

Working with \$http

Dependency Injection

```
app.controllers("myCTRL", function( $scope, $http ) {
```

```
});
```

\$http.get()

```
$http.get('/getSomethingFromServer').then( function(response) {
```

```
    //    success
```

```
    //    The data you get from the response is expected to be in JSON format.
```

```
    // response.data
```

```
    // response.status
```

```
    // response.statusText
```

```
    // response.headers
```

```
    // config
```

```
}, function(response) {
```

```
    //    error
```

```
    // response.data
```

```
    // response.status
```

```
    // response.statusText
```

```
    // response.headers
```

```
    // config
```

```
});
```

\$http.post()

```
var obj = { key1: val1, key2: val2 };
```

```
$http.post('/postSomethingToServer', obj ).then( successCB, errorCB );
```

Working with ngRoute

Step 1 – Add Library to your HTML

```
<script src="https://code.angularjs.org/1.x.x/angular-route.min.js"></script>
```

Step 2 – Add Dependency

```
var app = angular.module("myApp",["ngRoute"]);
```

Step 3 – Configure \$routeProvider

```
app.config(function( $routeProvider ){  
  
    $routeProvider  
    .when("/",{  
        templateUrl: "view/default.html",  
        controller: "listController"  
    })  
    .when("/edit/:id",{  
        templateUrl: "view/edit.html",  
        controller: "editController"  
    })  
    .when("/add",{  
        templateUrl: "view/add.html",  
        controller: "addController"  
    })  
  
});
```

Step 4 (Optional) – Working with \$routeParams

```
app.controller("myCTRL", function($scope, $routeParams ){  
  
    var id = $routeParams.id;  
  
});
```

MongoDB NodeJS Module

```
var mongodb = require("mongodb");

var db = mongodb.Db("mydb", new mongodb.Server('localhost', 27017) );

db.open( function(error) { } );

db.close();

db.collection("myCollection", function(error, collection) { } );

collection.insert( obj );
collection.insert(obj, function(err, insertedId) { } );

collection.findOne( { _id: new ObjectId('abcd...') }, function(error, obj) { } );

collection.find( { }, function( error, cursor ) { } );

cursor.toArray( function( error, arr ) { } );

collection.save(function(err, obj ){ var newObjID = obj._id; });

collection.deleteOne({_id: new mongodb.ObjectId('abcde.....')});
```

Working with ObjectId

```
var ObjectId = mongodb.ObjectId;

var _idToFind = new ObjectId('abcdef.... ');
```