

Ejercicio 10 - Angular Routing

Este ejercicio hace referencia a los siguientes ejemplos:

- [12 - Angular Routing \(https://github.com/albertsgrc/curso-mean-jedi/tree/master/ejemplos/12_angular_routing\)](https://github.com/albertsgrc/curso-mean-jedi/tree/master/ejemplos/12_angular_routing)

Introducción

En este ejercicio os damos una página ya implementada, a la que **le falta sólo la parte de enrutamiento**, es decir, de mostrar los templates html donde toque cuando se pulsen los elementos de navegación de la página o cuando se cambie la url.

La página final tiene este aspecto:



Como véis, consta de un único elemento de navegación: Una barra con las pestañas Piet Mondrian, Ciclistas y About.

En resumen, lo que os pedimos para este ejercicio es que implementéis la navegación mediante clic en las pestañas y usando el módulo `ui.router`, con tal de que se muestre la vista correspondiente a la pestaña pulsada justo debajo de la barra de pestañas.

Enunciado detallado

Pestañas

Si abris en el navegador la página tal cual os la damos, váis a ver una página vacía. Eso es porque en la vista `index.html` el body esta vacío. Deberéis implementar un estado de `ui.router` llamado

layout, que esté asociado al **template** `partials/layout.html` y al **controlador** `LayoutCtrl`

y que

cargue el template en el elemento `<body>` de `index.html` (habrá que añadir el tag `ui-view` al `body`). Este estado no tendrá url asociada

y será abstracto. Un estado es abstracto cuando no se puede acceder a él directamente, a no ser que sea accediendo

a uno de sus estados hijos. Es como una clase abstracta de Java, no se puede instanciar la clase abstracta, pero sí

las clases hijas que no sean abstractas. Para declarar un estado abstracto, podéis indicar la opción `abstract: true` al declarar el estado:

```
$stateProvider.state('layout', {  
  abstract: true,  
  controller: 'LayoutCtrl'  
  ...  
})
```

Recordad que para indicar el estado padre de un estado podéis usar la opción `parent`:

`<nombre_estado_padre>`

```
$stateProvider.state('pietMondrian', {  
  parent: 'layout'  
  ...  
})
```

Seguimos: Si os fijáis, el template `partials/layout.html` es el que contiene la barra de las pestañas

y la barra superior con el título, pero no tiene el contenido de cada una de las pestañas, ya que este contenido

estará separado en varios templates html situados en la carpeta `partials`.

Deberéis crear por lo tanto los siguientes estados para cada pestaña, todos ellos hijos del estado `layout` (notad que tiene

sentido ya que se colocarán en el template asociado al estado `layout`, por lo tanto serán *hijos* suyos):

- Estado **pietMondrian**: Asociado a la pestaña con título **PIET MONDRIAN**, al controlador **PietMondrianCtrl**, al template `partials/pietMondrian.html` y a la url `/piet-mondrian`.
- Estado **ciclistas**: Asociado a la pestaña con título **CICLISTAS**, al controlador **CiclistasCtrl**, al template `partials/ciclistas.html` y a la url `/ciclistas`.
- Estado **about**: Asociado a la pestaña con título **ABOUT**, al controlador **AboutCtrl**, al template `partials/about.html` y a la url `/about`.

Los templates de estos estados se cargarán concretamente en el elemento `<div>` (el cual se indica con un comentario) del template `partials/layout.html`.

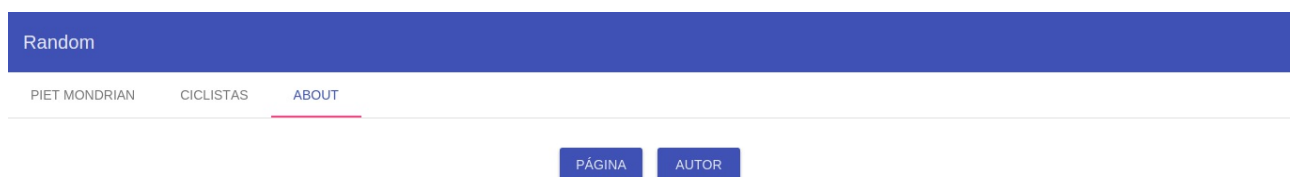
Para implementar que cuando se pulse la pestaña se muestre un estado concreto, **deberéis usar la directiva `ui-sref`** explicada en el ejemplo.

IMPORTANTE: Notad que como las pestañas tienen una url asociada, cuando un usuario entre directamente a la url correspondiente al estado (p.ej `http://localhost:8080/ciclistas`), además de mostrar el contenido del estado asociado a la url (algo que ya hace `ui-router` por defecto), deberá mostrarse como seleccionada la pestaña que toque, ya que sino siempre se mostrará seleccionada la primera independientemente de la url que pongamos. Para indicar a la página

que pestaña está seleccionada, deberéis assignar la variable `$scope.selectedIndex` del controlador `LayoutCtrl` al índice de la pestaña que toque (0, 1 o 2 dependiendo de si es `pietMondrian`, `ciclistas` o `about` respectivamente). Para hacerlo, deberéis poder discriminar en qué estado os encontráis inicialmente cuando se carga la página. Para ello podréis usar el método `includes(<nombre_estado>)` del servicio `$state` (`$state.includes(<nombre_estado>)`), que devuelve `true` si actualmente nos encontramos en el estado con nombre `<nombre_estado>`, o alguno de sus estados hijos. Notad que no hace falta actualizar la pestaña seleccionada (i.e actualizar la variable `$scope.selectedIndex`) cuando el usuario las va pulsando, ya que eso ya nos lo hace por defecto el módulo `angular-material` que hemos usado para construir la página. Sólo hay que actualizarla al principio, cuando se carga la página, i.e cuando se ejecuta el código de nuestro controlador `LayoutCtrl`.

Página About

Como podréis ver cuando hayáis implementado la página hasta este punto, la página `about` por defecto no muestra nada más que dos botones, **Página** y **Autor**:



El botón **Página**, al ser pulsado debería mostrar el contenido del template `partials/about/pagina.html`. El botón **Autor**, en cambio, debería mostrar el contenido del template `partials/about/autor.html`. Para implementar este funcionamiento usaremos también `ui.router`.

Así pues, habrá que declarar los siguientes estados, todos hijos del estado **about** (que **NO** será abstracto, ya que queremos que se pueda mostrar la página `about` sin ningún texto a la espera de que el usuario seleccione alguno de los botones):

- El estado **aboutPagina**, asociado al botón **Página** y al template **partials/about/pagina.html**.
- El estado **aboutAutor**, asociado al botón **Autor** y al template **partials/about/autor.html**.

Ninguno de los dos estados tendrá url asociada, ni controlador asociado. Los dos se cargarán en el elemento `` (el cual se indica con un comentario) del template **partials/about.html**. En este caso, para implementar que cuando se pulse el botón se vaya al estado correspondiente, **deberéis usar el servicio `$state` desde el controlador `AboutCtrl`**, llamando a su función `go(<nombre_estado>)`, no podréis usar la directiva `ui-sref`.

URL por defecto

La url por defecto de la página será `/piet-mondrian`.

Otros

IMPORTANTE: Toda la parte de declaración de estados de ui-router la tenéis que hacer obligatoriamente en el fichero `routing.js`.

Página final

Esto es lo que se debería ver en cada una de las pestañas/botones:

Piet Mondrian



Ciclistas

Random

PIET MONDRIAN

CICLISTAS

ABOUT



Fabian Cancellara
Contrarelojista/Clasicómano



Lance Armstrong
Escalador



Chris Froome
Escalador



Alberto Contador
Escalador

About sin haber pulsado ningún botón

Random

PIET MONDRIAN

CICLISTAS

ABOUT

PÁGINA

AUTOR

About habiendo pulsado el botón Página

Random

PIET MONDRIAN

CICLISTAS

ABOUT

PÁGINA

AUTOR

Esta página es muy random

About habiendo pulsado el botón Autor

Random

PIET MONDRIAN

CICLISTAS

ABOUT

PÁGINA

AUTOR

Albert Segarra Roca