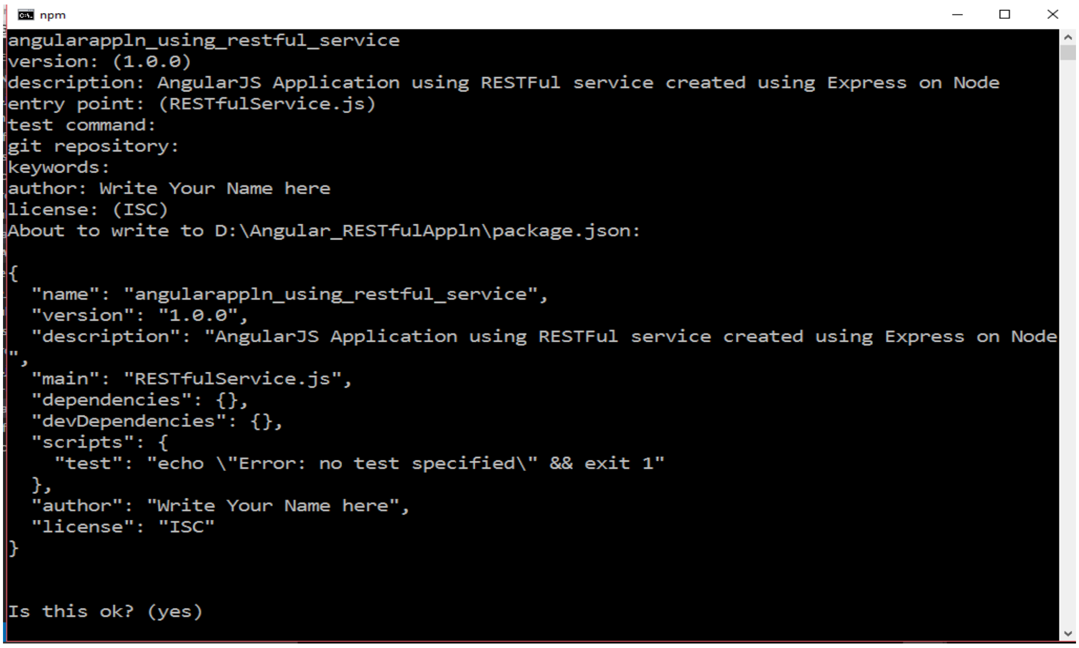
Creating RESTful Service using Express on Node and consuming it in an AngularJS Application

Phase I Create RESTful Service using Express on Node

For Service

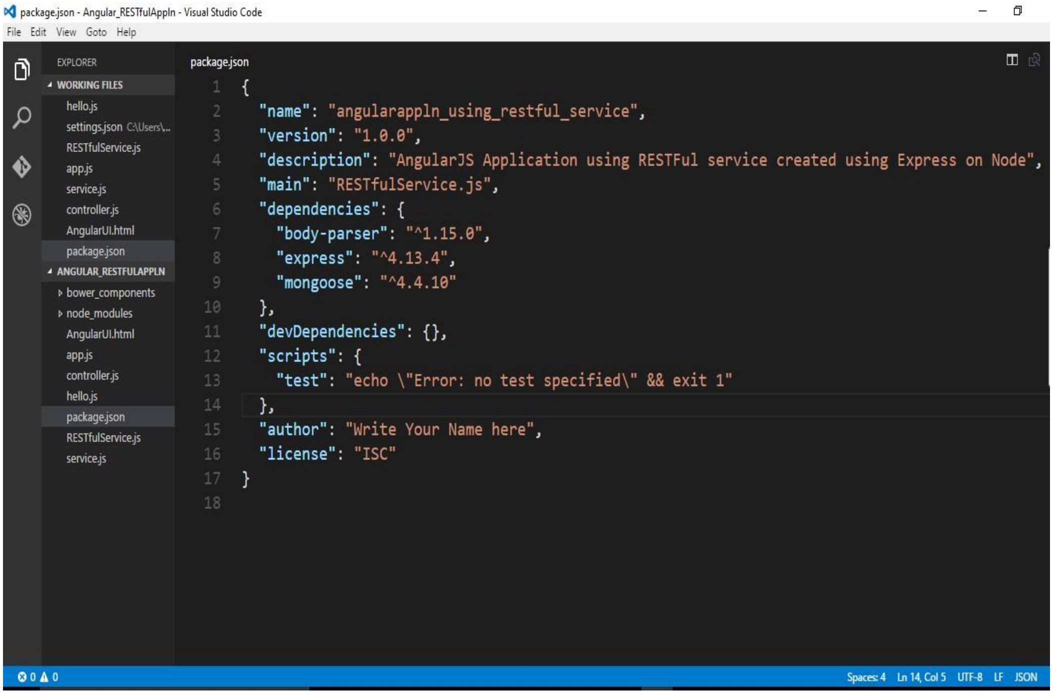
- Need to Install Node www.nodejs.org
    - Install LTS if in production phase – Long Time Support
    - Install Stable if in development phase – new features
- Need to install Mongodb www.mongodb.org
    - Install as per your operating system and 32 / 64 bit
    - Follow the appropriate steps to install and to launch mogod server which will be hosting mongodb database.
- Create One Folder for this application, I have given name as Angular_RESTFulAppln
- Use any IDE for javascript, here I am using Visual Studio Code, In VSCode open folder i.e. my application folder i.e. Angular_RESTFulAppln.
    - Create one .js file which will have RESTFul Service code.
    - I have created RESTFulService.js file and here I need to use some of the node modules like express, body-parser, and mongoose.
    - So definitely we need to install these node modules prior to use them in my application. Prior to install these modules i.e. dependencies we would create package.json file for application, so that if required dependencies won't be available at any particular location wherever application is deployed package.json would have information about the dependencies, which could be resolved by using "npm install" on node command prompt.
    - To create package.json for application we are using node command prompt, go to the application folder and use command "npm init", it would ask to specify appropriate information. E.g. like following
    - 

```
angularappln_using_restful_service
version: (1.0.0)
description: AngularJS Application using RESTFul service created using Express on Node
entry point: (RESTfulService.js)
test command:
git repository:
keywords:
author: Write Your Name here
license: (ISC)
About to write to D:\Angular_RESTfulAppln\package.json:

{
  "name": "angularappln_using_restful_service",
  "version": "1.0.0",
  "description": "AngularJS Application using RESTFul service created using Express on Node
",
  "main": "RESTfulService.js",
  "dependencies": {},
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Write Your Name here",
  "license": "ISC"
}


Is this ok? (yes)
```

    - This would generate package.json, so now when we install node modules like express, body-parser, mongoose (as per our application needs), package.json will have dependencies over there.
    - To install node modules locally use following command – Go to application folder

➔ “npm install express --save”
➔ “npm install body-parser --save”
➔ “npm install mongoose --save”

- –save actually adds dependencies within package.json file



```
{
  "name": "angularappln_using_restful_service",
  "version": "1.0.0",
  "description": "AngularJS Application using RESTFul service created using Express on Node",
  "main": "RESTfulService.js",
  "dependencies": {
    "body-parser": "^1.15.0",
    "express": "^4.13.4",
    "mongoose": "^4.4.10"
  },
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Write Your Name here",
  "license": "ISC"
}
```

- Now let’s have the actual RESTful Service code
- Code has comments which make it self-explanatory

```
//Express is required for creating Node.js based web apps
var express = require('express');

//body-parser is used to parse the Request body and populate the req.
var bodyParser = require('body-parser');
//mongoose is used for interacting with MongoDB

var mongoose = require('mongoose');

// Create Express app
var app = express();

// Setting port no for listening
app.set('port',3300);

app.use(bodyParser.json());

//Let’s specify mongodb’s database name and the fully qualified path of mongod // server
var dbHost = 'mongodb://localhost:27017/WebinarDB';
mongoose.connect(dbHost);
```

```
//Create a schema for Book
var bookSchema = mongoose.Schema({
name: String,
//Also creating index on field isbn
isbn: {type: String, index: true,unique:true},
author: String,
pages: Number
});

var Book = mongoose.model('Book',bookSchema);

// connecting to Mongod instance
mongoose.connection;

//Starting up the server on the port: 3300
app.listen(app.get('port'), function(){
console.log('Server up: http://localhost:' + app.get('port'));
});

// To allow CORS - Cross Origin Resrouce Sharing
app.all('*',function(req,res,next){
res.header('Access-Control-Allow-Origin', '*');
res.header('Access-Control-Allow-Methods', 'PUT, GET, POST, DELETE, OPTIONS');
res.header('Access-Control-Allow-Headers', 'Content-Type');
next();
});
```

- RESTFul Methods – Get All, Get, Add New, Update, Delete

```
// Get all the books
app.get('/books',function(req,res){
    Book.find({},function(err,result){
    if(err) throw err;
    res.json(result);
    });
});

//Get the details of the book with the given isbn
app.get('/book/:isbn', function(req, res){
  console.log("Fetching details for book with ISBN: " + req.params.isbn);

  Book.findOne({isbn: req.params.isbn}, function(err, result){
    if ( err ) throw err;
    res.json(result);
  });
});

//Add a new book
app.post("/book", function(req, res){
  console.log("Adding new Book: " + req.body.name);
});
```

```
        var book = new Book({
         name:req.body.name,
         isbn: req.body.isbn,
         author: req.body.author,
         pages: req.body.pages
        });
        //Saving the model instance to the DB
        book.save(function(err, result){
         if ( err ) throw err;
         res.json({
          message:"Successfully added the Book!",
          book:result
         });
        });
       });

       //Update an existing book app.put("/book/:isbn",
       function(req, res){ Book.findOne({isbn: req.params.isbn},
       function(err, result){
         if ( err ) throw err;

         if(!result){
          res.json({
           message:"Book with ISBN: " + req.params.isbn+" not found.",
          });
         }

         result.name   = req.body.name;
         result.isbn   = req.body.isbn;
         result.author = req.body.author;
         result.pages  = req.body.pages;

         result.save(function(err, result){
          if ( err ) throw err;
          res.json({
           message:"Successfully updated the book",
           book: result
          });
         });
    });
});

//Delete an existing book app.delete("/book/:isbn", function(req, res){
Book.findOneAndRemove({isbn: req.params.isbn}, function(err, result){
   res.json({
    message: "Successfully deleted the book",
    book: result
   });
 });
});
```

- RESTful Service is done here now launch mongod – server hosting mongodb database and also launch RESTful Service using **node Angular_RESTFulAppln**
- This would launch RESTFul Service on localhost:3300 – you can test it a browser using localhost:3300/book – gives you books list (if it is there in database)

Phase II – Create AngularJS Application as a UI for RESTful Service

- Now Let's create Angular Application for UI which would communicate to RESTFul Service
- Install AngularJS using bower package manager – Inside application folder over node command prompt type **"bower install angular"**
- In the same application folder create app.js having application module code like following

```
/// <reference path="./bower_components/angular/angular.min.js" />
var app = angular.module('crudModule',[]);
```

- Above code create application module named as 'crudModule' and specify if any dependencies are there.
- Let's create AngularJS Service which uses $http service to make ajax call to RESTful service

```
/// <reference path="./bower_components/angular/angular.min.js" />
/// <reference path="./app.js" />

app.service('crudService', function ($http) {
  //Create new record
  this.post = function (Book) {
    var request = $http({
      method: "post",
      url: "http://localhost:3300/book",
      data:  Book
    });
    return request;
  }

  //Get Single Record
  this.get = function (isbn) {
   return $http.get("http://localhost:3300/book/" + isbn);
  }

  //Get All Books
  this.getBooks = function () {
    return $http.get("http://localhost:3300/books");
  }

  //Update the Record
  this.put = function (isbn,Book) {
    var request = $http({
      method: "put",
      url: "http://localhost:3300/book/" + isbn,
      data: Book
    });
```

```
        return request;
    }

    //Delete the Record
    this.delete = function (isbn) {
        var request = $http({
            method: "delete",
            url: "http://localhost:3300/book/" + isbn
        });
        return request;
    }
});
```

- Now as angularjs service is done lets create controller which would user this service and push models data to the view and handles events related to the controls on the view.

```
/// <reference path="./bower_components/angular/angular.min.js" />
/// <reference path="./app.js" />

//The controller is having 'crudService' dependency.
//This controller makes call to methods from the service
app.controller('crudController', function ($scope, crudService) {

    $scope.IsNewRecord = 0; //flag value 1 to insert new record and 0 to update existing record

    loadRecords(); // Initially loads all records from the db to view (UI)

    //Function to load all Book records
    function loadRecords() {
        var promiseGet = crudService.getBooks(); //Method Call from the service

        promiseGet.then(function (pl) { $scope.Books = pl.data },
              function (errorPl) {
                  $log.error('failure loading Books', errorPl);
              });
    }

    //The Save scope method use to define the Book object.
    //In this method if IsNewRecord is zero then Update Book else
    //Create the Book information to the server

    $scope.save = function () {
        var Book = {
            name: $scope.name,
            isbn: $scope.isbn,
            author: $scope.author,
            pages: $scope.pages
        };
    //If the flag is 1 the it is new record



    if ($scope.IsNewRecord === 1) {
        var promisePost = crudService.post(Book);
        promisePost.then(function (pl) {
            $scope.isbn = pl.data.isbn;
            loadRecords();
            $scope.IsNewRecord = 0;
            $scope.Message = "Added Successfuly";
        }, function (err) {
            console.log("Err" + err);
        });
    } else { //Else Edit the record
        var promisePut = crudService.put($scope.isbn,Book);
        promisePut.then(function (pl) {
            $scope.Message = "Updated Successfuly";
            loadRecords();
        }, function (err) {
            console.log("Err" + err);
        });
    }
};

//Method to Delete
$scope.delete = function () {
    var promiseDelete = crudService.delete($scope.isbn);
    promiseDelete.then(function (pl) {
        $scope.Message = "Deleted Successfuly";
        $scope.name = "";
        $scope.isbn = "";
        $scope.author = "";
        $scope.pages = 0;
        loadRecords();
    }, function (err) {
        console.log("Err" + err);
    });
}

//Method to Get Single Book based on isbn no
$scope.getBook = function () {
    var promiseGetSingle = crudService.get($scope.isbn);

    promiseGetSingle.then(function (pl) {
        var res = pl.data;
        $scope.name = res.name;
        $scope.isbn = res.isbn;
        $scope.author = res.author;
        $scope.pages = res.pages;
```

```
            alert($scope.name+" "+$scope.isbn);
            $scope.IsNewRecord = 0;
        },
                function (errorPl) {
                    console.log('failure loading Book', errorPl);
                });
    }
    //Clear the Scope models
    $scope.clear = function () {
        $scope.IsNewRecord = 1;
        $scope.name = "";
        $scope.isbn = "";
        $scope.author = "";
        $scope.pages = 0;
    }
});
```

- Now let’s have view for the angular application so design AngularUI.html

```
<html ng-app="crudModule">
  <head>
    <title>CRUD Operations using AngularJS UI and RESTful service using Express on Node!</title>
    <style>
      table#tblContainer
      {
           width: 50%;
      }
    #tblContainer, #tblcollections, #tblCRUD, td
    {
      border:3px solid maroon;
      background-color: mintcream;
      color: black;
      margin-left: auto;
      margin-right: auto;
    }
    th {
      background-color:darkgray;
    }
     #dvcollection {
      height: 300px;
      overflow-y: scroll;
    }
    </style>

  </head>

  <body>

<h2 style="text-align:center;">CRUD Operations using AngularJS UI and RESTful
 service using Express on Node!</h2>
```

```
<table id="tblContainer" ng-controller="crudController">
  <tbody>
   <tr>
    <td>
      <table id="tblCRUD">
        <tbody>
         <tr>
          <td>
            <span>Name</span>
          </td>
          <td>
            <input type="text" id="name" required="true" ng-model="name"/>
          </td>
        </tr>
        <tr>
          <td>
            <span>ISBN</span>
          </td>
          <td>
            <input type="text" id="isbn" required="true" ng-model="isbn"/>
          </td>
        </tr>
        <tr>
          <td>
            <span>Author</span>
          </td>
          <td>
            <input type="text" id="author" required="" ng-model="author"/>
          </td>
        </tr>
        <tr>
          <td>
            <span>Pages</span>
          </td>
          <td>
            <input type="number" id="pages" required="" ng-model="pages"/>
          </td>
        </tr>
        <tr>
          <td>
          </td>
          <td>
            <input type="button" id="new" value="New" ng-click="clear()" />
            <input type="button" id="save" value="Save" ng-click="save()" />
            <input type="button" id="get" value="Get Book" ng-click="getBook()" />
            <input type="button" id="delete" value="Delete" ng-click="delete()" />
          </td>
        </tr>
```

```
            </tbody></table>
            <div>{{Message}}</div>
         </td>
         </tr>
         <tr>
         <td>
            <div id="dvcollection">
               <table id="tblcollections">
                  <thead>
                     <tr>
                        <th>Name</th>
                        <th>ISBN</th>
                        <th>Author</th>
                        <th>Pages</th>
                     </tr>
                  </thead>
                  <tbody ng-repeat="BOOK in Books">
                     <tr ng-click="getBook(BOOK)">
                        <td> <span>{{BOOK.name}}</span></td>
                        <td> <span>{{BOOK.isbn}}</span></td>
                        <td> <span>{{BOOK.author}}</span></td>
                        <td> <span>{{BOOK.pages}}</span></td>
                     </tr>
                  </tbody>
               </table>
            </div>
         </td>
      </tr>
   </tbody></table>

<script src="./bower_components/angular/angular.min.js"></script>
<script src="./app.js"></script>
<script src="./service.js"></script>
<script src="./controller.js"></script>
</body>
</html>
```

- To run this application first of all go to node command prompt set the path of RESTFul Service application path i.e. Angular_RESTFulAppln and run **node RESTfulservice** would host RESTFul Service at localhost:3300 address
- Open Application Folder in explorer, right click on AngularUI.html in browser like say in Chrome, so basically RESTful service hosted on localhost:3300 and Angular application we are hosting on file system and now angular UI making call to RESTful service hosted on localhost:3300 possible because of CORS being taken care.
- See the UI and Add new record, update existing record, delete available record, Get individual record details.

MongoDB data used for this demo

```
db.books.insert(
[
{
name: 'Let Us C',
isbn: 'isbn123',
author: 'Yashwant Kanetkar',
pages: 400
},
{
name: 'Let Us C++',
isbn: 'isbn556',
author: 'Yashwant Kanetkar',
pages: 200
},
{
name: 'How to angularJS',
isbn: 'isbn3434',
author: 'Abcd xyz',
pages: 500
},
{
name: 'How to noeJS',
isbn: 'isbn2342',
author: 'new guy',
pages: 540
},
])
```