

Lab 6.1: Restful Resources with Angular.js

IN705 Databases Three

Introduction

We've made a start with Angular.js, but we need to get the interaction with our REST API server going, meaning we need to start sending HTTP requests to our server. Since our API is reasonably RESTful, we can do this pretty easily using `ngResource`, a library for use with Angular.js.

1 Preliminaries: an issue with our API server

Our JavaScript client isn't going to play nicely with our REST server unless we make a little change to our HTTP headers coming from the server. We need to add a small amount of code to our controller classes.

At the top of your `UserController` class, add the line

```
before_filter :set_headers
```

This tells the controller to apply the `set_headers` method before performing other methods. The new method should be placed at the bottom of the class and looks like this

```
def set_headers
  headers['Access-Control-Allow-Origin'] = '*'
end
```

2 Adding to our Angular controller

First, we need to include `ngResource` into our app by modifying the module definition as follows

```
var app = angular.module('splatter-web', ['ngResource']);
```

Then we identify our RESTful resource as follows

```
app.factory("User", function($resource) {
  return $resource("http://clark.sqrawler.com:3000/users/:id");
});
```

Now our app has a `User` resource that can get data from and send data to our REST server.

Use the resource in your `UserController` like this

```
app.controller('UserController', function(User) { //notice that we pass User here
  this.u = User.get({id:1});
});
```

Now we get our user from the REST server.

3 Deployment

At this stage you should place your HTML and JavaScript files in the `/var/www/html` directory of your EC2 server so that they are accessible online.