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

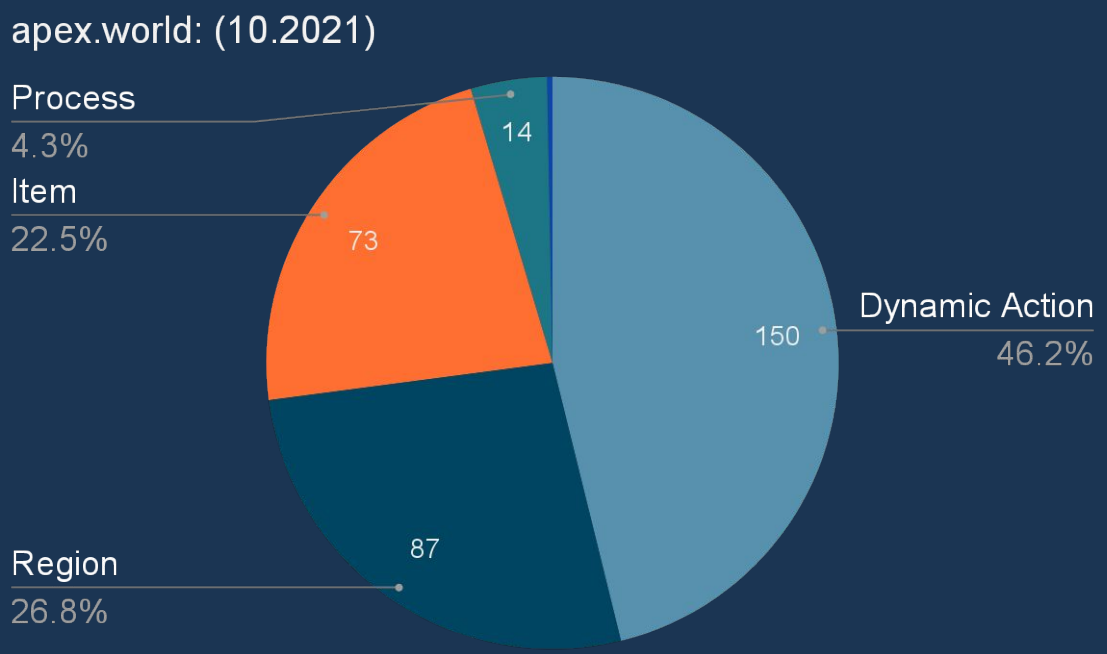
Drawbacks...

- 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



Categories

Home-made

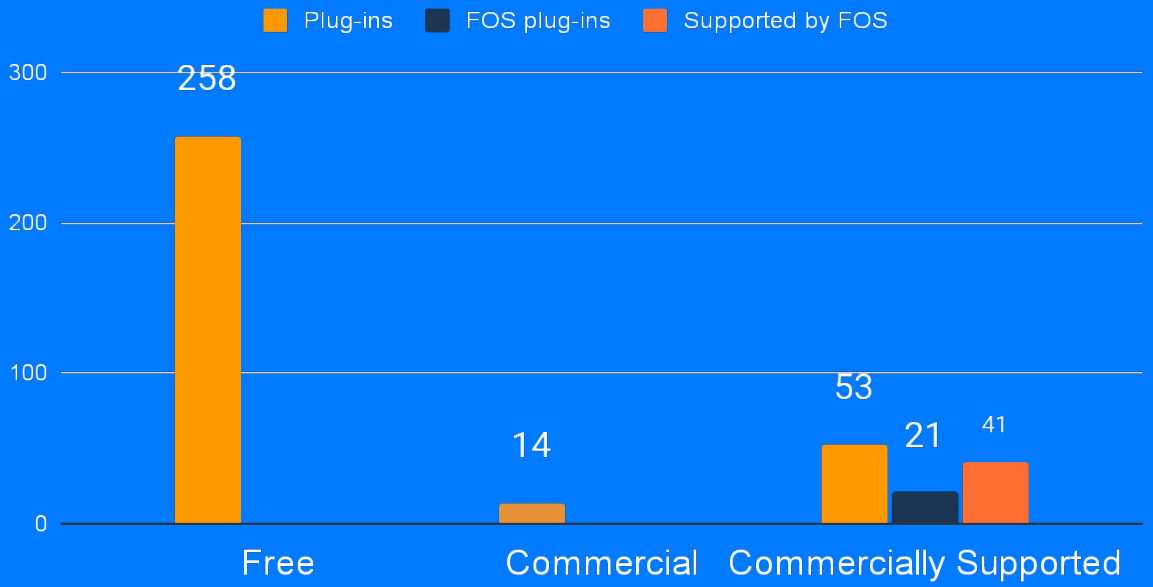
(internal use, built from scratch)

Third-party

(apex.world, GitHub,...)

Third party plug-in categories

- Free
  - No support guaranteed
- Commercial
- Commercially Supported



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	Implementing	Testing	Documenting	Publishing
- Goal	- Coding Standards	- PL/SQL	- Version Control	- Format
- Schedule	- “Techniques”	- JavaScript	- Help Texts(!)	- Tools
- Structure	- Rules		- Tools	
- Attributes				

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

2/4
50%

Frontend

- [x] Add navigation
  Frantisek Nagy
  FEATURE
- [x] Autoplay
  Frantisek Nagy
  FEATURE
- [ ] Display Thumbnails
  Zsolt Angyal
  FEATURE
- [ ] Fullscreen option

Back-end

1/3
33%

Backend

- [ ] BLOB source support
  Zsolt Angyal
  FEATURE
- [ ] Create the description box markup
  Zsolt Angyal
  FEATURE
- [x] URL source support
  Richard Baldogi
  FEATURE

Bugs

2/3
66%

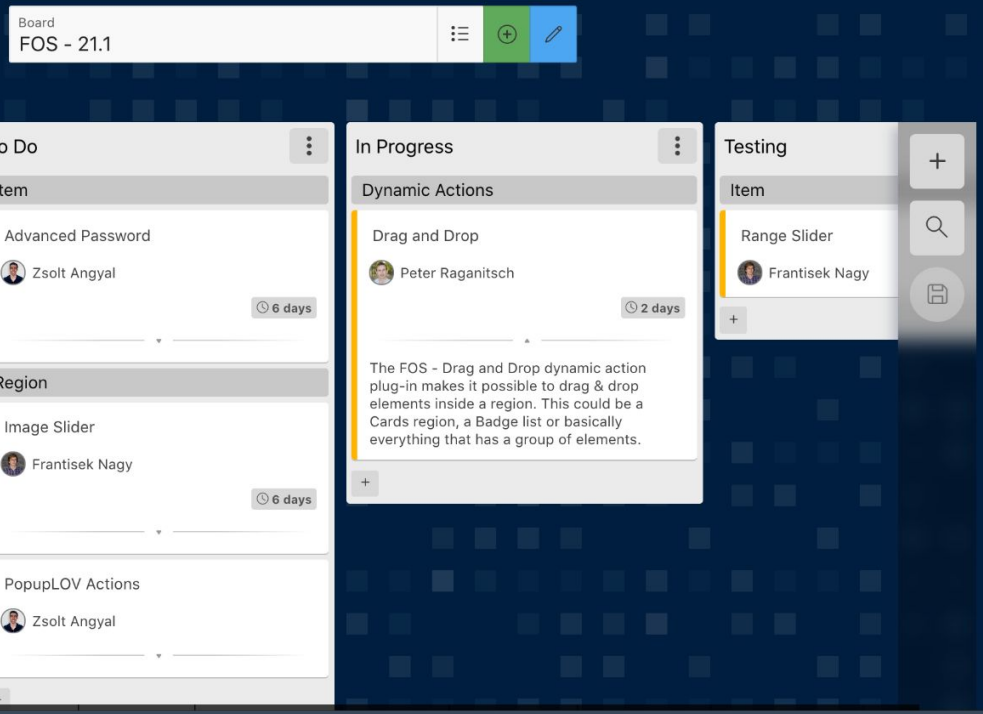
Backend

- [ ] Description text with URL source is not working
  Zsolt Angyal
  BUG

Frontend

- [x] Arrows are not visible
  BUG
- [x] Split-view is brok
  Frantisek Nagy
  BUG

FOS Planning Board



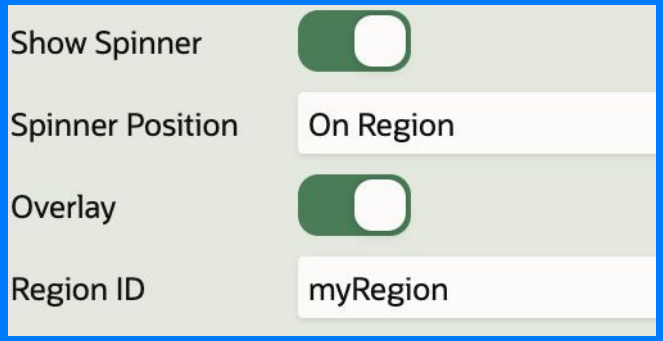
Structure, attributes

- 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

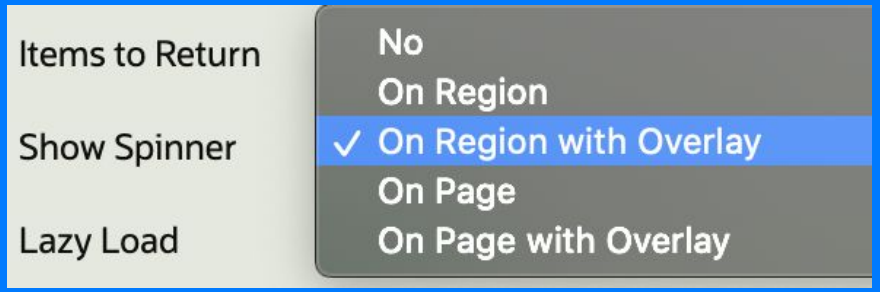
Attributes structure

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

Instead of this...



... you can do this

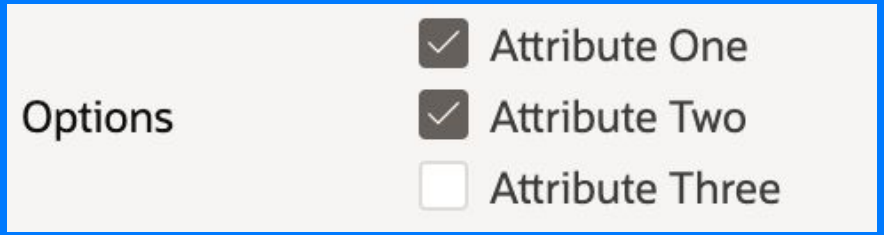


Make it comfortable for the user, do not ask for typing if it's not necessary

Options(JSON)

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

Declare the options in advance, with default values



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
  - ...

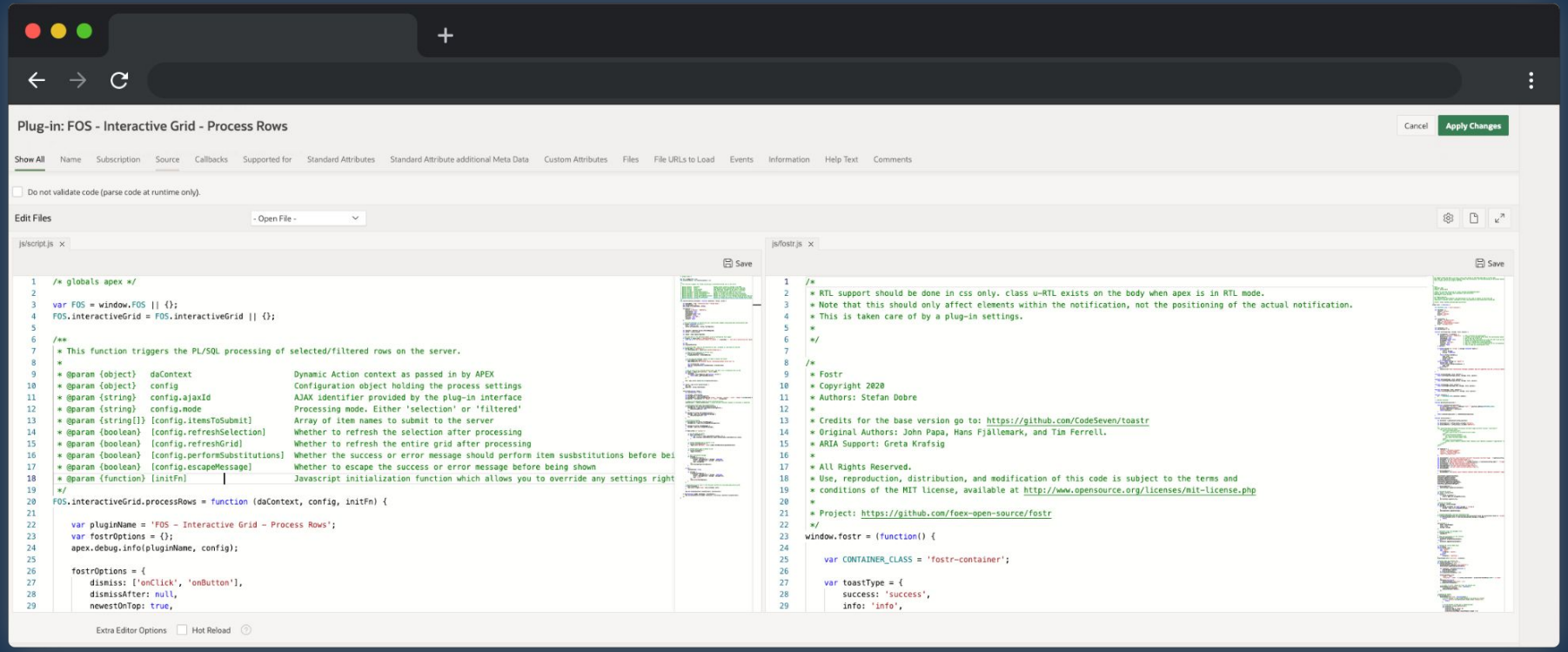
Implementing

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;
```

```
function render
  ( p_dynamic_action in apex_plugin.t_dynamic_action
  , p_plugin         in apex_plugin.t_plugin
  )
return apex_plugin.t_dynamic_action_render_result
as
    l_result     apex_plugin.t_dynamic_action_render_result;

    --attributes
    l_attribute1 p_dynamic_action.attribute_01%type := p_dynamic_action.attribute_01;
    l_attribute2 p_dynamic_action.attribute_02%type := p_dynamic_action.attribute_02;
    l_attribute3 p_dynamic_action.attribute_03%type := p_dynamic_action.attribute_03;

begin

    --debug
    if apex_application.g_debug
    then
        apex_plugin_util.debug_dynamic_action
          ( p_plugin         => p_plugin
          , p_dynamic_action => p_dynamic_action
          );
    end if;

    apex_json.initialize_clob_output;

    apex_json.open_object;
    apex_json.write('l_attribute1', l_attribute1);
    apex_json.close_object;

    l_result.javascript_function := 'function(){myFunction(this, ' ||  apex_json.get_clob_output || ');}';

    apex_json.free_output;

    return l_result;
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!

```
function render(p_region apex_plugin.t_region, p_plugin apex_plugin.t_plugin,
 p_is_printer_friendly boolean
)return apex_plugin.t_region_render_result
as
  l_result apex_plugin.t_region_render_result;
l_attr1 p_region.attribute_01%type := p_region.attribute_01;
    l_attribute2 p_region.attribute_02%type := p_region.attribute_02;
    l_attribute_three p_region.attribute_03%type := p_region.attribute_03;

    l_region_id     p_region.static_id%type    := p_region.static_id;
    l_ajax_id       p_region.static_id%type    := apex_plugin.get_ajax_identifier;
    --perform escaping
    l_region_id_esc p_region.static_id%type    := apex_escape.html_attribute(l_region_id);
begin
    --debug
```

```
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
as
    l_result        apex_plugin.t_region_render_result;

    --attributes
    l_attribute1    p_region.attribute_01%type := p_region.attribute_01;
    l_attribute2    p_region.attribute_02%type := p_region.attribute_02;
    l_attribute3    p_region.attribute_03%type := p_region.attribute_03;

    l_region_id     p_region.static_id%type    := p_region.static_id;
    l_ajax_id       p_region.static_id%type    := apex_plugin.get_ajax_identifier;

    --perform escaping
    l_region_id_esc p_region.static_id%type    := apex_escape.html_attribute(l_region_id);
```

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

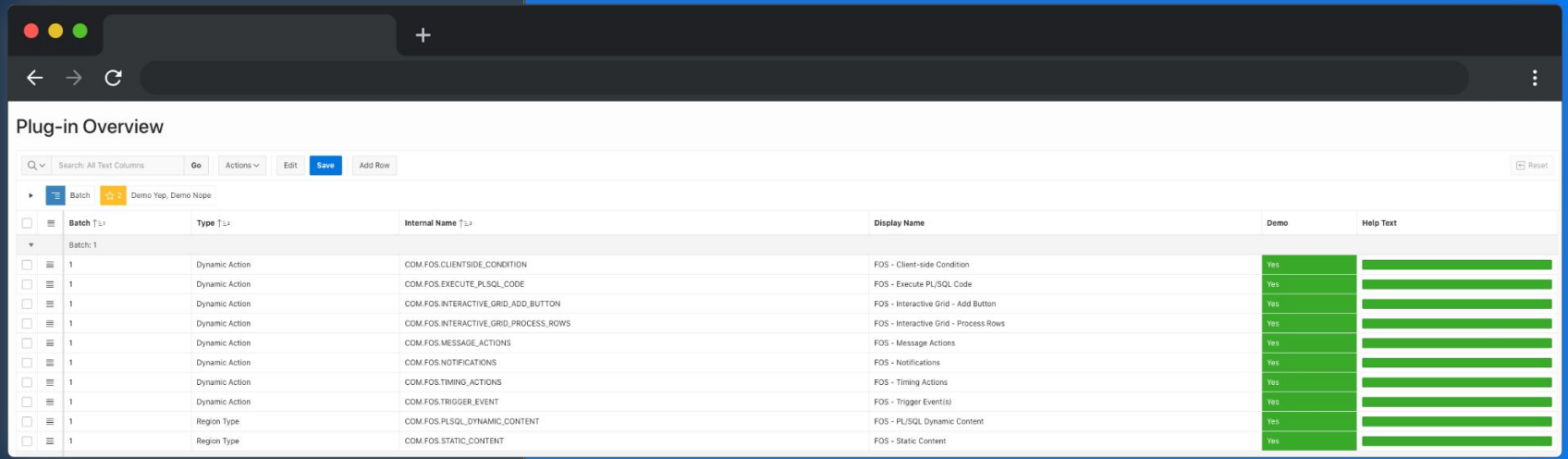
Save (future) work...

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

- Easy to forget some steps
- APEX views
-

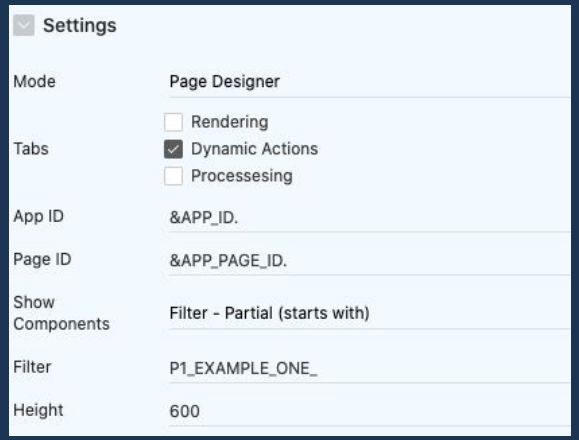
Complete overview of help-texts:

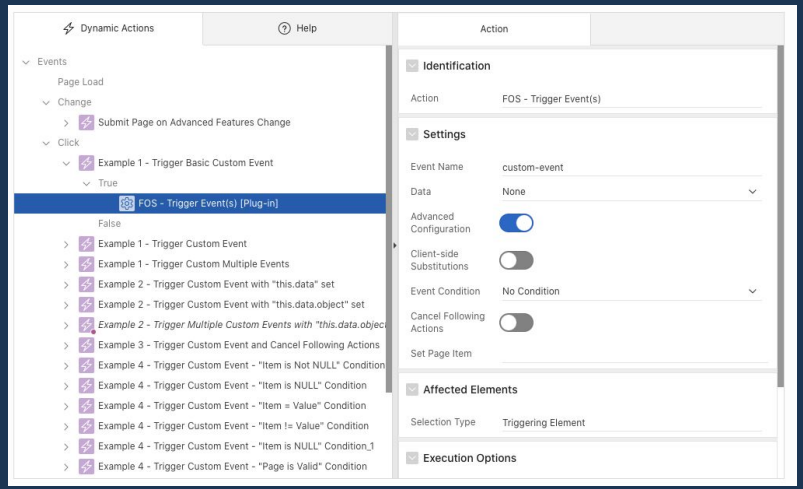
- apex_appl_plugins
- apex_appl_plugin_attributes
- apex_appl_plugin_attr_values
- apex_appl_plugin_std_attrs



Make it clear what is waiting for the user

… and again, use APEX to make your job easier



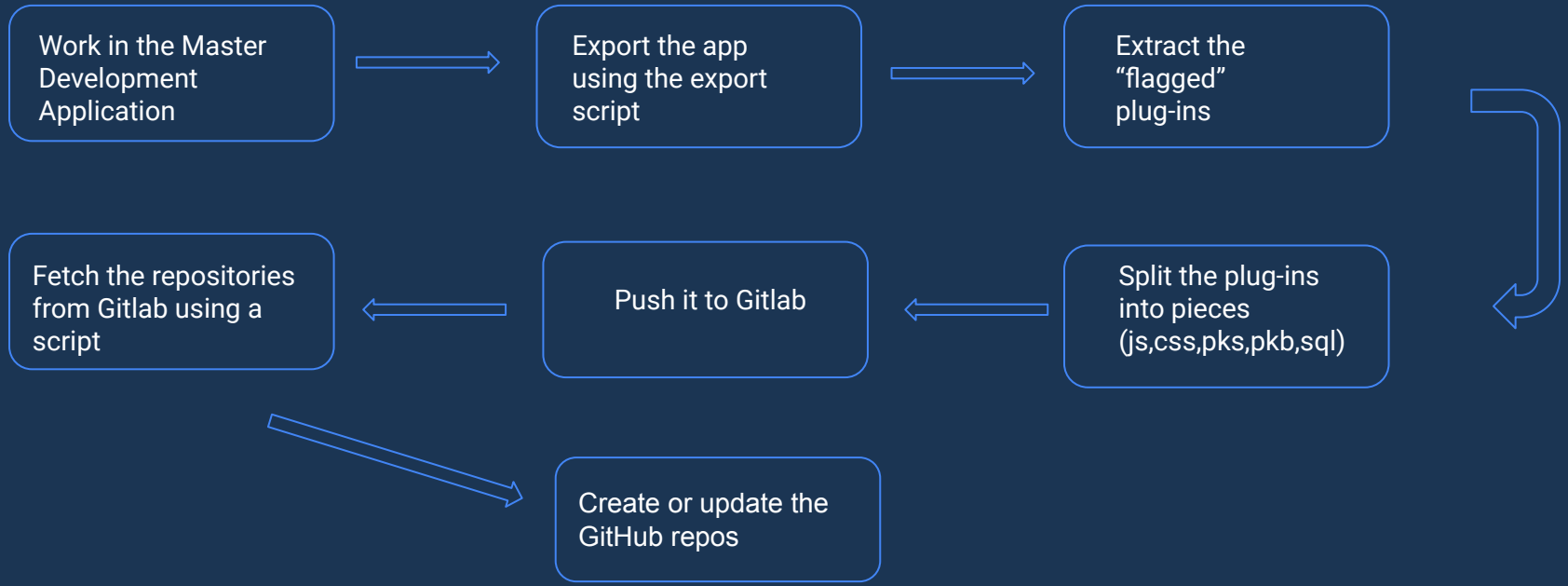


Publishing

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