

APEX Plug-in Development Done Right

Zsolt Angyal





Zsolt Angyal

@zsoltangyal

<https://www.foex.at>

zsolt.angyal@foex.at

Plug-in Development

Free Open Source

Internal Projects

Consulting

Football

Agenda

- Why plug-ins?
 - What're they?
 - Pros and Cons
- Types and categories of plug-ins
 - Internal, Third party
 - Free, Commercial
- Development process
 - Stages
 - Requirements
 - Difficulties
- FOS
 - What's this?
 - The idea behind(/purpose)
- Lessons we learned
 - Priorities
 - Tools
 - Documenting
 - Testing
 - Planning

Plug-ins... why?!

What're they good for...

- Introduce new feature
- Modularize already existing functionality into reusable components
- Increase development speed
- Update/debug in one place (centrally managed)
- Save time and cost

Drawbacks...

- Adds extra “risk” to the application
- Maintenance
 - Support/help
- Can be difficult to create
 - Few learning-sources/knowledge available
- Hard to create a visual component to fit the APEX context

Types

Plug-in Types

- Dynamic Action
- Region
- Items
- Process
- Authentication, Authorization
- REST Source

apex.world: (10.2021)

Process

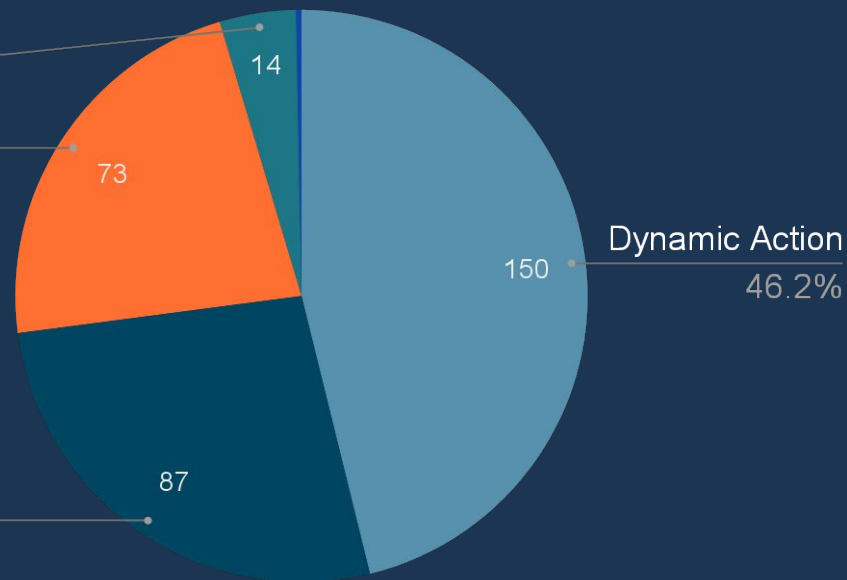
4.3%

Item

22.5%

Region

26.8%



Categories

Home-made

(internal use, built from scratch)

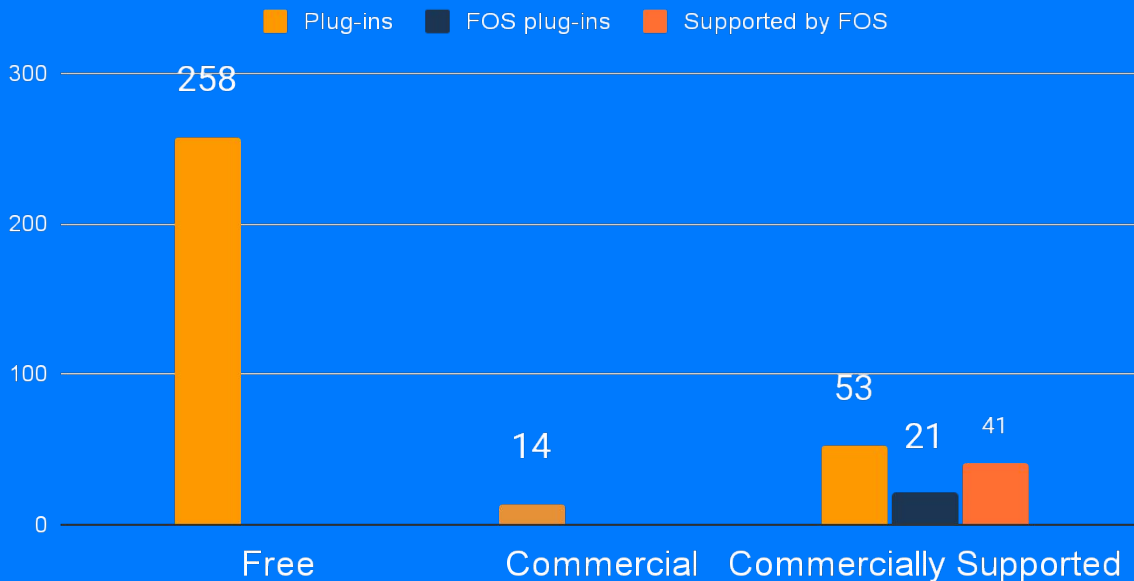
Third-party

(apex.world, GitHub,...)

Third party plug-in categories

- Free
 - No support guaranteed
- Commercial
- Commercially Supported

Plug-in categories with FOS



Free Open Source

General Information

- FOEX, 08.2020
- Biggest Open Source Project of such type
- FOS Browser Extension (by Stefan Dobre)
- 21 plug-ins created
- +40 plug-ins supported

What was the idea?

- Few open source (plug-in) projects in the community
- Reduce the risk of using a free third-party plug-in
- Provide a learning source
- Create quality plug-ins
- Give back to the community
- ...

Development Process

Stages

Planning

- Goal
- Schedule
- Structure
- Attributes

Implementing

- Coding Standards
- “Techniques”
- Rules

Testing

- PL/SQL
- JavaScript

Documenting

- Version Control
- Help Texts(!)
- Tools

Publishing

- Format
- Tools

Planning

Points to declare

- What's the problem we want to solve?
- Clear, well defined goals
- Necessary and optional features
- Tasks, schedule
- Stick to the plan!
- Tools:
 - Kanban board/Multi-list
 - Calendar/Scheduler

FOS Planning Board

Board
FOS - Image Slider

Front-end

50%

2/4

Frontend

☒ Add navigation

Frantisek Nagy

FEATURE

☒ Autoplay

Frantisek Nagy

FEATURE

☐ Display Thumbnails

Zsolt Angyal

FEATURE

☐ Fullscreen option

Back-end

33%

1/3

Backend

☐ BLOB source support

Zsolt Angyal

FEATURE

☐ Create the description box markup

Zsolt Angyal

FEATURE

☒ URL source support

Richard Baldogi

FEATURE

Bugs

66%

2/3

Backend

☐ Description text with URL source is not working

Zsolt Angyal

BUG

Frontend

☒ Arrows are not visible

BUG

☒ Split-view is brok

Frantisek Nagy

BUG

FOS Planning Board

Board
FOS - 21.1



Ideas

Dynamic Actions

Spinner Actions

Various spinner icons, configurable display location...

Tooltip

Visually appealing tooltip implementation with multiple features

Region

Monaco Editor



To Do

Item

Advanced Password

Zsolt Anyal

6 days

Region

Image Slider

Frantisek Nagy

6 days

PopupLOV Actions

Zsolt Anyal



In Progress

Dynamic Actions

Drag and Drop

Peter Raganitsch

2 days

The FOS - Drag and Drop dynamic action plug-in makes it possible to drag & drop elements inside a region. This could be a Cards region, a Badge list or basically everything that has a group of elements.



Testing

Item

Range Slider

Frantisek Nagy



Structure, attributes

- User-friendly attribute structure
- Same pattern across plug-ins
 - Attribute numbers
- Make it low-code!
- Third party library
 - Is it worth it
 - Do not reinvent the wheel

Attributes structure

- Do not waste
 - Always leave one free
- Clear, straightforward labels and options
- Helpt text to explain

Instead of this...

Show Spinner	<input type="checkbox"/>
Spinner Position	On Region
Overlay	<input type="checkbox"/>
Region ID	myRegion

... you can do this

Items to Return	No
	On Region
Show Spinner	✓ On Region with Overlay
	On Page
Lazy Load	On Page with Overlay

Make it comfortable for the user, do not ask for typing
if it's not necessary

Options(JSON)

```
{  
  "attributeOne": true,  
  "attributeTwo": true,  
  "attributeThree": false,  
  "attributeFour": "blue"  
}
```

Declare the options in advance, with default values

Options

- ☒ Attribute One
- ☒ Attribute Two
- ☐ Attribute Three

JavaScript Initialization Code

```
function(config){  
  config.attributeFour = 'blue';  
  config.notSoImportantAttribute = 1000;  
  config.overrideDefaultAttribute = '.foo';  
  return config;  
}
```

Attributes structure

- Keep it low code!
- Have default values
 - Quick start
- Try to “boolean” the attributes
- Basic and Advanced
- Use the native attributes
 - Javascript Initialization
 - Affected Elements
 - ...

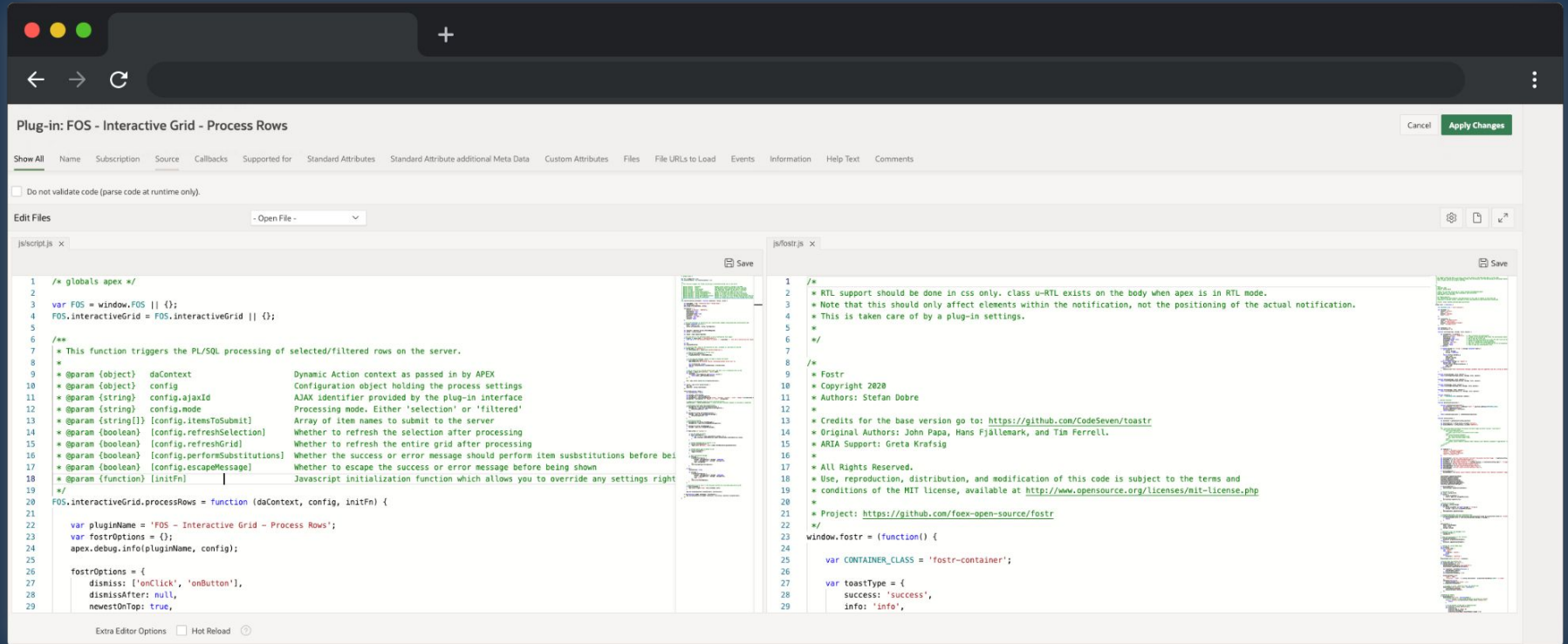
Implementing

Make things easier for yourself

- Tools
 - FOS Browser Extension
- Boilerplate Codes
- Coding Standards(!)
- Follow the rules

Create, update, delete, minify .js/.css files in directly in APEX

FOS Browser Extension



Boilerplate code, utility functions....

```
procedure htp_p_clob
(
  p_clob clob
)
as
  l_offset number;
  l_chunk varchar2(32767);
begin
  while apex_string.next_chunk
    (
      p_str => p_clob
      , p_chunk => l_chunk
      , p_offset => l_offset
      , p_amount => 30000
    )
  loop
    sys.htp.prn(l_chunk);
  end loop;
end;
```

```
1 function render
2 (
3   p_dynamic_action in apex_plugin.t_dynamic_action
4   , p_plugin        in apex_plugin.t_plugin
5 )
6 as
7   l_result apex_plugin.t_dynamic_action_render_result;
8
9   --attributes
10  l_attribute1 p_dynamic_action.attribute_01%type := p_dynamic_action.attribute_01;
11  l_attribute2 p_dynamic_action.attribute_02%type := p_dynamic_action.attribute_02;
12  l_attribute3 p_dynamic_action.attribute_03%type := p_dynamic_action.attribute_03;
13
14 begin
15
16   --debug
17   if apex_application.g_debug
18   then
19     apex_plugin_util.debug_dynamic_action
20     (
21       p_plugin => p_plugin
22       , p_dynamic_action => p_dynamic_action
23     );
24   end if;
25
26   apex_json.initialize_clob_output;
27
28   apex_json.open_object;
29   apex_json.write('l_attribute1', l_attribute1);
30   apex_json.close_object;
31
32   l_result.javascript_function := 'function(){myFunction(this, ' || apex_json.get_clob_output || ')}';
33
34   apex_json.free_output;
35
36   return l_result;
37 end render;
```

```
l_checkbox_attribute p_dynamic_action.attribute_01%type := p_dynamic_action.attribute_01;
l_checkbox_option_one boolean := instr(l_checkbox_attribute, 'option-one' ) > 0;
l_checkbox_option_two boolean := instr(l_checkbox_attribute, 'option-two' ) > 0;
l_checkbox_option_three boolean := instr(l_checkbox_attribute, 'option-three' ) > 0;
```

```
l_checkbox_attribute apex_t_varchar2 := apex_string.split(coalesce(p_dynamic_action.attribute_01,''),'');
l_checkbox_option_one boolean := 'option-one' member of l_checkbox_attribute;
l_checkbox_option_two boolean := 'option-two' member of l_checkbox_attribute;
l_checkbox_option_three boolean := 'option-three' member of l_checkbox_attribute;
```

Coding standards!

```
1 function render(p_region apex_plugin.t_region, p_plugin apex_plugin.t_plugin,  
2 | p_is_printer_friendly boolean  
3 )return apex_plugin.t_region_render_result  
4 as  
5 | l_result apex_plugin.t_region_render_result;  
6 l_attr1 p_region.attribute_01%type := p_region.attribute_01;  
7 | l_attribute2 p_region.attribute_02%type := p_region.attribute_02;  
8 | l_attribute_three p_region.attribute_03%type := p_region.attribute_03;  
9  
10 | l_region_id p_region.static_id%type := p_region.static_id;  
11 | l_ajax_id p_region.static_id%type := apex_plugin.get_ajax_identifier;  
12 | --perform escaping  
13 | l_region_id_esc p_region.static_id%type := apex_escape.html_attribute(l_region_id);  
14 begin  
15 | --debug
```

```
1 function render  
2 | ( p_region in apex_plugin.t_region  
3 | , p_plugin in apex_plugin.t_plugin  
4 | , p_is_printer_friendly in boolean  
5 | )  
6 return apex_plugin.t_region_render_result  
7 as  
8 | l_result apex_plugin.t_region_render_result;  
9  
10 | --attributes  
11 | l_attribute1 p_region.attribute_01%type := p_region.attribute_01;  
12 | l_attribute2 p_region.attribute_02%type := p_region.attribute_02;  
13 | l_attribute3 p_region.attribute_03%type := p_region.attribute_03;  
14  
15 | l_region_id p_region.static_id%type := p_region.static_id;  
16 | l_ajax_id p_region.static_id%type := apex_plugin.get_ajax_identifier;  
17  
18 | --perform escaping  
19 | l_region_id_esc p_region.static_id%type := apex_escape.html_attribute(l_region_id);  
20
```

Testing

You can never be 100% sure, but...

- Testing frameworks
 - utPLSQL, Cypress, etc...
- Use the plug-in
- The demo is the one of the best tests

Documenting

Save (future) work...

- Help Texts
- Demo
- Use APEX to make your job easier
- Version control
 - GitLab, GitHub

- Easy to forget some steps
- APEX views
-

Complete overview of help-texts:

- apex_appl_plugins
- apex_appl_plugin_attributes
- apex_appl_plugin_attr_values
- apex_appl_plugin_std_attrs

Plug-in Overview

Search: All Text Columns Go Actions Edit Save Add Row

Batch Demo Yap, Demo Nope

	Batch	Type	Internal Name	Display Name	Demo	Help Text
Batch: 1						
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.CLIENTSIDE.CONDITION	FOS - Client-side Condition	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.EXECUTE_PLSQL_CODE	FOS - Execute PL/SQL Code	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.INTERACTIVE_GRID_ADD_BUTTON	FOS - Interactive Grid - Add Button	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.INTERACTIVE_GRID_PROCESS_ROWS	FOS - Interactive Grid - Process Rows	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.MESSAGE_ACTIONS	FOS - Message Actions	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.NOTIFICATIONS	FOS - Notifications	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.TIMING_ACTIONS	FOS - Timing Actions	Yes	
<input type="checkbox"/>	1	Dynamic Action	COM.FOS.TRIGGER_EVENT	FOS - Trigger Event(s)	Yes	
<input type="checkbox"/>	1	Region Type	COM.FOS.PLSQL_DYNAMIC_CONTENT	FOS - PL/SQL Dynamic Content	Yes	
<input type="checkbox"/>	1	Region Type	COM.FOS.STATIC_CONTENT	FOS - Static Content	Yes	

Make it clear what is waiting for the user

... and again, use APEX to make your job easier

Settings

Mode

Page Designer

Rendering

☐

Tabs

Dynamic Actions

☒

Processing

☐

App ID

&APP_ID.

Page ID

&APP_PAGE_ID.

Show Components

Filter - Partial (starts with)

Filter

P1_EXAMPLE_ONE_

Height

600

Dynamic Actions

Help

Events

Page Load

Change

Click

Example 1 - Trigger Basic Custom Event

True

FOS - Trigger Event(s) [Plug-in]

False

Example 1 - Trigger Custom Event

Example 1 - Trigger Custom Multiple Events

Example 2 - Trigger Custom Event with "this.data" set

Example 2 - Trigger Custom Event with "this.data.object" set

Example 2 - Trigger Multiple Custom Events with "this.data.object" set

Example 3 - Trigger Custom Event and Cancel Following Actions

Example 4 - Trigger Custom Event - "Item is Not NULL" Condition

Example 4 - Trigger Custom Event - "Item is NULL" Condition

Example 4 - Trigger Custom Event - "Item = Value" Condition

Example 4 - Trigger Custom Event - "Item != Value" Condition

Example 4 - Trigger Custom Event - "Item is NULL" Condition

Example 4 - Trigger Custom Event - "Page is Valid" Condition

Action

Identification

Action

FOS - Trigger Event(s)

Settings

Event Name

custom-event

Data

None

Advanced Configuration

☒

Client-side Substitutions

☐

Event Condition

No Condition

Cancel Following Actions

☐

Set Page Item

Affected Elements

Selection Type

Triggering Element

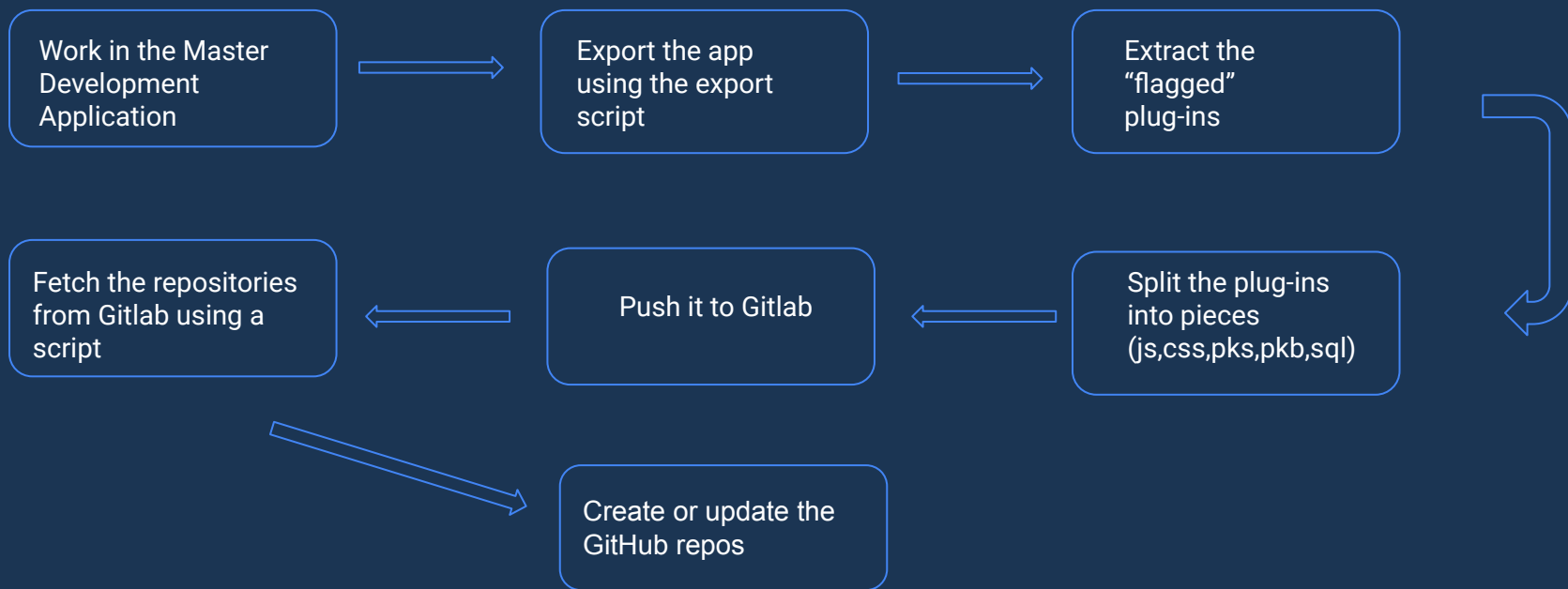
Execution Options

Publishing

Automate as many steps as possible

- Export script
- Publish script
- Gitlab for internal use and GitHub for the public
- Admin application
- Few manual steps

Automate as many steps as possible



Thank you