APEX
Plug-In in PL/SQL

Mathias Magnusson, Evil Ape

Me

- Mathias Magnusson
- “Founder” of Miracle in Sweden
- Now at Evil Ape - just started
- Founder of SWEOUG - Sweden Oracle User Group
- oradbdev.mathiasmagnusson.com / @mathiasmag
- Oracle ACE Associate

Three goals today

- Build a plugin in PL/SQL
- Validate data entered into of Plug-In
- Build it live

What is a Plug-In

- Defining an element that can be used in APEX
- Often a page item, but can be
  - Authentication
  - Authorization
  - Dynamic Action
  - Region
  - Process
- Defined as a shared component in an application

Cloud

- We never hear about the cloud
- Why is there no cloud info at conferences
- When will Oracle stop talking on-prem

EXADATA Express

- Set up in seconds
- Ready to go with APEX
- Web page for user management
- Fastest way to get APEX running
- Much Improved interface

Creating the Plug-In

Lets set up an new plug-in

It will:

- Render a text field just lika APEX
- Validate that it is an email address

Note

All we need is to provide HTML

APEX features are included for free

Procedure names below code area

Validation

To validate entered data upon submit

Can be used to enforce any rule you wish

Errors are provided to the user as usual

Remember to specify procedure name

Summary

- Plug-In with just PL/SQL
- Provide HTML
- Validate Input
- Your fantasy is the limiting factor

APEXDOJO

Want to get involved and play?

Join the the APEXDOJO on GitHub

Not too advanced, but a bit deeper

Be part or just use to learn or for your system

A Secure Select List

Malicious use of it

DEMO

What could be done to remedy it

DEMO

What is done here?

The LOV can be 5 kinds

Represented in 5 ways in metadata

Validating by finding executing LOV SQL

Validated on submit

Cascade function using AJAX procedure

Lessons Learned

Documentation is JS focused

Learning is a mixture of blogs and testing

Asking other who have gone before you

Easy enough when you get the hang of it

Wrap Up

Plug-ins uses PL/SQL naturally.

It takes very little to get the basic parts up

Great way to give other devs reusable code

There is more than just page item plugins

Join the APEXDOJO and come to play with us

```
procedure render (
                  p_item   in            apex_plugin.t_item
                , p_plugin in            apex_plugin.t_plugin
                , p_param  in            apex_plugin.t_item_render_param
                , p_result in out nocopy apex_plugin.t_item_render_result) is
  l_name wwv_flow_plugin_api.t_input_name;
begin
  l_name := apex_plugin.get_input_name_for_item;

  sys.htp.prn (
               '<input type=text' ||
               wwv_flow_plugin_util.get_element_attributes(p_item, l_name ) ||
               ' value="' || apex_escape.html(v(l_name)) || '" ' ||
--             'class="text_field apex-item-text" ' ||
--             'style="border:5px solid #000000" ' ||
               '>'
              );
end render;
```

```
procedure validate (
                  p_item   IN apex_plugin.t_item
                , p_plugin IN apex_plugin.t_plugin
                , p_param  IN apex_plugin.t_item_validation_param
                , p_result IN OUT nocopy
apex_plugin.t_page_item_validation_result) is
begin
  if not nvl(v(p_item.name), '-') like '%@%.%' THEN
    apex_error.Add_error(p_message          => 'Incorrect email'
                        ,p_display_location =>
apex_error.c_inline_with_field_and_notif
                        ,p_page_item_name   => p_item.name);
  end if;
end validate;
```