**Sample code documentation:**

**(Text in blue is added explaining method description)**

Document created for each methods

1. Component method

2. Service method

**Component Level: User Form Component**

We took User module as example for explaining the code documentation

|  |
| --- |
| /\*\*  \* Desc : Button click event to create/update a san-group  \* @namespace xio.SangroupFormComponent  \* @method onSubmit  \* @param {Object} e - btn click event  \* @return {void}  \*/  onSubmit(e) {  const progressRef = this.dialog.open(XioProgressComponent, { disableClose: true });  progressRef.componentInstance.progress\_data = 'processing'; this.sangroupForm.value.cb = (res, err) => {  if (res) {  this.navigateTo(e, '/san-group');  this.snackbarService.toastMe('SAN Group' + (this.isAddForm ? ' Created' : ' updated') + ' Successfully', 2000);  progressRef.close();  this.store.dispatch(new GetMenuAction({}));  }else if (err) {  progressRef.close();  if (this.isAddForm) {  this.sangroupForm['controls'] ['sangroup\_name'].setErrors({ remote: true });  }  }  };  this.store.dispatch((this.isAddForm ? (new AddSanGroupAction(this.sangroupForm.value)) : (new UpdateSanGroupAction(this.sangroupForm.value))));  } |

**Service Level: User Service**

|  |
| --- |
| /\*\*  \* Desc :getMenuContent All user's data  \* Method type : Get  \* URL : api/user/  \* @namespace xio.UserService  \* @method getAll  \* @return {Observable} http  \*\*/  getAll() {  return this.get(this.actionUrl)  .map((res: Response) => res.json().result.response.data)  .catch((error: any) => Observable.throw(error || 'Server IseError'));  } |

**Type Definition:**

Each module will have a file named <module\_name>.ts inside “model” folder which describes the type definition

Define Model Object - With respective Types :

Below model was implemented on part of code documentation for user management module

|  |
| --- |
| export interface User {  id: Number;  first\_name: string;  last\_name: string;  username: string;  email: string;  password: string;  } |

**Inside Reducer:**

Below code in blue highlists usage of type definition defined in model folder (in previous step).

|  |
| --- |
| export interface State{  user\_list:Array<User>,  user:any,  auth: Object  };  export const initialState: State = {  user\_list: [{  ‘id’: 0,  ‘username’: '',  ‘first\_name’: '',  ‘last\_name’: '',  ‘email’: '',  ‘password’: ''}],  user:{},  auth: {  username: 'Administrator'  }  }; |

**Inside the component:**

Below code in blue highlists usage of type definition in component level.

|  |
| --- |
| this.user$ = this.store.select(getUserLstState).subscribe(  (data: Array<User>) => {  this.gridOptions.rowData = data;  }); |