  begin  
10     sys.http.p('<div id="viewer"></div>');  
11  
12     --creating json  
13     apex_json.initialize_clob_output;  
14     apex_json.open_object;  
15     apex_json.write('pluginFolder', p_plugin.file_prefix);  
16     apex_json.write('ajaxIdentifier', apex_plugin.get_ajax_identifier);  
17     apex_json.close_object;  
18  
19     --onload code  
20     apex_javascript.add_onload_code(p_code => 'UC.region.PDFRegionPro.initPDF(' || apex_json.get_clob_output || ');');  
21     apex_json.free_output;  
22  
23     return l_result;  
24 end;
```

Send necessary
config/attributes to
frontend

Render function

```
2 function render (  
3   p_region in apex_plugin.t_region,  
4   p_plugin in apex_plugin.t_plugin,  
5   p_is_printer_friendly in boolean  
6 ) return apex_plugin.t_region_render_result  
7 as  
8   l_result          apex_plugin.t_region_render_result;  
9 begin  
10   sys.http.p('<div id="viewer"></div>');  
11  
12   --creating json  
13   apex_json.initialize_clob_output;  
14   apex_json.open_object;  
15   apex_json.write('pluginFolder', p_plugin.file_prefix);  
16   apex_json.write('ajaxIdentifier', apex_plugin.get_ajax_identifier);  
17   apex_json.close_object;  
18  
19   --onload code  
20   apex_javascript.add_onload_code(p_code => 'UC.region.PDFRegionPro.initPDF(' || apex_json.get_clob_output || ');');  
21   apex_json.free_output;  
22  
23   return l_result;  
24 end;
```



Reference function in JavaScript file

JavaScript onLoad code

- Executed via the render function on page load
- Links backend → frontend
 - Set configurations for the JavaScript library
 - Communicate with APEX
 - Fetch data, if needed

Modifying our JavaScript 'script.js' file

```
/* globals apex,$ */
var fileName;
window.UC = window.UC || {};
UC.region = UC.region || {};
UC.region.PDFRegionPro = UC.region.PDFRegionPro || {};

/**
 * Initialization function for the PDF Region Pro Plugin.
 *
 * @param {object}      config      The object holding the slider's configuration.
 * @param {string}      [config.pluginFolder]  Folder where the plugin is located.
 * @param {string}      [config.ajaxIdentifier] Identifier for the AJAX call.
 */

UC.region.PDFRegionPro.initPDF = function (config) {
  WebView(
    {
      // Path below needs to be dynamic
      path: config.pluginFolder + "lib",
    },
    document.getElementById("viewer")
  ).then((instance) => {
    if (navigator.appVersion.indexOf("MSIE") > -1) {
      return alert("This WebView version is not supported in your browser. ");
    }
  });
};
```

Modifying our JavaScript 'script.js' file

```
/* globals apex,$ */
var fileName;
window.UC = window.UC || {};
UC.region = UC.region || {};
UC.region.PDFRegionPro = UC.region.PDFRegionPro || {};

/**
 * Initialization function for the PDF Region Pro Plugin.
 *
 * @param {object}      config      The object holding the slider's configuration.
 * @param {string}      [config.pluginFolder]  Folder where the plugin is located.
 * @param {string}      [config.ajaxIdentifier] Identifier for the AJAX call.
 */

UC.region.PDFRegionPro.initPDF = function (config) {
  WebView(
    {
      // Path below needs to be dynamic
      path: config.pluginFolder + "lib",
    },
    document.getElementById("viewer")
  ).then((instance) => {
    if (navigator.appVersion.indexOf("MSIE") > -1) {
      return alert("This WebView version is not supported in your browser. ");
    }
  });
};
```

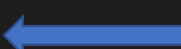
} Namespace to group our plug-in code

Modifying our JavaScript 'script.js' file

```
/* globals apex,$ */
var fileName;
window.UC = window.UC || {};
UC.region = UC.region || {};
UC.region.PDFRegionPro = UC.region.PDFRegionPro || {};

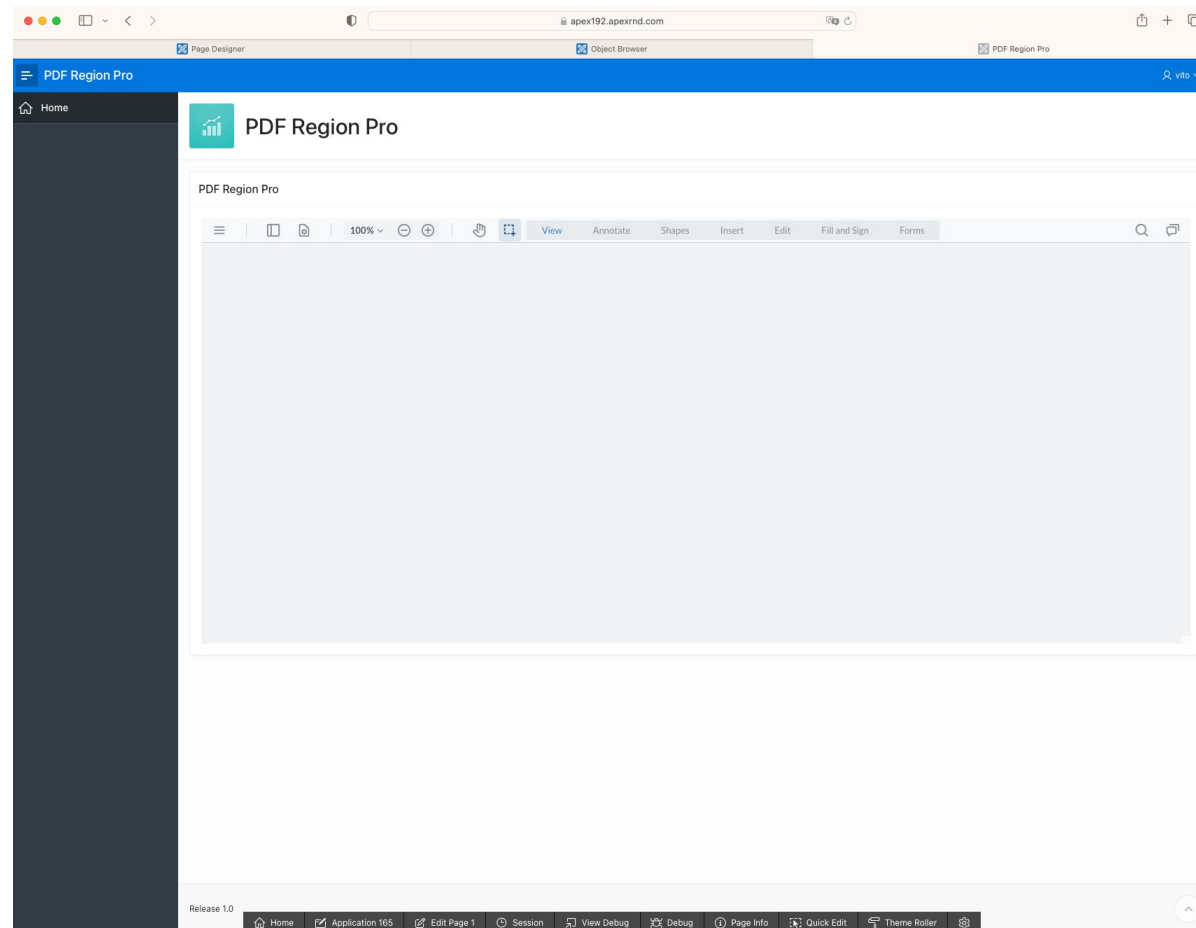
/**
 * Initialization function for the PDF Region Pro Plugin.
 *
 * @param {object}      config      The object holding the slider's configuration.
 * @param {string}      [config.pluginFolder]  Folder where the plugin is located.
 * @param {string}      [config.ajaxIdentifier] Identifier for the AJAX call.
 */

UC.region.PDFRegionPro.initPDF = function (config) {
  WebViewer(
    {
      // Path below needs to be dynamic
      path: config.pluginFolder + "lib",
    },
    document.getElementById("viewer")
  ).then((instance) => {
    if (navigator.appVersion.indexOf("MSIE") > -1) {
      return alert("This WebViewer version is not supported in your browser. ");
    }
  });
};
```

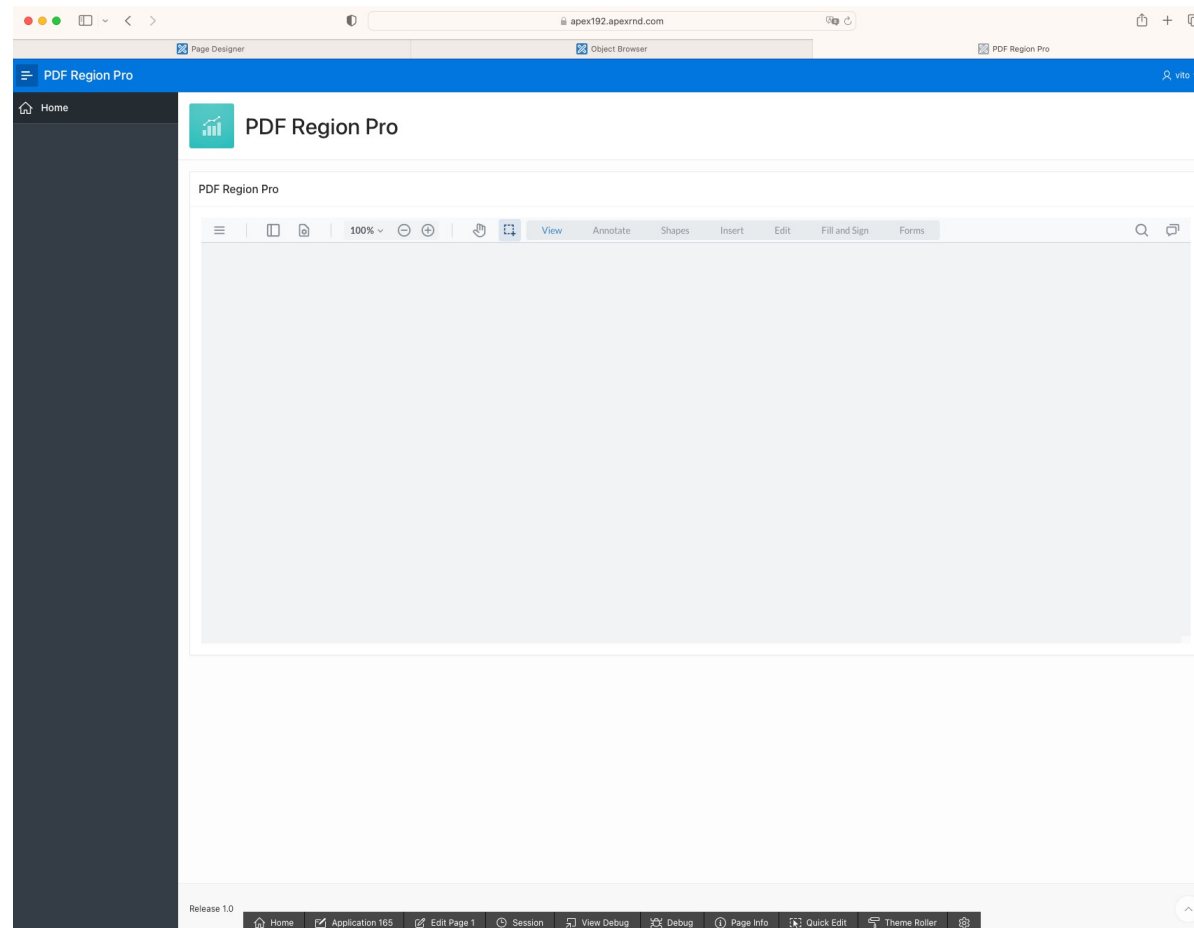


Reference the lib folder we uploaded in plug-in files

View the result



View the result: where is the PDF?



Core Concepts: Region Plug-in

- Attributes
 - Standard & Custom
- **Callbacks**
 - Render
 - **Ajax**
- Events

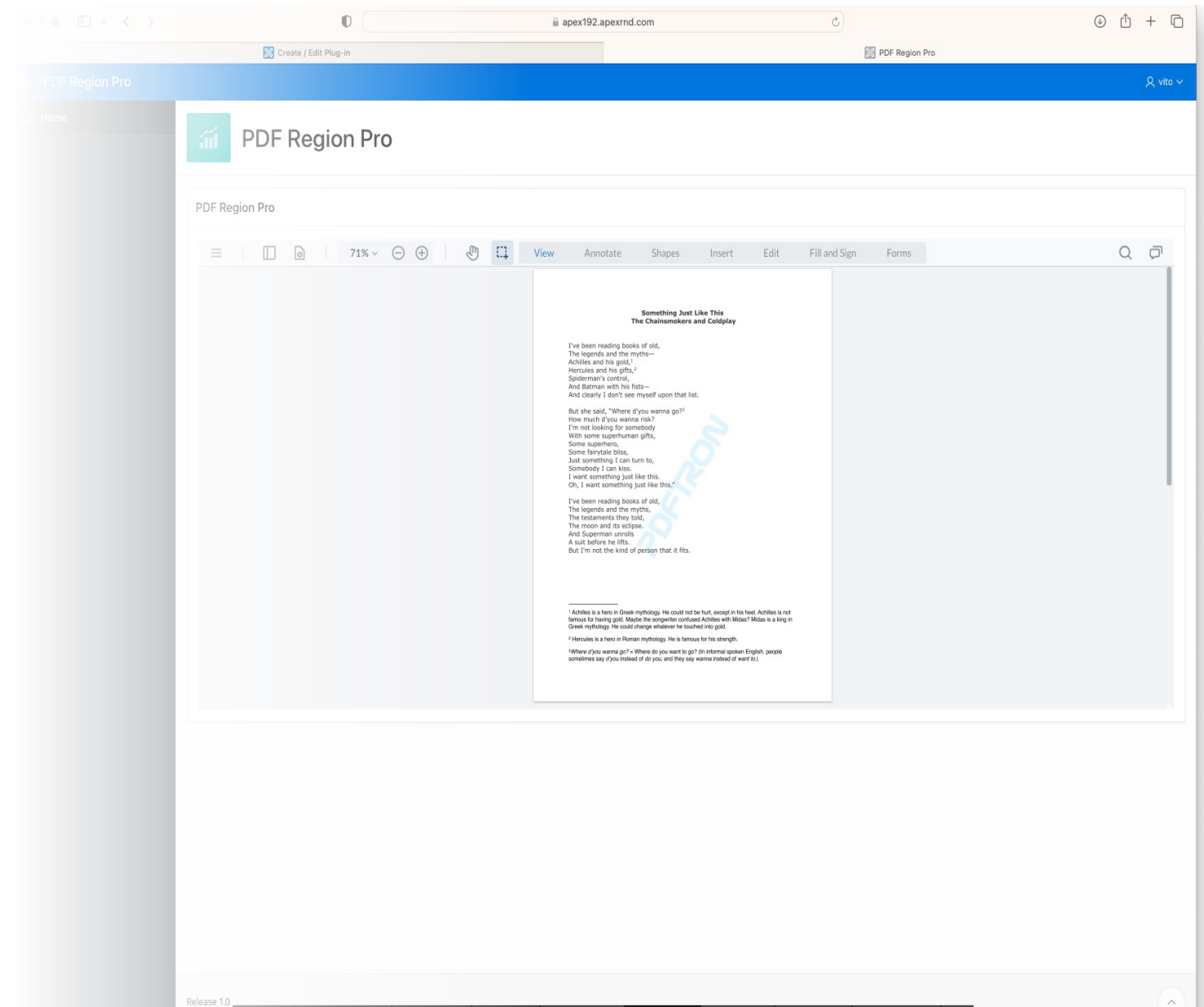
Calling the AJAX function in JS

```
UC.region.PDFRegionPro.initPDF = function (config) {  
  WebView(  
    {  
      // Path below needs to be dynamic  
      path: config.pluginFolder + "lib",  
    },  
    document.getElementById("viewer")  
  ).then((instance) => {  
    if (navigator.appVersion.indexOf("MSIE") > -1) {  
      return alert("This WebView version is not supported in your browser. ");  
    }  
  
    const result = apex.server.plugin(config.ajaxIdentifier, {  
      x01: '328863081987187356170865129631431428691',  
    })  
  
    result.done(function (data) {  
      instance.UI.loadDocument(base64ToBlob(data.clob));  
    });  
  });  
};  
  
function base64ToBlob(base64) {  
  const binaryString = window.atob(base64);  
  const len = binaryString.length;  
  const bytes = new Uint8Array(len);  
  for (let i = 0; i < len; ++i) {  
    bytes[i] = binaryString.charCodeAt(i);  
  }  
  
  return new Blob([bytes], { type: "application/pdf" });  
}
```

Callback: AJAX

```
function ajax (  
    p_region in apex_plugin.t_region,  
    p_plugin in apex_plugin.t_plugin  
) return apex_plugin.t_region_ajax_result  
as  
    l_return apex_plugin.t_region_ajax_result;  
    l_id      varchar2(4000) := apex_application.g_x01;  
  
    l_clob    clob;  
    l_blob    blob;  
begin  
  
    select doc  
    into l_blob  
    from pdf_demo_files  
    where id = l_id;  
  
    l_clob := apex_web_service.BLOB2CLOBBASE64(l_blob);  
  
    apex_json.open_object;  
    apex_json.write('clob', l_clob);  
    apex_json.close_object;  
  
    return l_return;  
end;
```

Result



Phase 5

UC Standards & Best Practices

- Easy to use
(well documented and tested, integrated with APEX)
- Use APEX components and meta-data
- Carefully plan attributes usage
- Add extensions for APEX developers for PL/SQL and JS
- Fight the temptation „I will fix it later”
- Handle errors before implementing desired functionality
- Add debugging from day 1

<>

Plug-in: Working item plug-in

CancelDeleteApply Changes

Show AllNameSettingsStatic List of ValuesDefault ValueConditionSupported forHelp Text and ExamplesComr

Name

* ScopeComponent?

* Attribute1?

* Display Sequence10?

* LabelSettings?

Settings

TypeCheckboxes?

* Lov TypeStatic

Common☒?

Show in Wizard☒?

Required☐?

Static List of ValuesAdd Value

Sequence	Display	Return
10	Override Application Settings	OAS
20	Enable Option 1	E01
30	Enable Option 2	E02

Default Value

Default Value?

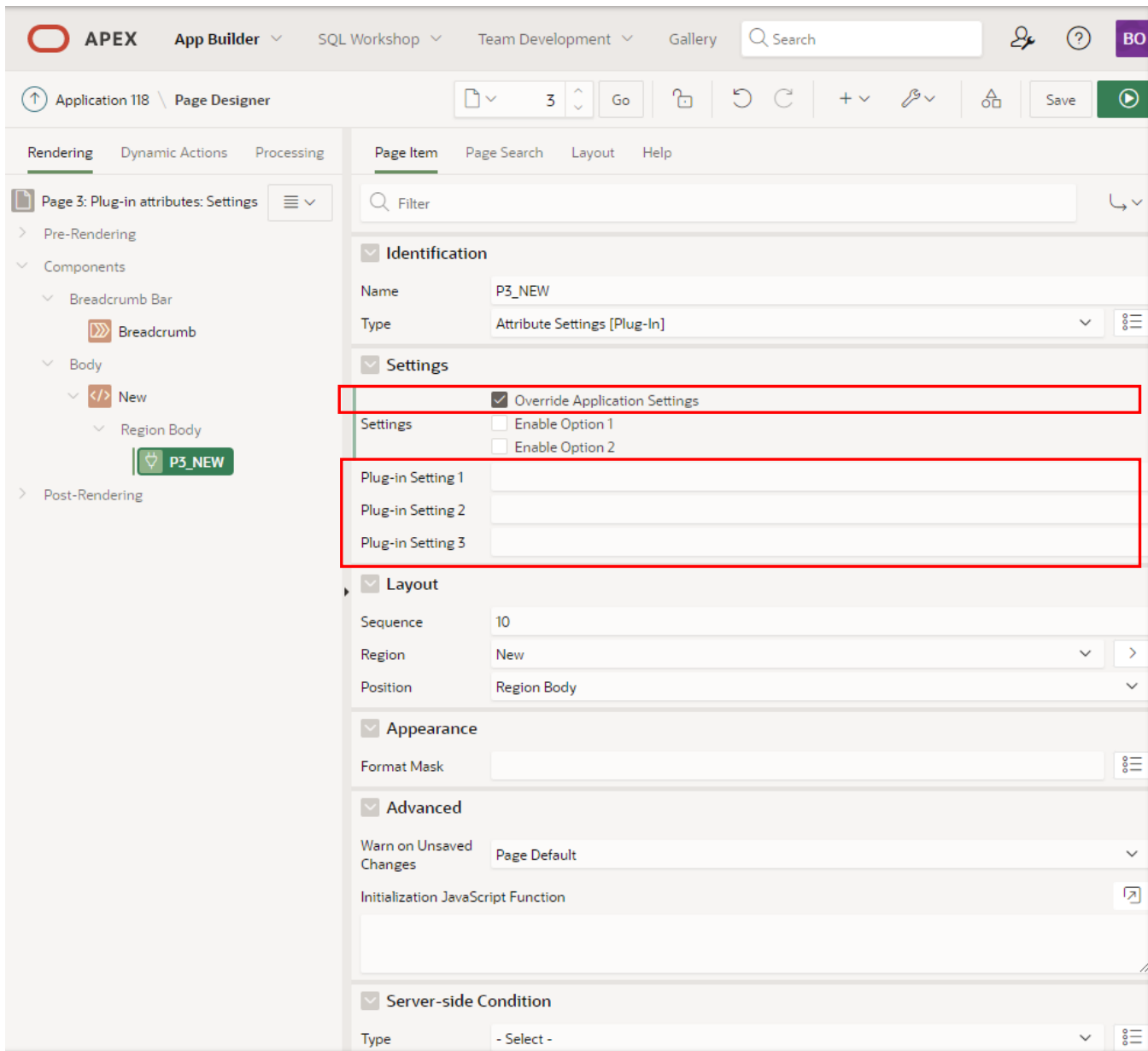
Attributes

A plug-in attribute is used to prompt the developer for additional data in the Builder when the plug-in is used.

☐ Stay on page?

CUSTOM ATTRIBUTES LIMITS / SETTINGS

- Checkbox type
- Unlimited slots*
- Ideal for boolean settings
- Ideal for attribute conditions



CUSTOM ATTRIBUTES LIMITS / SETTINGS

- Override application scope attributes
- Hide and show other page level attributes

ORACLE App Builder SQL Workshop Team Development App Gallery

Application 134 Page Designer 275 Go Save

Attributes Filter

Settings

Allowed File Types

- ☒ Microsoft Word
- ☒ Microsoft Excel
- ☒ Microsoft PowerPoint
- ☒ Open Document Text
- ☒ Open Document Spreadsheet
- ☒ Open Document Presentation
- ☐ Open Document Draw
- ☐ Disable Creating Files
- ☒ Disable Editor Messages
- ☒ Disable Exporting
- ☐ Disable Printing
- ☐ Disable Renaming
- ☐ Override Component Settings

Settings

Source: Custom table

Table Name: aoe_custom_tbl

Primary Key(s) Column(s): id

Item(s) Containing Primary Key(s) Value(s): P275_ID

Content Column: doc

Filename Column: doc_filename

MIME Type Column: doc_mimetype

Last Update Column: doc_lastupd

Version Column: version_number

Owner Column: doc_owner

Document Permissions: Everyone can read and edit

Custom PL/SQL: aoe_callback_pkg

Settings

Use the following options to configure the plug-in.

Available options include:

Disable Creating Files

Disable Editor Messages

When selected, only the editor error messages are displayed.

When not selected, the editor shows all messages including errors and success prompts.

Disable Exporting

When selected, exporting a document directly from the editor is disabled.

When not selected, exporting a document directly from the editor is enabled.

Disable Printing

When selected, the end-user can't print the document using browser print functionality - the editor print button is not displayed.

Printing is only possible when an APEX application implementing the plug-in, and the plug-in server-side files are accessible within the same domain. Otherwise, printing is not possible - clicking the print button raises a cross-origin JavaScript error.

Disable Renaming

Override Component Settings

When selected, components settings **Form action** and **REST URL** can be overridden on the region level.

CUSTOM ATTRIBUTES LIMITS / SETTINGS

- APEX Office Edit settings:
 - Enable developer to easily restrict what document types can be created or loaded
 - Commonly used settings are exposed for a developer convenient.

ORACLE App Builder SQL Workshop Team Development App Gallery

Application 163 Page Designer

Page Item Layout Component View Messages Page Search Help

Filter

Identification

Name P1_FROALA

Type UC - Rich Text Editor Pro [Plug-In]

Label

Label Froala

Settings

☐ Enable Browser Built-in Spell Check
☐ Enable Sticky Toolbar
☒ Override Application Attributes
☐ Validate CLOB Checksum On Page Submit
☒ Enable Plug-in: TUI Image Editor
☐ Enable Plug-in: Codox.io
☐ Enable Plug-in: Track Changes
☐ Enable Plug-in: Spell Check
☐ Enable Plug-in: Wiris

Toolbar Full

Height 330 px

Upload Image(s) on Request(s) UPDATE_CLOB_AND_UPLOAD

Image GET URL #DEFAULT_REST_URL_GET#

Image Upload URL #DEFAULT_REST_URL_MODULE#uploadFolder

Image Browse URL #DEFAULT_REST_URL_MODULE#browseFolder

Image Delete URL #DEFAULT_REST_URL_DELETE#

Initialization PL/SQL Code

Settings

Use following options to set the plug-in basic configuration.

Available options include:

Enable Browser Built-in Spell Check

Selecting this option enables the browser built-in spell checker to be active for the plug-in editor

Enable Sticky Toolbar

Selecting this option enables the plug-in editor toolbar to preserve the position while scrolling the page. When selected you can set the top offset (in pixels) using the plug-in attribute **Toolbar Top Offset**.

Override Application Attributes

Selecting this option allows overriding the plug-in application attributes on the item level.

Validate CLOB Checksum On Page Submit

Selecting this option allows validating CLOB checksum when the page is submitted. When selected you can restrict the plug-in built-in validation to given list of requests using the plug-in attribute **Validation Request(s)**.

About the plug-in built-in validation

When the page is submitted, the plug-in built-in validation compares CLOB checksum (set on page load) with CLOB checksum stored in the database. If CLOB checksum (set on page load) differs from CLOB checksum stored in the database, then the page submission is terminated with the plug-in error message **Value has been overridden in the meantime. Reload the page and save again..**

The feature should be used to prevent overriding CLOB content by two different end-users trying to save changes to the same CLOB. Another example is when the end-user opens the same CLOB in different browser tabs and tries to save different changes in these tabs.

This feature requires loading CLOB content using supporting plug-in **United Codes Froala RTE (Process)**, not the plug-in PL/SQL API, or using native APEX PL/SQL process.

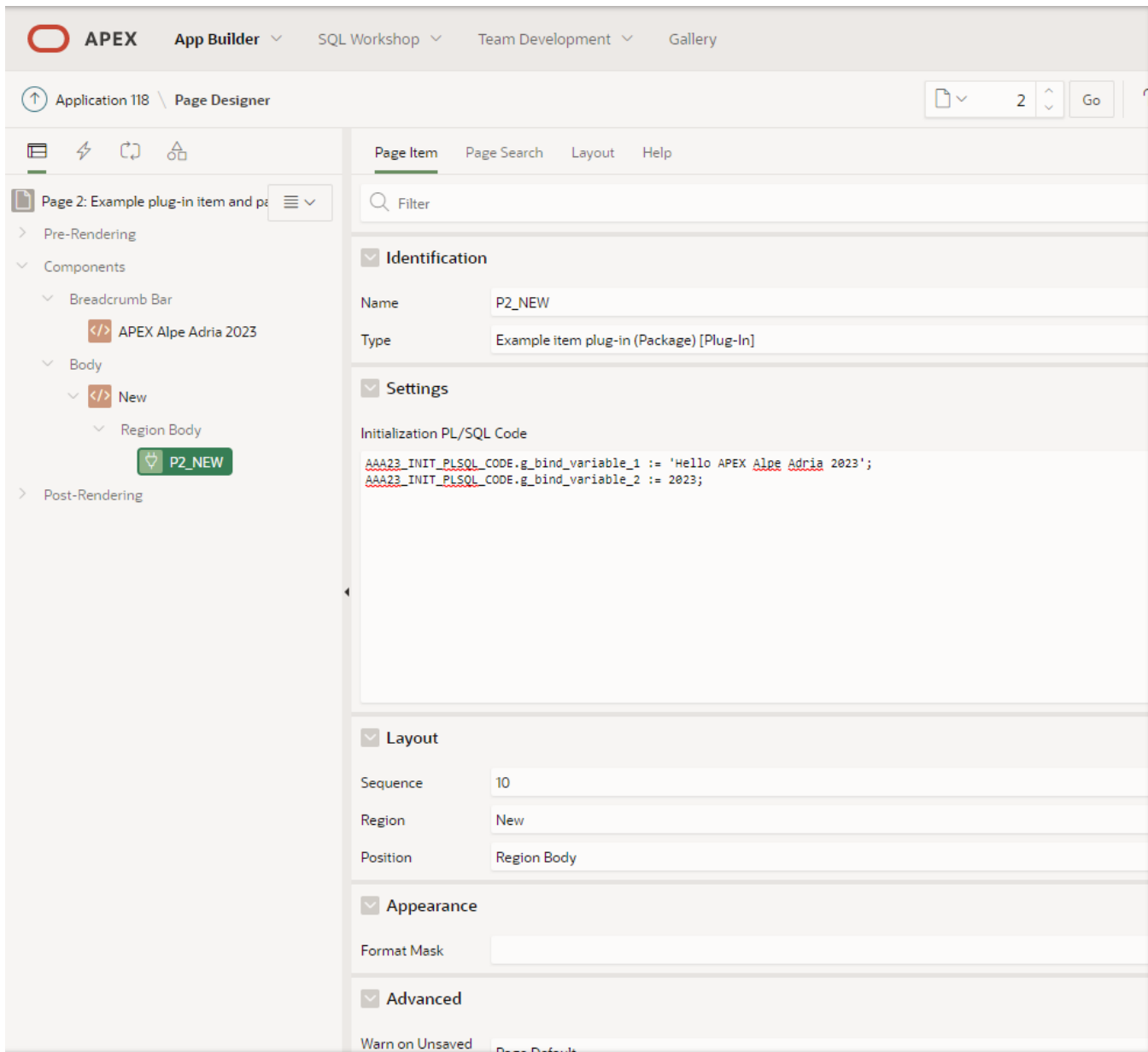
Enable Plug-in: TUI Image Editor

Selecting this option enables the plug-in with the TUI Image Editor. This allows end-users to edit images directly in an application. Edited images are re-uploaded or re-embedded depending on the Embed Images as Inline Base64 setting.

Learn more at <https://ui.toast.com/tui-image-editor>.

CUSTOM ATTRIBUTES LIMITS / SETTINGS

- Rich Text Editor Pro settings:
 - Possibility to override REST URL for image handling per component
 - Enable a developer what 3rd party plug-in are enabled



CUSTOM ATTRIBUTES LIMITS / INITIALIZATION PL/SQL CODE

- Flexible
- Unlimited slots*
- Prototype experimental features which can become new attribute
- Plug-in configuration based on APEX Session State

Attributes

Team Development
Gallery
Search
BO

For Item type plug-ins, identify the attributes that apply to this plug-in:

- Is Visible Widget - The widget is visible.
- Standard Form Element - The widget renders as a HTML form element (e.g. input, select or textarea). If not checked, `apex_plugin_util.get_element_attributes` will automatically add `aria-labelledby` to allow the non-standard form element to be labelled appropriately, so as to be accessible to screen readers. **Note:** This relies on the `#LABEL_ID#` substitution being defined in the item's corresponding label template. The Is Visible Widget attribute must also be checked.
- Session State Changeable - The value for an item can be changed in session state. The Is Visible Widget attribute must be checked.
- Has Read Only Attribute - The user can specify a "Read Only" condition for the item. The Is Visible Widget attribute must be checked for this attribute to be enabled.
- Has Escape Special Characters Attribute - The item has the Escape special characters field in the Security section when the item is edited. The Is Visible Widget attribute must be checked.
- Has Quick Pick Attributes - The item has quick pick attributes. The Is Visible Widget attribute must be checked.
- Has Source Attributes - The item has source related attributes such as Source Used, Source Type and Format Mask.
- Format Mask Date Only - For items that handle date information, the Format Mask field popup restricts the date selections by displaying date values to select from. The Has Source Attributes must be checked.
- Format Mask Number Only - For items that handle numeric information, the Format Mask field popup restricts the numeric selections by displaying

Cancel

Apply Changes

ported for

Standard Attributes

Standard Attribute additional Meta

☐ Standard Form Element	☐ Session State Changeable	☐ Has Read Only Attribute
☐ Has Quick Pick Attributes	☐ Has Source Attributes	☐ Format Mask Date Only
☐ Has Element Attributes	☐ Has Width Attributes	☐ Has Height Attribute
☐ Has Placeholder Attribute	☐ Has Icon Attribute	☐ Has Encrypt Session State Attribute
☐ Has LOV Display Null Attributes	☐ Has Cascading LOV Attributes	☐ Join LOV for Column Display
☐ Has Link	☒ Has "Initialization JavaScript Code" Attribute	☐ Session State Supports CLOB

Plug-ins

A plug-in is an extension to the built-in types of Oracle APEX. Use plug-ins for declarative use of new item, region, process and dynamic action types in your application.

☐ Stay on page

Tasks

Component Settings

Export Plug-in

Utilization

KNOW APEX CAPABILITIES / STANDARD ATTRIBUTES

- Check what you have by default
- Standard attributes – enable native component attributes
- Enable JavaScript Initialization Code
 - Unlimited configuration on JS level
 - Must be evaluated in plug-in JS
 - Handle JavaScript errors made by a developer

```

1  create or replace package AAA23_INIT_PLSQL_CODE as
2      g_bind_variable_1 varchar2(4000);
3      g_bind_variable_2 varchar2(4000);
4
5      procedure render (
6          p_item      in          apex_plugin.t_item,
7          p_plugin     in          apex_plugin.t_plugin,
8          p_param      in          apex_plugin.t_item_render_param,
9          p_result     in out nocopy apex_plugin.t_item_render_result
10     );
11
12 end AAA23_INIT_PLSQL_CODE;
13
14 create or replace package body AAA23_INIT_PLSQL_CODE as
15
16     procedure render (
17         p_item      in          apex_plugin.t_item,
18         p_plugin     in          apex_plugin.t_plugin,
19         p_param      in          apex_plugin.t_item_render_param,
20         p_result     in out nocopy apex_plugin.t_item_render_result
21     )
22     is
23         v_attr_plsql_init_code apex_application_page_items.attribute_01%type;
24         v_result               apex_plugin.t_item_render_result;
25         v_sql_parameters       apex_exec.t_parameters;
26     begin
27         v_attr_plsql_init_code := p_item.attribute_01;
28
29         apex_exec.execute_plsql(
30             p_plsql_code      => v_attr_plsql_init_code,
31             p_auto_bind_items => true,
32             p_sql_parameters  => v_sql_parameters
33         );
34
35         apex_debug.info('g_bind_variable_1: %s', NVL(g_bind_variable_1, '%NULL%'));
36         apex_debug.info('g_bind_variable_2: %s', NVL(g_bind_variable_2, '%NULL%'));
37
38         p_result := v_result;
39     end render;
40
41 end AAA23_INIT_PLSQL_CODE;

```

CUSTOM ATTRIBUTES LIMITS / INITIALIZATION PL/SQL CODE

- Plug-in PL/SQL code in package
- Variables in package specification
- Variables are available within transaction for plug-in render or ajax
- Execute dynamic PL/SQL code using APEX_EXEC

Elapsed	Execution	Message	Level	Elapsed	Execution	Message
0.03769	0.00003	Plugin meta data:	4	0.03802	0.00002	Dynamic Action meta data:
0.03772	0.00002	file_prefix: r/p_bartosz/118/files/plugin/82224950153154224/v1/	4	0.03804	0.00002	id: 82031057329348206
0.03774	0.00004	attribute_01:	4	0.03806	0.00002	name: For item
0.03778	0.00002	attribute_02:	4	0.03807	0.00002	action: PLUGIN_AAA23_DYNAMIC_ACTION
0.03779	0.00002	attribute_03:	4	0.03809	0.00002	attribute_01:
0.03781	0.00002	attribute_04:	4	0.03811	0.00002	attribute_02:
0.03783	0.00002	attribute_05:	4	0.03812	0.00002	attribute_03:
0.03785	0.00002	attribute_06:	4	0.03814	0.00002	attribute_04:
0.03787	0.00002	attribute_07:	4	0.03816	0.00002	attribute_05:
0.03788	0.00002	attribute_08:	4	0.03818	0.00002	attribute_06:
0.03790	0.00002	attribute_09:	4	0.03819	0.00002	attribute_07:
0.03792	0.00002	attribute_10:	4	0.03821	0.00002	attribute_08:
0.03794	0.00002	attribute_11:	4	0.03823	0.00002	attribute_09:
0.03795	0.00002	attribute_12:	4	0.03824	0.00002	attribute_10:
0.03797	0.00002	attribute_13:	4	0.03826	0.00002	attribute_11:
0.03799	0.00002	attribute_14:	4	0.03828	0.00002	attribute_12:
0.03800	0.00001	attribute_15:	4	0.03830	0.00002	attribute_13:
				0.03831	0.00002	attribute_14:
				0.03833	0.00002	attribute_15:
				0.03835	0.00002	init_javascript_code: function(){ alert(1); }
				0.03837	0.00002	triggering_region_id:
				0.03838	0.00002	affected_elements_type: ITEM
				0.03840	0.00002	affected_region_id:
				0.03842	0.00002	affected_button_id:
				0.03844	0.00008	affected_elements: P5_ITEM_AFFECTED_ELEMENT

KNOW APEX CAPABILITIES / DATA FOR PL/SQL

- Use APEX_PLUGIN_UTIL to debug plug-in

```

1 function render (
2   p_dynamic_action in apex_plugin.t_dynamic_action,
3   p_plugin          in apex_plugin.t_plugin
4 ) return apex_plugin.t_dynamic_action_render_result is
5   type t_item_row is table of APEX_APPLICATION_PAGE_ITEMS%rowtype;
6
7   v_result          apex_plugin.t_dynamic_action_render_result;
8 begin
9
10  APEX_PLUGIN_UTIL.DEBUG_DYNAMIC_ACTION(
11    p_plugin          => p_plugin,
12    p_dynamic_action  => p_dynamic_action
13  );
14
15  return v_result;
16
17 end render;
```

```

1  function render (
2      p_dynamic_action in apex_plugin.t_dynamic_action,
3      p_plugin          in apex_plugin.t_plugin
4  ) return apex_plugin.t_dynamic_action_render_result is
5      type t_item_row is table of APEX_APPLICATION_PAGE_ITEMS%rowtype;
6
7      v_da      apex_plugin.t_dynamic_action;
8      v_result  apex_plugin.t_dynamic_action_render_result;
9  begin
10     v_da := p_dynamic_action;
11
12     APEX_PLUGIN_UTIL.DEBUG_DYNAMIC_ACTION(p_plugin, p_dynamic_action);
13
14     -- fetch dynamic action row
15     select * into v_row_da_action from APEX_APPLICATION_PAGE_DA_ACTS where action_id = v_da.id;
16
17     -- fetch affected elements
18     if v_da.affected_elements_type = 'ITEM' then
19         select
20             * bulk collect into v_rows_item
21         from
22             APEX_APPLICATION_PAGE_ITEMS
23         where
24             INSTR(v_da.affected_elements, ITEM_NAME ) > 0
25         ;
26     elsif v_da.affected_elements_type = 'BUTTON' then
27         select * from apex_application_page_buttons where button_id = v_da.affected_button_id;
28
29     elsif v_da.affected_elements_type = 'REGION' then
30         SELECT * FROM APEX_APPLICATION_PAGE_REGIONS where region_id = v_da.affected_region_id;
31
32     end if;
33
34     -- rest render logic
35     -- ...
36     return v_result;
37 end render;

```

KNOW APEX CAPABILITIES / EVERYTHING YOU NEED IS IN

- Fetch an application data in plug-in render, ajax and other integration procedures

```

1 create or replace package body EXAMPLE_PKG as
2
3 -----
4 -- inferior function
5 function function_1(
6     p_da in apex_plugin.t_dynamic_action,
7     p_plug in apex_plugin.t_plugin
8 ) return varchar2 is
9 begin
10     return null;
11 end procedure_2;
12
13 -----
14 -- inferior procedure executing other function
15 procedure procedure_1(
16     p_da in apex_plugin.t_dynamic_action,
17     p_plug in apex_plugin.t_plugin
18 ) is
19     v_result varchar2(4000);
20 begin
21     v_result := function_1(
22         p_da => p_da,
23         p_plug => p_plug
24     );
25 end procedure_2;
26
27 -----
28 function render (
29     p_da in apex_plugin.t_dynamic_action,
30     p_plug in apex_plugin.t_plugin
31 ) return apex_plugin.t_dynamic_action_render_result
32 is
33     v_res apex_plugin.t_dynamic_action_render_result;
34 begin
35     procedure_1(
36         p_da => p_da,
37         p_plug => p_plug
38     );
39
40     return v_res;
41
42 end render;
43 end EXAMPLE_PKG;

```

```

1 create or replace package body EXAMPLE_PKG as
2
3     g_da apex_plugin.t_dynamic_action;
4     g_plug apex_plugin.t_plugin;
5
6 -----
7 -- inferior function
8 function function_1 return varchar2 is
9 begin
10     return null;
11 end procedure_2;
12
13 -----
14 -- inferior procedure executing other function
15 procedure procedure_1 is
16     v_result varchar2(4000);
17 begin
18     v_result := function_1;
19 end procedure_2;
20
21 -----
22 function render (
23     p_da in apex_plugin.t_dynamic_action,
24     p_plug in apex_plugin.t_plugin
25 ) return apex_plugin.t_dynamic_action_render_result
26 is
27     v_res apex_plugin.t_dynamic_action_render_result;
28 begin
29     g_da := p_da;
30     g_plug := p_plug;
31
32     procedure_1();
33
34     return v_res;
35 end render;
36
37 end EXAMPLE_PKG;

```

UC BEST PRACTICES / Use Global variables

- Use for plug-in integration procedures
- Global variables are valid within transaction
- Benefits
 - Have access to the plug-in data whenever needed
 - Inferior procedures and functions require less parameters in specification to preserve the context

```

1  create or replace package body AAA23_EXAMPLE_ITEM as
2      c_null constant varchar2(6) default '%NULL%';
3      c_ajax_mode_refresh constant varchar2(7) default 'refresh';
4      c_ajax_mode_reset constant varchar2(5) default 'reset';
5      c_err_ajax constant number default -20001;
6
7      e_ajax exception;
8      e_render_init_plsql exception;
9
10     PRAGMA exception_init(e_ajax, c_err_ajax);
11
12     procedure ajax_wrapper is
13         v_ajax_mode varchar2(4000);
14     begin
15
16         v_ajax_mode := NVL(APEX_ESCAPE.HTML( APEX_APPLICATION.G_X01 ), c_null);
17
18         if v_ajax_mode = c_ajax_mode_refresh then
19             ajax_refresh;
20
21         elsif v_ajax_mode = c_ajax_mode_reset then
22             ajax_reset;
23
24         else
25             raise_application_error(c_err_ajax, apex_string.format('Invalid AJAX mode: %s', v_ajax_mode ));
26         end if;
27
28     exception
29     when e_ajax then
30         apex_debug.error('The plug-in AJAX handler was called with invalid ajax mode: %s', v_ajax_mode);
31         raise;
32
33     when others then
34         apex_debug.error('An unexpected error was raised: %s', SQLERRM);
35         raise;
36
37     end ajax_wrapper;
38
39 > procedure ajax_main( ...
44 > ) is
45 > begin...
52 > exception...
65 > end ajax_main;
66
67 end AAA23_EXAMPLE_ITEM;

```

UC BEST PRACTICES / Use constants

- Reference data passed from frontend
- Reference attribute options values
- Reference APEX data stored in tables, for example value „Yes”
- Benefits:
 - Impossible to make typos
 - All important literals in one place

```

1  create or replace package body AAA23_EXAMPLE_ITEM as
2
3      g_item      apex_plugin.t_item;
4      g_plugin    apex_plugin.t_plugin;
5      g_param_item apex_plugin.t_item_render_param;
6      g_param_ajax apex_plugin.t_item_ajax_param;
7
8      procedure render_main(
9          p_item      in      apex_plugin.t_item,
10         p_plugin    in      apex_plugin.t_plugin,
11         p_param     in      apex_plugin.t_item_render_param,
12         p_result    in out nocopy apex_plugin.t_item_render_result
13     )
14     is
15     begin
16         g_item      := p_item;
17         g_plugin    := p_plugin;
18         g_param_item := p_param;
19
20         APEX_PLUGIN_UTIL.DEBUG_PAGE_ITEM(g_plugin, g_item);
21
22         render_wrapper;
23
24         p_result.is_navigable := true;
25
26     exception
27         when others then
28             APEX_ERROR.ADD_ERROR(
29                 p_message      => 'Rendering the plug-in failed.',
30                 p_additional_info => get_err_add_info(SQLCODE),
31                 p_display_location => apex_error.c_on_error_page,
32                 p_page_item_name => g_item.name
33             );
34
35     end render_main;
36
37 end AAA23_EXAMPLE_ITEM;

```

Wrapped procedure

Only „others“ exception handler

UC BEST PRACTICES / PL/SQL ERROR HANDLING

- Wrap procedures
- APEX_ERROR handles error in top level wrapper

```

1  create or replace package body AAA23_EXAMPLE_ITEM as
2      c_err_ajax          constant number          default -20001;
3      c_err_render_init_plsql constant number      default -20002;
4      c_error_prefix      constant varchar2(2)     default 'UC';
5
6      function get_err_add_info(
7          p_sqlcode in number
8      ) return varchar2 is
9          v_sql_err_code  varchar2(4000);
10         v_result         varchar2(4000);
11     begin
12         v_sql_err_code := c_error_prefix||p_sqlcode;
13
14         return case p_sqlcode
15             when c_err_ajax then
16                 apex_string.format('An internal error was raised in AJAX procedure: %s', v_sql_err_code)
17             when c_err_render_init_plsql then
18                 apex_string.format('An internal error was raised in rendering procedure: %s', v_sql_err_code)
19             else
20                 'Unexpected PL/SQL error was raised. Check debug messages to learn more.'
21             end;
22     end get_err_add_info;
23
24
25 > procedure render_main( ...
30 > ) is
31 > begin...
42 exception
43     when others then
44         APEX_ERROR.ADD_ERROR(
45             p_message      => 'Rendering the plug-in failed.',
46             p_additional_info => get_err_add_info(SQLCODE),
47             p_display_location => apex_error.c_on_error_page,
48             p_page_item_name => g_item.name
49         );
50
51     end render_main;
52
53 end AAA23_EXAMPLE_ITEM;

```

UC BEST PRACTICES / PL/SQL error handling

- Provide message to end-user based on raised exception code
- All error codes not registered should provide generic message for not handled error

```

1  ✓ create or replace package body AAA23_EXAMPLE_ITEM as
2
3      e_render_init_plsql      exception;
4      c_err_render_init_plsql  constant number          default -20002;
5
6      PRAGMA exception_init(e_render_init_plsql , c_err_render_init_plsql);
7
8  ✓ procedure execute_init_plsql is
9      v_attr_plsql_init_code   apex_application_page_items.attribute_01%type;
10     v_sql_parameters          apex_exec.t_parameters;
11  ✓ begin
12     v_attr_plsql_init_code := g_item.attribute_05;
13     v_attr_plsql_init_code := NVL(v_attr_plsql_init_code, 'return;');
14
15 >     apex_exec.execute_plsql( ...
16 > );
17
18
19
20
21  ✓ exception
22  ✓   when others then
23     raise_application_error(c_err_render_init_plsql, 'Executing init PL/SQL code raised an error:'||SQLERRM, true);
24  end execute_init_plsql;
25
26  ✓ procedure render_wrapper is
27     v_item_attributes          varchar2(4000);
28
29     v_item_name                varchar2(4000);
30     v_item_jquery              varchar2(4000);
31     v_item_ajax_id             varchar2(4000);
32  ✓ begin
33     execute_init_plsql;
34
35 >     apex_javascript.add_onload_code(' ...
36 > ');
37
38
39
40
41
42
43
44
45
46 >     v_item_attributes := apex_plugin_util.get_element_attributes( ...
47 > );
48
49
50
51
52
53
54
55     sys.http.prn( apex_string.format('<input type="text" %s>', v_item_attributes) );
56
57  ✓ exception
58  ✓   when e_render_init_plsql then
59     apex_debug.error('Evaluating "Initialization PL/SQL Code" raised an error: %s', SQLERRM);
60     raise;
61  ✓   when others then
62     apex_debug.error('An unexpected error was raised while rendering plug-in: %s', SQLERRM);
63     raise;
64  end render_wrapper;
65 end AAA23_EXAMPLE_ITEM;

```

UC BEST PRACTICES / PL/SQL ERROR HANDLING

- Register plug-in exceptions
- Use „raise_application_error” to raise plug-in registered errors
 - „true” preserves previous errors

JS error handling

- Use `throw new Error('Error message')`
- Use `try()` `catch(error)` clause and populate an error to main handler
- Unhandled errors cause APEX toolbar icon to flash red
- Take advantage of LOG LEVEL to narrow amount of messages
- Consider handling errors on three levels:
 - PL/SQL render procedure
 - Plug-in initialization code and functions (methods)
 - Plug-in async functions → events, AJAX requests, promises

```

359 },
360 _wrap_func: function( pShowError, pFnInstance ){
361     var
362         fnName = '_wrap_func',
363         fnParams = [],
364         fnResult
365     ;
366
367     this.log.trace( fnName, 'start' );
368
369     try {
370
371         for (var i=2; i < arguments.length; i++) {
372             fnParams.push( arguments[i] );
373         }
374
375         this.log.trace( fnName, '... execute wrapped function', {
376             "func": pFnInstance,
377             "params": fnParams
378         });
379
380         fnResult = pFnInstance.apply( this, fnParams );
381
382         this.log.trace( fnName, '... function result:', {"result": fnResult} );
383         this.log.trace( fnName, 'end' );
384
385         return fnResult;
386     } catch( error ) {
387         if ( error.cause ) {
388             this.log.error('Plug-in function wrapper raised an error:', error);
389             this.log.warn('An error cause:', error.cause);
390         }
391         else {
392             this.log.error('Plug-in function wrapper raised an error:', error);
393         }
394
395         if ( pShowError ) {
396             this._wrap_apex_message_showErrors( 'Plug-in function wrapper raised an error: '+error.message, true );
397         }
398     }
399
400 }
401
402 this.log.trace( fnName, 'end' );
403 }
404 });

```

UC BEST PRACTICES / WRAP ASYNC FUNCTIONS

- Use JS wrapper for async functions
- Use function binding to keep context of the plug-in JS
- Log information about initial error



Phase 6

Sample App & Documentation

Sample Application

- Implementation examples
- Easy to reuse
- Explanation

Documentation

- About
- Getting Started
- Different Attributes
- Database Objects
- Known Issues

PDF Region Pro

PDF Region Pro delivers the highest quality rendering, conversion, and PDF manipulation capabilities through a single, customizable Oracle APEX plug-in.

Type: Region

Author: Vito Van Hecke

Level: Gold

APEX versions: 19.2

Maintained by: United Codes

Last updated: 30-APR-2023

The screenshot shows a PDF document viewer interface. At the top, there are tabs for 'Open document', 'Upload document', and 'Document details'. Below the tabs is a toolbar with various icons for document manipulation. The main content area displays a form with several bullet points and a signature field. The signature field is labeled 'Sign Here' and includes a 'Signature' placeholder. Below the signature field, there are fields for 'Print Name' and 'Date (MM-DD-YYYY)'. The form also includes a checkbox for 'I certify that I have the capacity to sign for the entity identified on line 1 of this form.' and a note about submitting a new form within 30 days if the certification becomes incorrect. The document is titled 'W-9BEN-F (Rev. 10-2011)'.

Close

[View Documentation](#)

[View Sample App](#)

Rich Text Editor Pro

United Codes Rich Text Editor Pro integrates the popular Froala WYSIWYG Editor into an awesome plug-in for APEX. This enhanced HTML/Rich Text editor takes care of the CLOB handling for you - so you don't have to worry about the 32,767 character limit! Easily handles images, is highly customizable, and supports multiple languages.

Type: Item

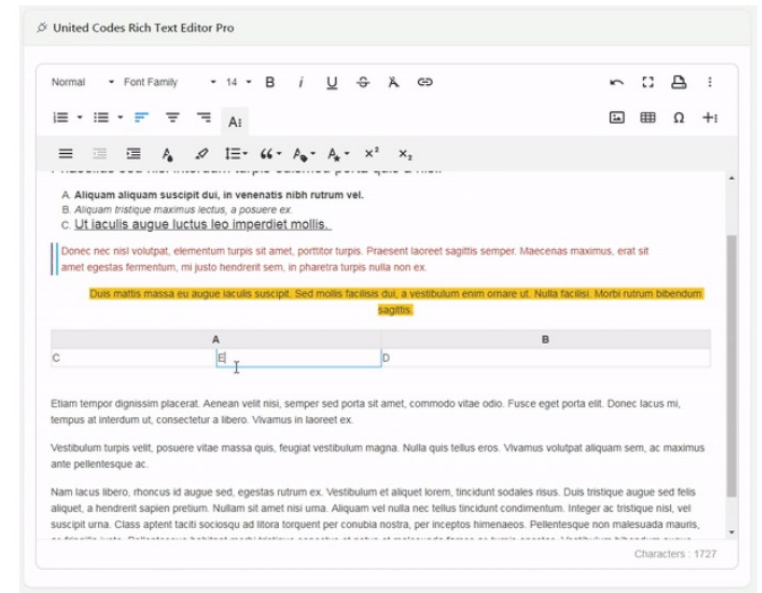
Author: Bartosz Ostrowski

Level: Gold

APEX versions: 19.2 and up

Maintained by: United Codes

Last updated: 21-MAR-2023



Close

[View Documentation](#)

[View Sample App](#)

RTE Editor Pro

More than just
1 plug-in!

Name	Date Modified	Size	Kind
apexplugin.json	21 Mar 2023 at 20:23	2 KB	JSON
> app	31 Jan 2023 at 15:00	--	Folder
> build	17 Mar 2023 at 09:55	--	Folder
> doc	10 Mar 2023 at 16:14	--	Folder
> docs	17 Mar 2023 at 09:55	--	Folder
> install_extras	17 Mar 2023 at 09:55	--	Folder
README.md	14 Mar 2023 at 19:46	4 KB	Document
> src	17 Mar 2023 at 09:55	--	Folder
> apps	13 Mar 2023 at 20:11	--	Folder
> help_creator	13 Mar 2023 at 20:11	--	Folder
> sample_app	17 Mar 2023 at 09:55	--	Folder
> assets	31 Jan 2023 at 15:46	--	Folder
> froala	13 Mar 2023 at 20:11	--	Folder
> css	13 Mar 2023 at 20:11	--	Folder
> js	13 Mar 2023 at 20:11	--	Folder
> plsql_src	13 Mar 2023 at 20:11	--	Folder
> tables	13 Mar 2023 at 20:11	--	Folder
> plugin	17 Mar 2023 at 09:55	--	Folder
> assets	13 Mar 2023 at 20:11	--	Folder
> css	13 Mar 2023 at 20:11	--	Folder
dynamic_action_plugin_u...s_froala_rte_da_conf.sql	17 Mar 2023 at 09:55	35 KB	Document
item_type_plugin_unitedcodes_froala_rte.sql	17 Mar 2023 at 09:55	6.6 MB	Document
> js	13 Mar 2023 at 20:11	--	Folder
> plsql	13 Mar 2023 at 20:11	--	Folder
process_type_plugin_u...s_froala_rte_process.sql	17 Mar 2023 at 09:55	14 KB	Document
README.md	31 Jan 2023 at 15:54	1 byte	Document
> rest	13 Mar 2023 at 20:11	--	Folder
ORDS_REST.sql	13 Mar 2023 at 20:11	9 KB	Document
> packages	13 Mar 2023 at 20:11	--	Folder
> sequences	13 Mar 2023 at 20:11	--	Folder
> tables	13 Mar 2023 at 20:11	--	Folder
> scripts	17 Mar 2023 at 09:55	--	Folder
ddl_uc_rte_manual_deinstall.sql	17 Mar 2023 at 09:55	873 bytes	Document

3 plug-ins

- item
- dynamic action
- process

2 apps

REST API

Name	Date Modified	Size	Kind
apexplugin.json	21 Mar 2023 at 20:23	2 KB	JSON
> app	31 Jan 2023 at 15:00	--	Folder
> build	17 Mar 2023 at 09:55	--	Folder
> doc	10 Mar 2023 at 16:14	--	Folder
> docs	17 Mar 2023 at 09:55	--	Folder
> install_extras	17 Mar 2023 at 09:55	--	Folder
README.md	14 Mar 2023 at 19:46	4 KB	Document
▼ src	17 Mar 2023 at 09:55	--	Folder
▼ apps	13 Mar 2023 at 20:11	--	Folder
> help_creator	13 Mar 2023 at 20:11	--	Folder
> sample_app	17 Mar 2023 at 09:55	--	Folder
▼ assets	31 Jan 2023 at 15:46	--	Folder
> froala	13 Mar 2023 at 20:11	--	Folder
> css	13 Mar 2023 at 20:11	--	Folder
> js	13 Mar 2023 at 20:11	--	Folder
▼ plsql_src	13 Mar 2023 at 20:11	--	Folder
▼ tables	13 Mar 2023 at 20:11	--	Folder
▼ plugin	17 Mar 2023 at 09:55	--	Folder
> assets	13 Mar 2023 at 20:11	--	Folder
> css	13 Mar 2023 at 20:11	--	Folder
dynamic_action_plugin_u...s_froala_rte_da_conf.sql	17 Mar 2023 at 09:55	35 KB	Document
item_type_plugin_unitedcodes_froala_rte.sql	17 Mar 2023 at 09:55	6.6 MB	Document
> js	13 Mar 2023 at 20:11	--	Folder
> plsql	13 Mar 2023 at 20:11	--	Folder
process_type_plugin_u...s_froala_rte_process.sql	17 Mar 2023 at 09:55	14 KB	Document
README.md	31 Jan 2023 at 15:54	1 byte	Document
▼ rest	13 Mar 2023 at 20:11	--	Folder
ORDS_REST.sql	13 Mar 2023 at 20:11	9 KB	Document
> packages	13 Mar 2023 at 20:11	--	Folder
> sequences	13 Mar 2023 at 20:11	--	Folder
> tables	13 Mar 2023 at 20:11	--	Folder
▼ scripts	17 Mar 2023 at 09:55	--	Folder
ddl_uc_rte_manual_deinstall.sql	17 Mar 2023 at 09:55	873 bytes	Document

▼ Identification

Name

P304_CLOB_CONTENT

Type

UC - Rich Text Editor Pro [Plug-In]

▼

▼ Label

Label

Content

▼ Settings

Settings

☒ Enable Browser Built-in Spell Check

☐ Enable Sticky Toolbar

☐ Override Application Attributes

☐ Validate CLOB Checksum On Page Submit

☒ Enable Plug-in: TUI Image Editor

☒ Enable Plug-in: Codox.io

☒ Enable Plug-in: Track Changes

☒ Enable Plug-in: Spell Check

☒ Enable Plug-in: Wiris

Toolbar

Full

▼

Height

500

px

Upload Image(s)
on Request(s)

Initialization PL/SQL Code

Item Plug-in
(supports extra plug-ins)

Dynamic Action Plug-in

Identification

Action UC - Rich Text Editor Pro (Extend) [Plug-I ▼] ⋮

Settings

Action

Toolbar Customization

Group

Buttons Visible

Change Toolbar

Change Settings

Image Browser Parameters

Reset value

Update CLOB

Upload Parameters

Affected Elements

Identification

Action UC - Rich Text Editor Pro (Extend) [Plug-I ▼] ⋮

Settings

Action Change Toolbar ▼

Toolbar Customization Change Number of Visible Buttons ▼

Group Paragraph ▼

Buttons Visible 10

Affected Elements

Process Plug-in

Type	UC - Rich Text Editor Pro (Process) [Plug·
Settings	
Type	
Item	
Schema	
Table Name	UC_FROALA_SAMPLE_CLOBS
CLOB Column Name	clob_content
PK(s) Column(s) Name(s)	session_id,id

Clear CLOB

Load CLOB (SQL Query)

Load CLOB (Function Body returning CLOB)

✓ Load CLOB (Table)

Update CLOB (PL/SQL Code)

Update CLOB (Table)

Delete Removed Images

Name	Load CLOB
Type	UC - Rich Text Editor Pro (Process) [Plug·
Settings	
Type	Load CLOB (Table)
Item	P304_CLOB_CONTENT
Schema	#OWNER#
Table Name	UC_FROALA_SAMPLE_CLOBS
CLOB Column Name	clob_content
PK(s) Column(s)	

Item plug-in exposes dynamic actions

UC Froala Before Initialization [UC - Rich Text Editor Pro]
UC Froala CLOB Reset [UC - Rich Text Editor Pro]
UC Froala CLOB Updated [UC - Rich Text Editor Pro]
UC Froala Content Changed [UC - Rich Text Editor Pro]
UC Froala Image Browser Error [UC - Rich Text Editor Pro]
UC Froala Image Upload Error [UC - Rich Text Editor Pro]
UC Froala Images In Collection [UC - Rich Text Editor Pro]
UC Froala Initialized [UC - Rich Text Editor Pro]
UC Froala Option Changed [UC - Rich Text Editor Pro]

Q&A

Twitter: @dgielis
Blog: dgielis.com
Email: dgielis@united-codes.com

Twitter: @united_codes
Website: www.united-codes.com

Plug-ins Pro
<https://www.plugin-pro.com>

The Educational Conference For Oracle Technology Users

ODTUG
Kscope23
aurora, co june 25-29



Interested in volunteering?
Scan the code to sign up



Don't Forget To
Fill Out Your Evals