Ajax in AngularJS

- $http Service
- $q Service

$http Service

The $http service is a core Angular service that facilitates communication with the remote HTTP servers via the browser's XMLHttpRequest object or via JSONP.

Simple GET request example :

```
$http.get('/someUrl', config).
 success(function (data, status, headers, config) {
    // this callback will be called asynchronously when the response is available
 }).
 error(function (data, status, headers, config) {
    // called asynchronously if an error occurs or server returns response with an error status.
 });
```

Simple POST request example (passing data) :

```
$http.post('/someUrl', { msg: 'hello word!' }).
 success(function (data, status, headers, config) {
    // this callback will be called asynchronously when the response is available
 }).
 error(function (data, status, headers, config) {
    // called asynchronously if an error occurs or server returns response with an error status.
 });
```

Methods

- get(url, [config]);
- head(url, [config]);
- post(url, data, [config]);
- put(url, data, [config]);
- delete(url, [config]);

[config] object important properties

- **url** – {string} – Absolute or relative URL of the resource that is being requested.
- **method** – {string} – HTTP method (e.g. 'GET', 'POST', etc)
- **params** – {Object.<string|Object>} – Map of strings or objects which will be turned to ?key1=value1&key2=value2 after the url. If the value is not a string, it will be JSONified.
- **data** – {string|Object} – Data to be sent as the request message data.
- **responseType** - {string} – can be json/text/document...

Returns:

- **data** – {string|Object} – The response body from the server.
- **status** – {number} – HTTP status code of the response.
- **headers** – {function([headerName])} – Header getter function.
- **config** – {Object} – The configuration object that was used to generate the request.
- **statusText** – {string} – HTTP status text of the response.

Example:

```
<div ng-app="mainApp" ng-controller="studentsController">
   <ul>
      <li ng-repeat="s in students">
         {{s.RollNo + ", " + s.Name + ', ' + s.Marks}}
      </li>
   </ul>
</div>
<script>
   mainApp = angular.module("mainApp", [])
   mainApp.controller("studentsController", function ($scope, $http) {
      $http.get("Students.txt")
         .success(function (data) {
            console.log(data)
            $scope.students = data;
         })
         .error(function (msg) {
            $scope.students = [];
            alert(msg)
         })
   });
```

Students.txt

```
[ {
      "Name" : "S1",
      "RollNo" : "1",
       "Marks" : "100"
   },
   {
      "Name" : "S2",
      "RollNo" : "2",
      "Marks" : "90"
   },
   {
      "Name" : "S3",
      "RollNo" : "3",
      "Marks" : "60"
   },
   {
       "Name" : "S4",
      "RollNo" : "4",
      "Marks" : "70"
```

</script>	}]

$q Service

A service that helps us to run functions asynchronously, and use their return values (or exceptions) when they are done processing.

var deferred = **$q.defer();**

The Deferred object:

The purpose of the deferred object is to expose the associated Promise instance as well as APIs that can be used for signaling the successful or unsuccessful completion, as well as the status of the task.

Properties

- **promise** – promise object associated with deferred.

Methods

- **resolve(value)** – resolves the derived promise with the value.
- **reject(reason)** – rejects the derived promise with the reason.
- **notify(value)** - provides updates on the status of the promise's execution. This may be called multiple times before the promise is either resolved or rejected.

Promise Object Methods:

- **then**(successCallback, errorCallback, notifyCallback)
- **catch**(errorCallback)
- **finally**(callback, notifyCallback) – Executes callback irrespective of success or failure and is used for cleaning resources.

Example:

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
  <script src="angular.js"></script>
</head>
<body ng-app="myApp" ng-controller="queueServiceDemoController">
  <input type="text" name="txtName" ng-model="personName" />
  <input type="button" name="btnSayHello" value="SayHello" ng-click="callSayHello()" />
```

```
{{greetings}}
<script>
  var myApp = angular.module("myApp", []);
  myApp.controller("queueServiceDemoController", function ($scope,$q) {
    function sayHelloAsync(name) {
      var deferred = $q.defer();
      setTimeout(function () {
        if (name.toLowerCase() == "")
          deferred.reject("Anonymous cannot be greeted")
        else
          deferred.resolve("Hello " + name);
      }, 1000);
      return deferred.promise;
    }

    $scope.callSayHello = function (name)
    {
      var promise = sayHelloAsync($scope.personName);
      promise.then(function (data) {
        $scope.greetings = data;
      },
      function (msg) {
        $scope.greetings = msg;
      })
    }
  })
</script>
</body>
</html>
```

Building Custom Service with $http and $q Services

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
  <script src="angular.js"></script>
</head>
<body>
  <div ng-app="mainApp" ng-controller="studentController">
    <ol>
      <li ng-repeat="stud in students">
        {{stud.Name + " - " + stud.Marks}}
      </li>
    </ol>
  </div>
  <script>
    mainApp = angular.module("mainApp", [])
    mainApp.factory('StudentsService', function ($q,$http) {
      var factory = {};
      factory.getAllStudents = function () {
        var deferred = $q.defer();
        $http.get('Students.txt')
        .success(function (data) { deferred.resolve(data) })
         .error(function (reason) { deferred.reject(reason) })
        return deferred.promise;
      }
      return factory;
    });

    mainApp.controller("studentController", function ($scope, StudentsService) {
      var promise = StudentsService.getAllStudents();
      promise.then(
        function (students) {
                $scope.students = students
        },
```

```
            function (reason) {
                        $scope.notification = reason;
            },
            function (update) {
                        $$scope.notification = update;
          });
      });
  </script>
</body>
</html>
```