



# Hello, UI Plugins!

Tutorial on writing simple plugin

Vojtech Szöcs

Software Engineer, Red Hat

January 24, 2013

# About the speaker



- oVirt UI maintainer
  - Web Admin
  - User Portal
- Deals with front-end application architecture
- Leads UI Plugins feature development

# In this tutorial, we will ...



- Fly through UI plugin basics
- Walk through main steps in plugin development
- Learn about supported API functions / events
- End up with a working plugin that you can play with
- Have fun!

# UI plugin basics



- Extend oVirt Web Admin user interface

oVirt Open Virtualization Manager

Logged in user: admin@internal | Configure | Guide | About | Sign Out

Search: Host: [x] [star] [magnifying glass]

Data Centers Clusters **Hosts** Networks Storage Disks Virtual Machines Pools Templates Volumes Users Events

Tree

Expand All Collapse All [refresh]

System

- Default
- MyDC
  - Storage
  - Networks
  - Templates
  - Clusters
    - MyCluster
      - Hosts
        - dev01aaa

- VMs

Bookmarks

Tags

New Edit Remove Activate Maintenance Configure Local Storage Power Management Assign Tags [refresh] 1-2 [left] [right]

Name	Hostname/IP	Cluster	Data Center	Status	Running VMs	Memory	CPU	Network	SPM
dev01aaa	10.34.63.161	MyCluster	MyDC	Up	0	16%	0%	0%	Normal
test	10.34.60.88	Default	Default	Maintenance	0	0%	0%	0%	Normal

Last Message: [check] 2013-Jan-09, 17:01 User admin@internal logged in. [Alerts (5)] [Events] [Tasks (0)]

# What's currently possible



The screenshot displays the oVirt Open Virtualization Manager interface. The top navigation bar includes tabs for Data Centers, Clusters, Hosts, Networks, Storage, Disks, Virtual Machines, Pools, Templates, Volumes, Users, and Custom Main Tab. The Hosts tab is active, showing a table of hosts with columns for Name, Hostname/IP, Cluster, Data Center, Status, Running VMs, Memory, CPU, Network, and SPM. A context menu is open over the 'test' host, listing actions such as New, Edit, Remove, Activate, Maintenance, Confirm 'Host has been Rebooted', Configure Local Storage, Assign Tags, and Custom Action Button. The Custom Action Button is highlighted in red. Below the table, the host configuration details are shown, including OS Version, Kernel Version, KVM Version, LIBVIRT Version, VDSM Version, SPICE Version, iSCSI Initiator Name, Active VMs, CPU Name, CPU Type, CPU Sockets, CPU Cores per Socket, CPU Threads per Core, Physical Memory, Swap Size, Shared Memory, Max free Memory for scheduling new VMs, Memory Page Sharing, and Automatic Large Pages. The Custom Sub Tab is also highlighted in red. The bottom status bar shows the last message, user login information, and alerts, events, and tasks.

Name	Hostname/IP	Cluster	Data Center	Status	Running VMs	Memory	CPU	Network	SPM
dev01aaa	10.34.63.161	MvCluster	MvDC	Up	0	16%	0%	0%	Normal
test	10.34.60.88			Maintenance	0	0%	0%	0%	Normal

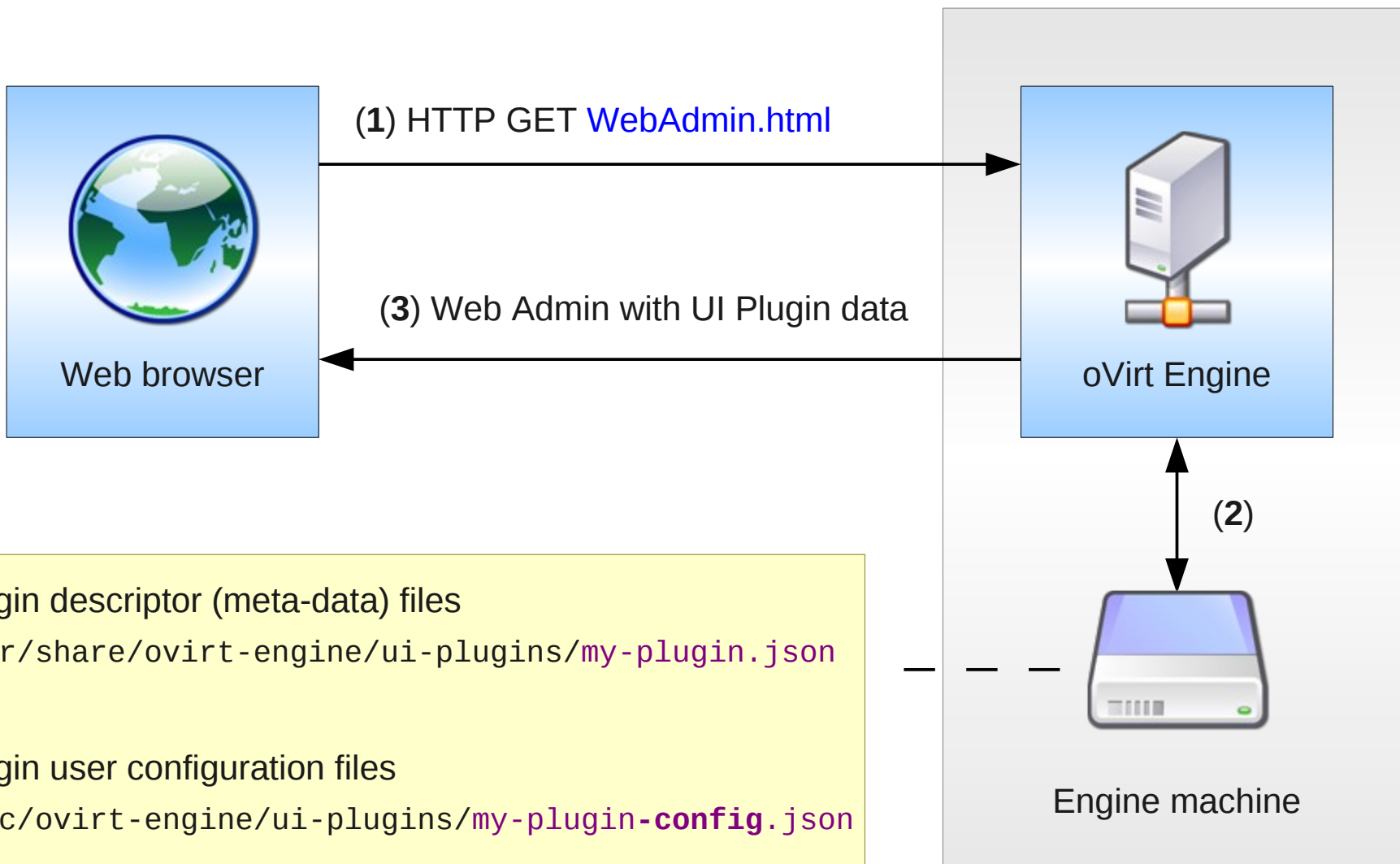
OS Version: RHEL - 6Server - 6.1.0.2.e  
Kernel Version: 2.6.32 - 131.6.1.el6.x86\_6  
KVM Version: 0.12.1.2 - 2.184.el6  
LIBVIRT Version: 0.9.4 - 7.2.el6  
VDSM Version: vdsm-4.9-104.el6  
SPICE Version: 0.8.2 - 3.el6  
iSCSI Initiator Name: iqn.1994-05.com.redhat:b

Active VMs: 0  
CPU Name: Intel Nehalem Family  
CPU Type: Intel(R) Xeon(R) CPU  
CPU Sockets: 1  
CPU Cores per Socket: 8  
CPU Threads per Core: Unsupported

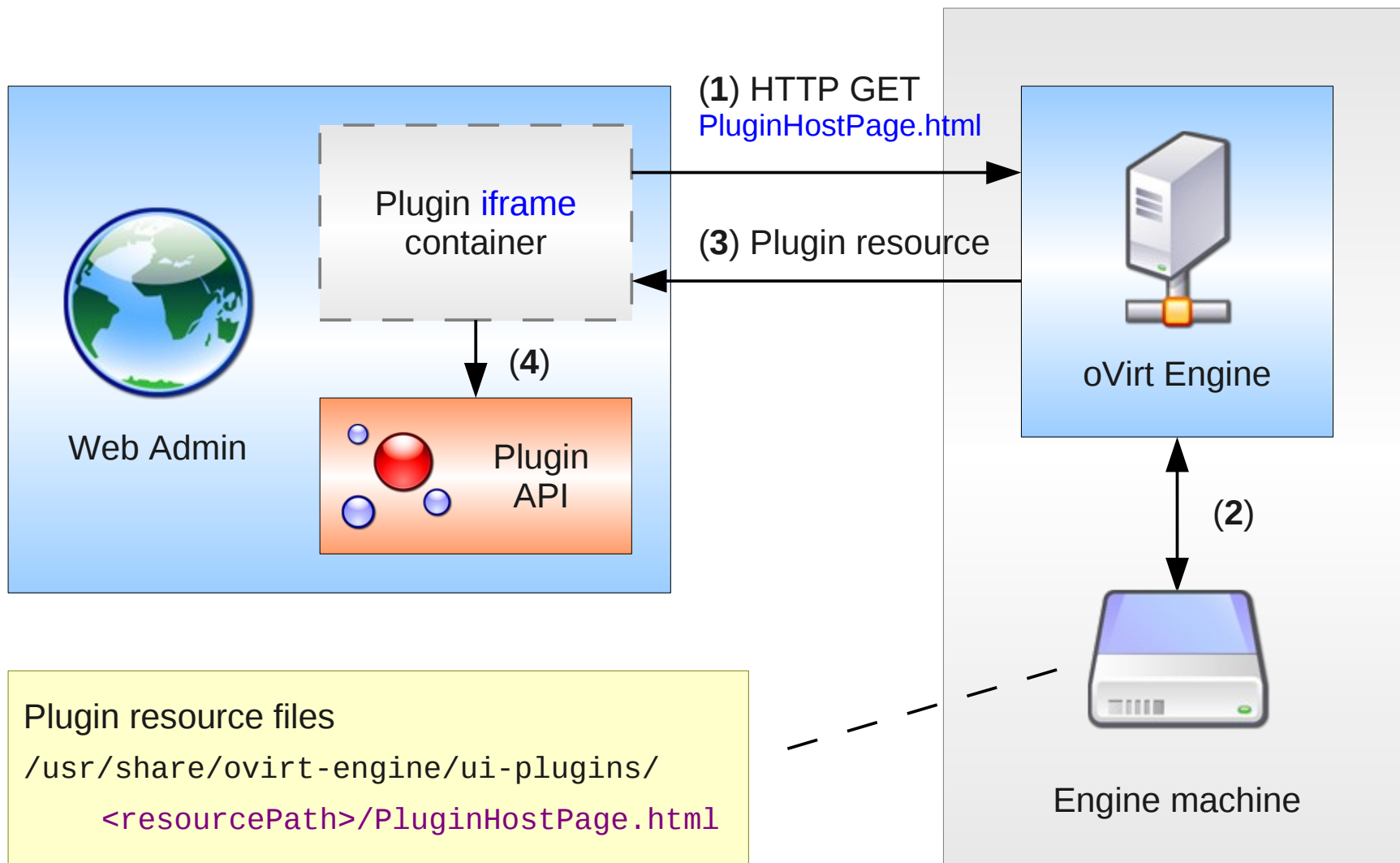
Physical Memory: 7861 MB total, 1258 MB use  
Swap Size: 1023 MB total, 0 MB used, 1  
Shared Memory: 0%  
Max free Memory for scheduling new VMs: 7605 MB  
Memory Page Sharing: Inactive  
Automatic Large Pages: Always

Action Items  
Power Management is not configured for this Host. [Enable Power Management](#)

# Discovering plugins



# Loading plugins



# Writing plugins



```
<!DOCTYPE html>
<html>
<head>
  <!-- Fetch additional resources if necessary -->
  <script type="text/javascript" src="jquery-min.js"></script>
```

```
<!-- Actual plugin code -->
<script>
  // Access plugin API from iframe context
  var api = parent.pluginApi('myPlugin');

  // Register plugin event handler functions
  api.register({
    UiInit: function() {
      api.addMainTab('Foo Tab', 'foo-tab', 'http://foo.com/');
    }
  });

  // Tell plugin infrastructure that we are ready
  api.ready();
</script>
```

```
</head>
<body> <!-- HTML body is intentionally empty --> </body>
</html>
```



# UI plugin basics



- Plugin descriptor
  - Meta-data + default configuration  
`/usr/share/ovirt-engine/ui-plugins/<descriptorName>.json`
- Plugin user configuration
  - Override default configuration, tweak runtime behavior  
`/etc/ovirt-engine/ui-plugins/<descriptorName>-config.json`
- Plugin host page
  - Hosts actual plugin code (JavaScript)  
`/usr/share/ovirt-engine/ui-plugins/<resourcePath>/<hostPage>.html`

# Plugin descriptor



- Meta-data + default configuration

```
/usr/share/ovirt-engine/ui-plugins/<descriptorName>.json
```

```
{
```

```
// A name that uniquely identifies the plugin (required)
```

```
"name": "foo",
```

```
// URL of plugin host page that invokes the plugin code (required)
```

```
"url": "/webadmin/webadmin/plugin/foo/start.html",
```

```
// Default configuration object associated with the plugin (optional)
```

```
"config": { "band": "ZZ Top", "classic": true, "score": 10 },
```

```
// Path to plugin static resources (optional)
```

```
// Used when serving plugin files through PluginResourceServlet
```

```
// This path is relative to /usr/share/ovirt-engine/ui-plugins
```

```
"resourcePath": "foo-files"
```

```
}
```

# Plugin user configuration



- Override default configuration, tweak runtime behavior

/etc/ovirt-engine/ui-plugins/<descriptorName>-**config.json**

```
{  
  
  // Custom configuration object associated with the plugin (optional)  
  // This overrides the default plugin descriptor configuration, if any  
  "config": { "band": "AC/DC" },  
  
  // Whether the plugin should be loaded on WebAdmin startup (optional)  
  // Default value is 'true'  
  "enabled": true,  
  
  // Relative order in which the plugin will be loaded (optional)  
  // Default value is Integer.MAX_VALUE (lowest order)  
  "order": 0  
}
```

# Runtime plugin configuration



- Merge user configuration (if any)  
on top of default configuration (if any)

```
{ "band": "ZZ Top", "classic": true, "score": 10 }
```

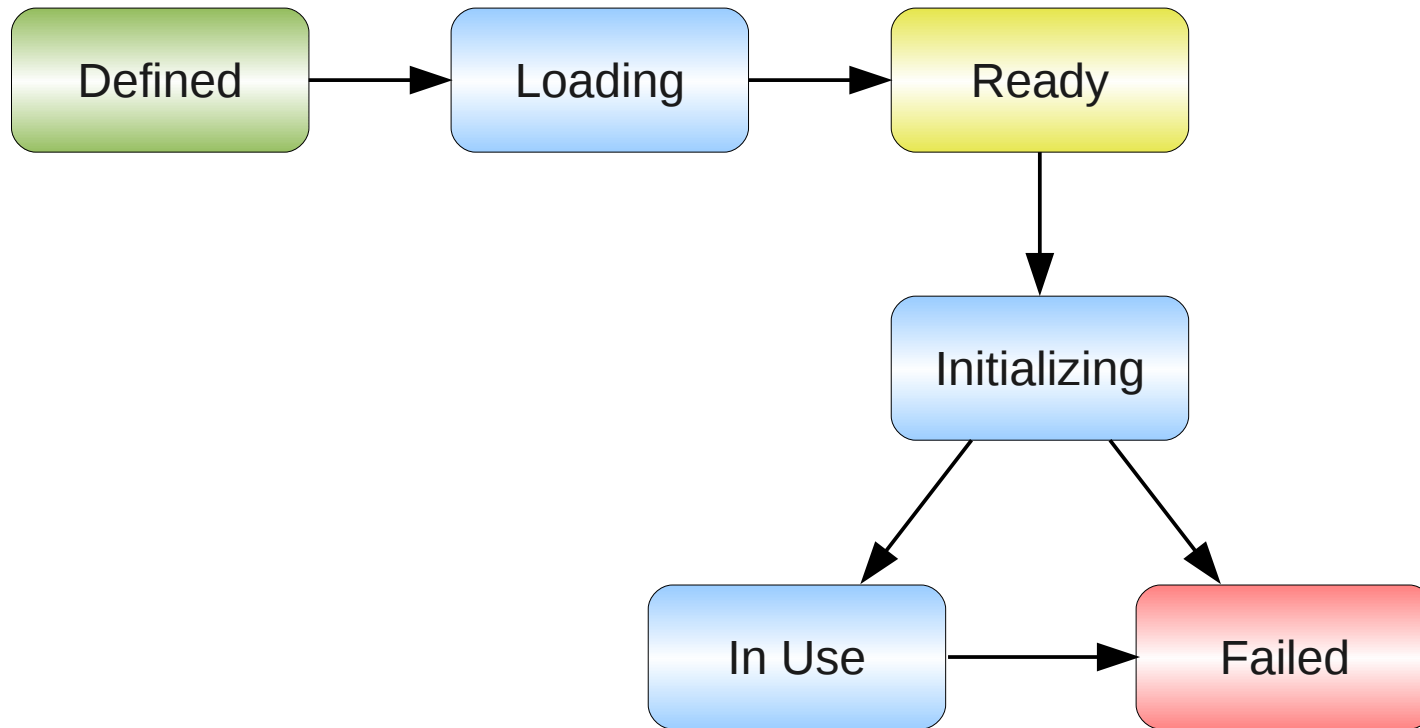
+

```
{ "band": "AC/DC" }
```

=

```
{ "band": "AC/DC", "classic": true, "score": 10 }
```

# Plugin lifecycle



# Main steps in plugin development



- (1) Write plugin descriptor
- (2) Write plugin host page
- (3) See plugin in action



# Supported API functions



- `addMainTab(label, historyToken, contentUrl)`
- `addSubTab(entityTypeName*, label, historyToken, contentUrl)`
- `setTabContentUrl(historyToken, contentUrl)`
- `setTabAccessible(historyToken, tabAccessible)`

String

Boolean

Number

Object

\* supported values

Cluster, DataCenter, Disk, Host,  
Storage, Template, VirtualMachine

# Supported API functions



- `addMainTabActionButton(entityTypeName*, label, buttonInterface)`
- `showDialog(title, contentUrl, width, height)`
- `loginUserName()`
- `loginUserId()`

String

Boolean

Number

Object

\* supported values

Cluster, DataCenter, Disk, Host,  
Storage, Template, VirtualMachine



# Supported API events



- UserLogin(`fullUserName`, `userId`)
- UserLogout()
- RestApiSessionAcquired(`sessionId`)
- {entity\*}SelectionChange(selectedItems[])

String

Boolean

Number

Object

\* supported values

Cluster, DataCenter, Disk, Host,  
Storage, Template, VirtualMachine

# Useful links



- <http://www.ovirt.org/wiki/Features/UIPlugins>
- <git://gerrit.ovirt.org/samples-uiplugins>
- <http://lists.ovirt.org/mailman/listinfo/engine-devel>



# Thank you!

<http://www.ovirt.org/>

vszocs@redhat.com

vszocs at #ovirt (irc.oftc.net)