

AngularJS - Routing

@kevinderudder

Agenda

- Need of Routing
- Layout Views
- #
- Browser History
- HTML5 Routing

Routing

- In a web application you probably have multiple pages or views
- Routing allows you to manage these multiple views
 - Pass parameters
 - Browser history
 - ...

Routing

- Based on the URL (duh)
 - Locate the resource / view / controller
 - “courses/angular”
- Rules
 - When our visitor comes to a url with a specific fragment, you need to act on that
 - “courses/angular/info”
- Happens on the client

Routing and Angular

- Add angular-route.js on your page
 - Registers a new ng-route module
- Tell angular how to respond to a specific url
 - \$routeProvider

```
$routeProvider.when(  
    "/info",  
    {  
        templateUrl: "courseinfo.html",  
        controller: "CourseController"  
    }  
);
```

Routing and Angular

When our app
gets bootstrapped

```
app.config(function ($routeProvider) {  
    // if the user comes to this part of the application  
    $routeProvider.when("/main", {  
        templateUrl: "main.html",  
        controller: "MainController"  
    })  
    .when("/user/:username", {  
        templateUrl: "user.html",  
        controller: "UserController"  
    })  
    .otherwise({ redirectTo: '/main' });  
});
```

Layout Views

HEADER

DIFFERENT TEMPLATES (ng-view)

FOOTER

ng-view

- This is the element where all the templates will be displayed

```
<div ng-view>  
  
</div>
```

- Put the views in a template folder
 - Best practice
 - Not called view, because this is more a final product

Added to the URL to specify that
this is extra information for the
page itself

\$routeParams

- When you need to pass a variable or value to a view use the colon
 - /trainers/:trainername
- By doing this, angular knows that this is a piece of info that will be provided via the url
 - Via the \$routeParams service

Default route

```
app.config(function ($routeProvider) {  
    // if the user comes to this part of the application  
    $routeProvider.when("/main", {  
        templateUrl: "main.html",  
        controller: "MainController"  
    })  
    .when("/user/:username", {  
        templateUrl: "user.html",  
        controller: "UserController"  
    })  
    .otherwise({ redirectTo: '/main' });  
});
```

Your default route

Browser History

- AngularJS is also taking care of your history
- Although you are staying on the same page, angular will add the views to the history

\$route Service

- You can also add your own properties to the routes

```
$routeProvider.when("/main", {  
    customProperty: 'my custom prop',  
    templateUrl: "main.html",  
    controller: "MainController"  
})
```

- Now you can use the \$route service in your controller to work with this property

```
console.log($route.current.customProperty)
```

QueryString

```
$route.current.params.yourproperty
```

HTML5 routing

- Goal is to get rid of the # sign in the url
 - Only for newer browsers

```
app.config(function ($routeProvider, $locationProvider) {  
    // if the user comes to this part of the application  
    $locationProvider.html5Mode(true);  
});
```

Template vs templateUrl

- Template is a direct output instead of a url

```
$routeProvider.when("/main", {  
    template: "Angular Course",  
    controller: "MainController"  
})
```

- To generate templates on the fly
 - Via a service