Asynchronous Programming 2

# Exercise 01: Get Data from API Using Callback

You are given a predefined project from the file **javascript-async-app.zip.**

The project is a simple mini app which display a list of users with his posts and comments. Investigate it carefully before reading the rest.

To get data from an API, we will use $.get from jQuery

|  |
| --- |
| $.get('https://jsonplaceholder.typicode.com/users', function(users) {  // if we are here, which means the data is retrieved from the API  console.log(users);  }); |

In the above example, $.get take 2 parameters:

1. The URL to the API
2. The callback function that will be called when the data is retrieved from the first parameter. It is very similar to fs.readFile of NodeJS.

**PUT YOUR CODE IN FILE main.js LINE 7 OF FUNCTION DemoController**

Your task is to do the following using **Callback style.**

1. Get 10 users from this API (<https://jsonplaceholder.typicode.com/users>)

Example output:

|  |
| --- |
| [  {  id: 1,  name: 'Leanne Graham',  username: 'Bret',  email: 'Sincere@april.biz',  address: {  street: 'Kulas Light',  suite: 'Apt. 556',  city: 'Gwenborough',  zipcode: '92998-3874',  geo: {  lat: '-37.3159',  lng: '81.1496'  }  },  phone: '1-770-736-8031 x56442',  website: 'hildegard.org',  company: {  name: 'Romaguera-Crona',  catchPhrase: 'Multi-layered client-server neural-net',  bs: 'harness real-time e-markets'  }  },  ……….  ]; |

1. For each user get his posts using this API where userId is the id of a user ([https://jsonplaceholder.typicode.com/posts?userId=<<userId](https://jsonplaceholder.typicode.com/posts?userId=%3c%3cuserId)>>)

For example: to get the posts of userId 2, you need to call $.get from this API (https://jsonplaceholder.typicode.com/posts?userId=2)

1. For each post get all its comments using this API (<https://jsonplaceholder.typicode.com/comments?postId=1>)
2. Combine all 3 data together to get a list of Users which have a list of posts and each post have a list of comments

Your response will something like this (other field are omitted for the sake of readability)

|  |
| --- |
| [  {  id: 1,  posts: [  {  userId: 1,  id: 10,  comments: {  commentId: 123,  postId: 123  name: ...  }  }  ]  ....  },  ...  ]; |

1. Once you the data from (4), call the following function

|  |
| --- |
| $scope.$apply(function() {  $scope.users = users;  }); |

Where the variable **users** stored a list of Users which have a list of posts and each post have a list of comments. If you are doing correctly, you should see the figure below:

|  |
| --- |
|  |

# Exercise 02: Promises

In this exercise you do the same tasks like Exercise 1, but you have to use Promise API instead of Callback Style.

To do so, you must first convert the $.get to a function that return a Promise.