Connection to server:

In the present code base we have stored the server details like the base urls and endpoints in the file js/constant.js and we can see all the api calls in the lib/settings.js so in order to connect to the Sugarizer server and handel the data coming from the server we are reusing this files. And we are using Axios which is a popular HTTP client library to make HTTP request and interact with API's

Server information in js/constant.js is stored as follows:

```
var constant = {};
constant.staticInitActivitiesURL = "activities.json";
constant.http = "http://";
constant.https = "https://";
constant.dynamicInitActivitiesURL = "/api/v1/activities/";
constant.serverInformationURL = "/api/";
constant.loginURL = "/auth/login/";
constant.signupURL = "/auth/signup/";
constant.initNetworkURL = "/api/v1/users/";
constant.statsURL = "/api/v1/stats/";
constant.defaultServer = constant.https + "server.sugarizer.org";
constant.minPasswordSize = 4;
```

The Api calls we make throughout this project are as follows (reusing lib/server.js):

```
var server = {};
// Get server information
    server.getServerInformation = function(serverurl, response, error) {
        var serverUrl = serverurl;
        if (serverUrl.length && serverUrl[serverurl.length-1] == '/') {
            serverUrl = serverUrl.substr(0, serverUrl.length-1);
        }
        var ajax = axios.create({
            responseType: "json"
        });
        ajax.get(serverUrl +
constant.serverInformationURL).then(function(inResponse) {
            if (response) response(null, inResponse.data);
        }).catch(function(inResponse) {
            if (error) error(inResponse, getErrorCode(inResponse));
        });
    }
// Get activities
    server.getActivities = function(response, error) {
        var ajax = axios.create({
```

```
            responseType: "json",
            headers: computeHeader(preferences.getToken())
        });
        ajax.get(server.getServerUrl() +
constant.dynamicInitActivitiesURL).then(function(inResponse) {
            var newResponse = {data: inResponse.data};
            if (response) response(null, newResponse);
        }).catch(function(inResponse) {
            var code = getErrorCode(inResponse);
            if (code == 3) {
                if (sessionExpired()) {
                    return;
                }
            }
            if (error) error(inResponse, code);
        });
    }

    // Get user information
    server.getUser = function(userId, response, error, optserver) {
        var ajax = axios.create({
            responseType: "json",
            headers: computeHeader(preferences.getToken())
        });
        ajax.get((optserver ? optserver : server.getServerUrl()) +
constant.initNetworkURL + userId).then(function(inResponse) {
            response(null, inResponse.data);
        }).catch(function(inResponse) {
           ……..
        });
    }

    // Create a new user and for validating a username
    server.postUser = function(user, response, error) {
        var ajax = axios.create({
            responseType: "json"
        });
        ajax.post(server.getServerUrl() + constant.signupURL, {user:
JSON.stringify(user)}).then(function(inResponse) {
            if(!user.beforeSignup) {
```

```
                var newuser = {"name": user.name, "password":
user.password};
                server.loginUser(newuser, function(loginSender,
loginResponse) {
                    preferences.setToken({'x_key': loginResponse.user._id,
'access_token': loginResponse.token});
                });
            }
            if (response) response(null, inResponse.data);
        }).catch(function(inResponse) {
            ……….
        });
    }

    // Create a new user
    server.loginUser = function(user, response, error) {
        var userValue = user;
        userValue.role = ["student","teacher"];
        var ajax = axios.create({
            responseType: "json"
        });
        ajax.post(server.getServerUrl() + constant.loginURL, {user:
JSON.stringify(userValue)}).then(function(inResponse) {
            response(null, inResponse.data);
        }).catch(function(inResponse) {
            ……….
        });
    }

    // Update user
    server.putUser = function(userId, settings, response, error) {
        var ajax = axios.create({
            responseType: "json",
            headers: computeHeader(preferences.getToken())
        });
        ajax.put(server.getServerUrl() + constant.initNetworkURL + userId,
{user: JSON.stringify(settings)}).then(function(inResponse) {
            response(null, inResponse.data);
        }).catch(function(inResponse) {
            ……….
```

```
        });
    }

    // Delete user
    server.deleteUser = function(userId, response, error) {
        var ajax = axios.create({
            responseType: "json",
            headers: computeHeader(preferences.getToken())
        });
        ajax.delete(server.getServerUrl() + constant.initNetworkURL +
userId).then(function(inResponse) {
            response(null, inResponse.data);
        }).catch(function(inResponse) {
            ……….
        });
    }
```

Let us see what different api calls are made on each screen.

First Screen And Login Screen:

We are making a post request to the endpoint `"/auth/signup/"`

1) in order to validate the username in both sign case and login case and based on the status code we get we will display the changes.

2)Whenever a new user signin we are making this http request.

We are making a post request to the endpoint `"/auth/login/"`

1)Whenever a user tries to login we are making this request.

We are making a get request to the endpoint `”/auth/signup/:id”`

1)Whenever a user tries to login we are making this request

Home Screen:

In home screen inorder to display all favorite activities in spiral view we are getting the activities list from lib/activity.js and in that we are making a get request to the endpoint `"/api/v1/activities/"`

List View:

In listview in order to display all activities we are getting the activities list from lib/activity.js and in that we are making a get request to the endpoint `"/api/v1/activities/"`

Settings:

When we want to change the password using My security then first when type our present password we are sending an post request to `"/auth/login/"`

And after that when we change the password we are making a put request `"/api/v1/users/:uid"`
When we want to delete a user by using My privacy setting we are sending a delete request to the endpoint `"/api/v1/users/:uid".`