# Assessment:

## Bug Tracker

### REST APIs:

[Run "node server/server.js"]

Project

GET - http://localhost:3000/api/projects

POST - http://localhost:3000/api/projects

PUT - http://localhost:3000/api/projects/:id

GET Bugs of a project - http://localhost:3000/api/projects/:id/bugs

DELETE - http://localhost:3000/api/projects/:id

Bug:

GET - http://localhost:3000/api/bugs

POST - http://localhost:3000/api/bugs

PUT - http://localhost:3000/api/bugs/:id

DELETE - http://localhost:3000/api/bugs/:id

Create an application to create, list and close bugs.

### Application Use Cases:

Route : #/projects/

List all the projects as links (/projects/:id/bugs)

Route : #/projects/**:id**/bugs

List all the bugs that belong to the project (**:id**) with a link (/bugs/:id) for each bug. Visually differentiate "open" and "closed" bugs.

Route : #/bugs/:id

Display the information about the bug.

If the bug status is "Open" the display a "Close" button using which the user can close the bug.

Route : #/bugs/new

Display a screen where the user can create a new bug for a particular project (all project names are displayed in a drop down list from which the user can choose one)

## Test Cases

### ProjectsListController

* When initialized, should call the query method of the ProjectsService
* The data returned by the ProjectsService has to be assigned to $scope

### BugsListController

* When initialized, should call the ProjectsService.get() with the value of the route parameter "id" and the result should be assigned to $scope.project
* When initialized, $scope.bugs should be populated with the bugs list of the given project
* $scope.closeBug() method should update the bug's closeDate with the current date and time.
* $scope.closeBug() should initiate the BugService.update() method

### NewBugController

* $scope.Save() method should initiate the call to the BugService.save() method
* $location.url() should be triggered appropriately.