### FMS UI Section

* UI File locations
  + Javascripts Files - FMS\_UI\public\scripts\modules\sample-module
  + HTML Files - FMS\_UI\public\views
  + CSS Files – FMS\_UI\public\stylesheets
  + Node-modules – Folder gets added only after npm installation.
  + bower\_components – Folder contains all the JS components library
* Basic UI Flow
  + Index.html – Page is divided 4 parts
    - Header includes GE logo & Project Name
    - Navigation includes Navigation Menu & Login Details
    - Main section
    - Footer section
  + App.js
    - Includes tabs details
  + Main.js
    - Includes JS file name which contains controllers.
  + Routes.js
    - state provider i.e. state, url, templateUrl & controller name
    - url provider for default redirection
  + Config.js
    - Includes entry for new component like highchart, datatables, scroll, openlayer maps etc

**FMS Java Section**

* Java Files
  + Main File – FMSMain.java
  + Controller File – FMSController.java

**FMS DB Section**

****

**FMS Deployment Details**

* Connect the server

set http\_proxy=http://sjc1intproxy01.crd.ge.com:8080

set https\_proxy=http://sjc1intproxy01.crd.ge.com:8080

set HTTP\_PROXY=http://sjc1intproxy01.crd.ge.com:8080

set HTTPS\_PROXY=http://sjc1intproxy01.crd.ge.com:8080

set no\_proxy=api.grc-apps.svc.ice.ge.com,login.grc-apps.svc.ice.ge.com,loggregator.grc-apps.svc.ice.ge.com,uaa.grc-apps.svc.ice.ge.com,console.grc-apps.svc.ice.ge.com,$no\_proxy

set path=C:\app\Administrator\product\11.2.0\client\_1\bin;C:\oracle\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files\TortoiseSVN\bin;C:\Program Files\CloudFoundry;C:\Users\pp811725\Documents\Userdata\ProjectSoftwares\nodejs;C:\Program Files\Git\bin;C:\Users\pp811725\AppData\Roaming\npm;C:\Users\pp811725\Documents\Userdata\ProjectSoftwares\Predix2.0\jdk1.8.0\_60\bin;

set JAVA\_HOME = C:\Users\pp811725\Documents\Userdata\ProjectSoftwares\Predix2.0\jdk1.8.0\_60

cf login -a <https://api.system.aws-usw02-pr.ice.predix.io>

Enter Email & Password

* Deployment of Backend Code
  + Go to Code path e.g. <SYSTEM\_PATH>\git\FMS\_SERVICES\FMS
  + Update the name & path in Manifest.yml e.g. fms-dev & target\FMS-0.0.1-SNAPSHOT.jar.
  + The name mentioned in manifest.yml file will be used for url. E.g. <https://fms-dev.run.aws-usw02-pr.ice.predix.io/>
  + Update DB related changes in application.properties file.
  + Once code is ready for deployment, run the command “cf push”.
* Deployment of UI Code
  + Go to Code path e.g. <SYSTEM\_PATH>\git\FMS\_UI\FMS\_UI
  + Manifest.yml
    - Update the name e.g. fms-dev-ui
    - The name will be used for url. E.g. <https://fms-dev-ui.run.aws-usw02-pr.ice.predix.io>
    - Services – name of the services which will bind to application
    - env
      * Run the command “cf env fms-dev-ui” to get the below details.
      * UAA\_SERVER\_URL – Bind to application
      * REDIS – Name of redis which will be bind to application
  + Nginx.conf
    - Verify the below UAA parameter
      * session\_name
      * session\_secret
      * client\_id
      * uaa\_authorization\_header
      * user\_token
    - Add/Update the URL for the Bookmark service
  + Connect.js
    - Update backend service url once backend server is up
    - Update UAA details as per above file changes
  + Copy.js
    - List down all required files
  + Dist.js
    - Comment “jshint” & “test” entry.
  + Once code is ready for deployment, run the command “grunt dist”
  + It will create www folder under dist folder.
  + Once compilation done successfully, run the command “cf push”.
* Deployment of DB
  + Common application to manage the database In cloud is phpPgAdmin & Pg\_Studio
  + Configuration of PG\_Studio
    - Open cmd
    - Give the following proxies
    - set http\_proxy=http://sjc1intproxy01.crd.ge.com:8080
    - set https\_proxy=http://sjc1intproxy01.crd.ge.com:8080
    - set HTTP\_PROXY=http://sjc1intproxy01.crd.ge.com:8080
    - set HTTPS\_PROXY=http://sjc1intproxy01.crd.ge.com:8080
    - set no\_proxy=api.grc-apps.svc.ice.ge.com,login.grc-apps.svc.ice.ge.com,loggregator.grc-apps.svc.ice.ge.com,uaa.grc-apps.svc.ice.ge.com,console.grc-apps.svc.ice.ge.com,$no\_proxy
    - set path=C:\app\Administrator\product\11.2.0\client\_1\bin;C:\oracle\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files\TortoiseSVN\bin;C:\Program Files\CloudFoundry;C:\Users\pp811725\Documents\Userdata\ProjectSoftwares\nodejs;C:\Program Files\Git\bin;C:\Users\pp811725\AppData\Roaming\npm;C:\Users\pp811725\Documents\Userdata\ProjectSoftwares\Predix2.0\jdk1.8.0\_60\bin;
    - set JAVA\_HOME = C:\Users\pp811725\Documents\Userdata\ProjectSoftwares\Predix2.0\jdk1.8.0\_60
    - **clone the git url** : git clone <https://github.com/john-k-ge/pg_studio_1.2_cf>
    - pg\_studio\_1.2\_cf folder will be created in the userspace.
    - Go to that location : cd C:\(userspace)\pg\_studio\_1.2\_cf
    - cf login -a <https://api.system.aws-usw02-pr.ice.predix.io>
    - Email & password
    - Select the space , If available
    - Modify the manifest.yml available in pg\_studio\_1.2\_cf
    - Cf push
    - Cf a (get the app name, select your application as the name specified in your manifest.yml)
    - Cf s ( get the service name, postgres service)
    - Bind the application using : cf bs (appname) (service name) (service is already bound)
    - Restage the app if required (hint will be provided) cf restage (appname)
    - To get the credentials to connect to DB use cf env appname
    - For more info , refer REFER <https://www.predix.io/support/article/KB0010829>
  + Go to the url : <https://my-pg-studio-fms.run.aws-usw02-pr.ice.predix.io> (pg\_studio) or <https://phppgadmin-dtsfms-dev.run.aws-usw02-pr.ice.predix.io> (phppgadmin)
  + Give the required credentials , ‘Database Host’, ‘Database Port’, ‘Database Name’, ‘Username’ and ‘Password’ (PG\_Studio).
  + Cloud DB can be accessed.
* Common commands which will be used while deployment
  + cf s / cf services
    - List of services
  + cf a
    - List of application
  + cf bs <app\_name> <service\_name> / cf bind-service <app\_name> <service\_name>
    - Bind the service to application name
  + cf logs <app\_name> --recent
    - To see the logs
  + cf env <app\_name>
  + cf unbind-service <app\_name> <service\_name>
    - Unbind the service to application name
  + cf restage <app\_name>
  + cf
    - Help/List of all the commands