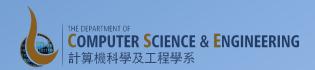
Client-Server Communication using \$resource

Jogesh K. Muppala





Angular ngResource

- The ngResource module provides a higher level abstraction than \$http for interacting with a RESTful API server
- Not part of Angular core

Installing and Using ngResource

Installing:
 bower install angular-resource –S

Angular \$resource Service

- Convenient wrapper around a REST API to perform CRUD operations
 - No need to deal with \$http directly, higher level abstraction
- Dependency injection:

```
.service('menuFactory', ['$resource', 'baseURL',
function($resource,baseURL) {
```

Usage:

```
$resource(url, [paramDefaults], [actions], options);
```

\$resource default actions

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

Examples:

```
$resource(baseURL+"dishes/:id",null,
{'update':{method:'PUT' }});
```

Using Resource methods:

```
.query(), .get(), .save(), .remove(), .delete()
```

• \$ prefix can be used with all non GET methods

Custom PUT request:

```
$resource(baseURL+"dishes/:id",null, {'update':{method:'PUT'}})
.update({id:$scope.dish.id},$scope.dish);
```

```
$resource(baseURL+"dishes/:id",null, {'update':{method:'PUT'}})
.query(
   function(response) {
        $scope.dishes = response;
        $scope.showMenu = true;
   function(response) {
        $scope.message = "Error: "+response.status + " " + response.statusText;
   });
```

```
$resource(baseURL+"dishes/:id",null, {'update':{method:'PUT' }})
.get({id:0}).$promise.then(
    function(response){
         $scope.dish = response;
         $scope.showDish = true;
    function(response) {
         $scope.message = "Error: "+response.status + " " + response.statusText;
```

Exercise: Using \$resource

- Install and use Angular ngResource module
- Use Angular \$resource to access the server that exports a REST API
- Handle errors caused during communication with a server using Angular \$resource
- Submit user's comments about a dish to the server by updating the information on the server