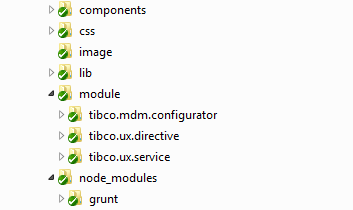
Architecture Specification

Files structure

Following is the screenshot of the Files Archtexture



*Figure 1*

Now in the APP there are 6 main folder to manager the Configurator application.

What's in the each folder?

"components" folder is include all the useful and reuse components developed by self.

"css" folder is the style sheet folder, include all the common css files

"image" folder include all the images in the application.

"lib" folder include all the third-part lib which app dependent.

"module" folder is the mainly develop folder, include all the directive and services developed by self and various module, as shown in Figure 2

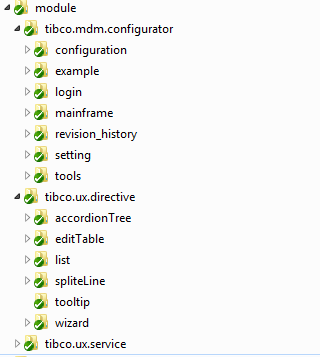


Figure 2

"node\_module" folder is the Engineering command line build tools--GruntJs, we use this tool to build and assist develop work. And will use this tool in automated testing.

The code architecture

All the things will start with "index.html"

In this file will load all the css style sheet, required js files and the localization source files according to the "config.xml".

Configurator is a Single Page Application, so how does it manage and load various modules?

AngularJs has a route service to implement this. We can use it config the routing rule like below code:

"mainframe.js"

$routeProvider.when('/:sectionName', {

templateUrl : function(parameter) {

var templateURL = "module/tibco.mdm.configurator/" + parameter.sectionName + "/view/template.html";

return templateURL;

}

}).when('/:sectionName/:subSectionName', {

templateUrl : function(parameter) {

var templateURL = "module/tibco.mdm.configurator/" + parameter.sectionName;

if(parameter.subSectionName!=null&&jQuery.trim(parameter.subSectionName !== "")) {

templateURL += "/" + parameter.subSectionName;

}

templateURL = templateURL + "/view/template.html";

return templateURL;

}

})

.otherwise({ // by default, show configuration panel

redirectTo : '/login'

});

"index.html"

<div class="bodyDiv">

<div class="headerDiv">

... ...

</div>

<div ng-view class="contentDiv"></div>

</div>

<div class="footerDiv">

<span class="copyRight">{{local.footer\_copyRightText}}</span>

</div>

When the url changed, The route service will use <div ng-view class="contentDiv"></div> load the template which the url direct and execute related controller to control data to display.

Build and testing

We use Grunt to build and as the drver of automated testing.

And use Jasmine to do the test.

Plan:

Optimize the architecture, make it to easy develop and testing.

1. separated the directive.js to more js file according its functionality, so that we can easy to develop it together and easy do the unit test
2. separated the service.js to more js file according its functionality, so that we can easy to develop it together and easy do the unit test
3. Do some readjustment to implement modularization development.
4. Use GruntJs to simplify the development process and testing process.