

Welcome to the Angular JS meetup!

Feel free to grab a drink, some pizza, sit down and relax!

We also have a raffle at: <https://goo.gl/zVmF6U>

2 JetBrains software licenses to win for your favorite IDE or JetBrains product!

www.jetbrains.com/webstorm



How to interact with RESTful web services?

Alain Chautard
www.interstate21.com



What is a RESTful Web Service?

- REST (REpresentational State Transfer) is an architecture style based on HTTP (GET, POST, DELETE, PUT)
- Typical REST requests look like this:
 - GET <http://myserver.com/user/1> => Returns the user with ID = 1
 - POST <http://myserver.com/user/1> => Updates user with ID = 1
 - PUT <http://myserver.com/user/> => Creates a new user
 - DELETE <http://myserver.com/user/2> => Deletes user with ID = 2

Examples of RESTful web services can be found at <https://www.mashape.com/>

How to make HTTP requests with Angular?

- Use the \$http service: [https://docs.angularjs.org/api/ng/service/\\$http](https://docs.angularjs.org/api/ng/service/$http)
- Here is a simple example:
<http://codepen.io/achautard/pen/tEhLe?editors=101>
- The HTTP service supports all HTTP actions and automatically parses JSON objects for you

Example with HTTP POST

```
var dataObj = {  
    name : $scope.name,  
    employees : $scope.employees,  
    headoffice : $scope.headoffice  
};  
  
var res = $http.post('/savecompany_json', dataObj);  
  
res.success(function(data, status, headers, config) {  
    $scope.message = data;  
});  
  
res.error(function(data, status, headers, config) {  
    alert( "failure message: " + JSON.stringify({data: data}));  
});
```

What if I need custom headers or data format?

- Use \$httpClientProvider to configure everything:

```
$httpClientProvider.defaults.headers.get = { 'My-Header' : 'value' }  
$httpClientProvider.defaults.headers.common.Authorization = 'Basic YmVlcDpib29w'
```

- There are also transformations & interceptors available
 - Transformations allow you to format any request/response data
 - Interceptors allow you to handle errors or do any kind of pre-processing or post-processing (loader spinners, logging, error management, etc.)

```
$httpClientProvider.interceptors.push(function($q, dependency1, dependency2) {  
  return {  
    'request': function(config) { // do something with each request },  
    'response': function(response) { // do something else with each response }  
  };  
});
```

What about a higher level API?

- \$http is fun but... What if our server calls could look like this:

```
User.get( {userId:123}, function(data){  
    $scope.user = data;  
});
```

or even better:

```
var user = User.get({userId:123});
```

\$resource is that higher level API

- \$resource lets you define your RESTful calls:

```
var User = $resource('/user/:userId', {userId:'@id'});
```

```
var user = User.get({userId:123}, function() {
```

```
  user.abc = true;
```

```
  user.$save(); // Instances even have action methods
```

```
});
```


\$resource is just great

- \$resource defines default action aliases:

```
{ 'get': {method:'GET'},  
  'save': {method:'POST'},  
  'query': {method:'GET', isArray:true},  
  'remove': {method:'DELETE'},  
  'delete': {method:'DELETE'} };
```

- Examples:

```
var User = $resource('/user/:userId', {userId:'@id'});  
User.save(myUser); // performs a HTTP POST  
User.remove(myUser); // performs a HTTP DELETE
```

\$resource is just great

- You can also define your own actions - and customize everything to your needs:

```
app.factory('ActivityData', function($resource, Constants) {  
    return $resource(Constants.URL + "/data/:userEmail/:dataType/:startTime", { dataType: 'all'},  
        { get: {method:'GET', cache: true, isArray: true},  
          getTotals: {method:'GET', cache: true},  
          getStepsTotal: {method:'GET', cache: true, params: { startTime: 0, dataType: 'steps'}}  
        });  
});
```

- HTTP headers can be customized as well - just like with \$http - you could even have URL variations for each action

\$resource in action

- This is how the previous resource would be used in an app:

```
app.controller('teamController', function($scope, $rootScope, TeamData) {  
  
    TeamData.getAllTotals({startTime: $rootScope.startTime}, function(data){  
        $scope.teamsData = data;  
    });  
});
```

- \$resource documentation: [https://docs.angularjs.org/api/ngResource/service/\\$resource](https://docs.angularjs.org/api/ngResource/service/$resource)

The End

Enter the raffle at:

<https://goo.gl/zVmF6U>

2 JetBrains software licenses
to win for your favorite IDE!

www.jetbrains.com/webstorm

