

Node, REST and Angular

av Robert Welin-Berger
robertwb@kth.se

Bakgrund till val av Angular

- Varför inte Angular2, ni är såååå 2016.
- Ni får köra Angular2 dock på eget bevåg.
- Typescript != Javascript, inlärningskurva
- Guider, standarder och market-share
- Erfarna assistenter

Angular

- MVW - Model, View, Whatever
- Full egen miljö
- Flexibel integration
- Single page application
- Väl integrerat med andra populära ramverk
- Two way data binding
- Dependency injection

Angular - Struktur

- Index.html
- <div ng-view></div>

```
app = angular.module("queue",[  
    'ngRoute',  
    'queueControllers',  
    'admin']);
```

Angular - Struktur

```
$routeProvider.  
when('/list', {  
    templateUrl: 'list.html',  
    controller: 'listController'  
})
```

Angular - Two way data binding

- `$scope.rooms = data.list;`
- `#Magic`
-
- `<div ng-repeat="room in rooms"`
`ng-click="redirect(room)">`
- `{{room.name}}`
- `<input type="text" ng-model="name">`

Angular - Two way vodoo

```
scope.$watch(  
  function() { return food; }, //watched value  
  function(newValue, oldValue) {  
    if ( newValue !== oldValue ) {  
      scope.foodCount = scope.foodCount + 1;  
    }  
  }  
)
```

Angular - Two way digest

- Digest
- `scope.$digest();`
- `$apply(/* thing to apply on */)`
- Don't dig too deep
- “somewhat complex”

Angular - Dependency Injection

- I want something, but it doesn't exist yet
- Singletons
- “Solves” mutual dependency nicely
- '\$scope', 'dep2', function(\$scope, dep2)

Angular - Service

- Kommer att dependency injiceras
- Använder sig av dependency injection
- Användbara för att spara global data
- Fungerar bra som wrappers
- Generell funktionalitet

Angular - Service

```
.factory('HttpService', function($http) {  
  return {  
    get: function(path, callback){  
      $http.get('/API/' + path).success(callback);  
    }  
  };  
});
```

Angular - Features and quirks!

- Filters!

ng-repeat="booking in bookedUsers | filter:
attending | filter: soon | orderBy : booking.time"

- Single page applications are cheap and fast!
- POJO - Plain Old Javascript Objects
- Magic! Good or bad?
- Can be really slow for many DOM objects

REST

- Representational state transfer
- Client-Server
- Stateless
- Layered system
- Uniform interface
- Cacheable

Rest - messages

- Idempotent - safety repeatable
- Safe - Won't change anything
- GET - Safe, Idempotent
- POST - no guarantees
- PUT - Idempotent
- DELETE Idempotent
- PATCH (optional)

Rest - examples

- /customers/12345
- /customers/12345/orders
- /customers/12345
- /customers/12345/orders/98765

NPM

- Node packet manager
- `npm install stuff --save`
- Added to `package.json`
- `^1.15.2` // x eller senaste
- Ägaren kan ladda upp vad som helst
- Kan tas bort när som helst
- Kan ersättas av vem som helst