

SAIL POINT PLUGINS

At the surface, it's a webpage - HTML, CSS, JS

- Some use AngularJS

Underneath, there's Java - classes, inputs, services, controllers, endpoints, etc.

- Some new annotations
- Some new classes/interfaces
- will import SailPoint libraries
- Can hit these Plugin endpoints internally AND externally via Postman or another FE/API

STRUCTURE

- ① MANIFEST - an XML document telling SP what's in our Plugin
- REST resources (active endpoints)
 - Snippets (active UI elements, etc.)
 - Executors
 - Basic plugin info
 - name
 - version
 - min upgradeable version

- ② BUILD FILE - An XML file
- tells Apache Ant how to construct the build of your plugin for use by SailPoint
- ★ - Can use stock build.xml file for most instances
- compiles classes
 - creates a folder structure
 - creates a .jar file
 - ZIPS up the whole thing into something SP can digest

- ③ DB SETUP/BREAKDOWN SCRIPTS (UPGRADE ALSO) - .mysql scripts to run when you install/uninstall/upgrade a plugin
- db/install/install.mysql
db/uninstall/uninstall.mysql
"/upgrade/upgrade.mysql

- ④ UI - HTML, CSS/JS possible

★ AngularJS optional

- page.xhtml is the base file for the plugin's visuals
- can have others for secondary pages, modals, etc.
- can include CSS on the page or link to it
- can run JS scripts to add that functionality
- via your JS, you can access the Java endpoints

- ⑤ REST RESOURCES - Provide endpoints you can hit to utilize plugin features
- POST, GET, PUT, DELETE [C R U D]
 - point to Service classes to run your logic and execute your queries
 - NOT using a repo here!!
 - manual queries
 - CAN use models to conform your data

SERVICES include logic & queries, unlike SPs

- ⑥ EXECUTORS - 3 kinds - Service Policy Task
- take care of other functionality OUTSIDE of basic REST operations

- ⑦ XML Artifacts - Can include new XML Objects/Types with our plugin
- if we need a specific policy, alert, task, etc. that ONLY this plugin will use, we can include it with our installation
 - import/install/artifact.xml
"/upgrade/etc...

PROCESS -

- ① Develop your plugin
 - adding libraries to the classpath
 - create all the elements/structures
- ② Run your build.xml file via Ant
- ③ Take the resulting ZIP and import it into SP

Gear Menu → Plugins → New+

★ MAI require Tomcat restart...