**I2File Login Flow**

1. On click of login button from “login.component.html”, login() function will be called after all mandatory fields are filled and login form is validated successfully.
2. Login() function will be executed in “login.component.ts” and will invoke a service named “authentication.service.ts”, which will execute a parameterize login() function having username and password as it’s respective parameter declared in “user.ts”.
3. The parameterize login() function will send http post request to the server endpoint with json object and RequestOption.
4. The http request will be received by “AuthenticateUserController.java” class.
5. The “ApplicationFilter.java” class will decrypt the json data and set jsondata as a attribute in HttpServletRequest .
6. “AuthenticateUserController.java” class will execute login() function which returns “ResponseObject.java” class object.
7. The execution cycle for login() function in “AuthenticateUserController.java” class is as follows :

* Through HttpServletRequest object, jsondata will be retrieve and will be mapped with “User.java” class with help of “ObjectMapper” class .
* Now, “UserService.java” interface has a method named “authenticateuser” which is implemented by “UserServiceImpl.java” class, the method will take user object as a parameter and authenticate the user by invoking “authenticateuser” method through “EFMClient.java” class object which implements Tyler service and returns “AuthenticateResponseType” object.
* The “authenticateuser” method of “UserServiceImpl.java” will store JsonRequest and JsonResponse in a table named “efmaudittrail”. If response code is “0” it will store last\_access\_time and user email and is\_i2file\_user flag in table name “loginhistory”.
* Before Storing in “loginhistory” table it will check userExists in a table named “userprofile” and also checks userExists in “loginhistory” table.
* If userExists in “loginhistory” table it will update only last\_access\_time field, else insert a new record.
* If userExists in “userprofile” table it will update, else insert a new record.
* The response data and status is encrypted and send by the server endpoint to the client.

1. On client side “authentication.service.ts” will receive the response . It will decrypt the data and store it in “mainResponseModel.ts” and return status and response object to “login.component.ts”.
2. If status is true then the “UtilityService.service.ts” object will set “mainResponseModel” by calling “setMainResponseModel” method.
3. Now, Router will navigate to the dashboard.