EBook Gratis

APRENDIZAJE angular-ui-router

Free unaffiliated eBook created from
Stack Overflow contributors.

#angular-ui-router

Tabla de contenido

Acerca de 1

Capítulo 1: Empezando con angular-ui-router 2

Observaciones 2

Versiones 2

Examples 2

Hola mundo ejemplo 3

Vista basica 5

Definiendo un estado con vista múltiple 5

Resolviendo datos a un estado 6

Usando eventos de transición 6

Capítulo 2: Tipos de parametros personalizados 9

Parámetros 9

Examples 9

Parámetro de número de página 9

Parámetro booleano 10

Parámetro de ruta (con una barra no codificada en el interior) 10

Capítulo 3: Transición de estado 11

Examples 11

Recargar estado actual 11

Use $ state.go para la transición entre estados 11

Use $ state.transitionTo para la transición entre estados 12

Creditos 13

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: angular-ui-router

It is an unofficial and free angular-ui-router ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official angular-ui-router.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con angular-ui-router

Observaciones

Angular UI-Router es un marco de enrutamiento de aplicaciones de una sola página del lado del cliente para AngularJS .

Los marcos de enrutamiento para SPA actualizan la URL del navegador a medida que el usuario navega a través de la aplicación. A la inversa, esto permite cambios en la URL del navegador para conducir la navegación a través de la aplicación, lo que permite al usuario crear un marcador en una ubicación dentro del SPA.

Las aplicaciones de UI-Router se modelan como un árbol jerárquico de estados. UI-Router proporciona una máquina de estados para administrar las transiciones entre los estados de las aplicaciones de manera similar a una transacción.

Tomado de la página de UI-Router Github

Versiones

Versión	Fecha de lanzamiento
0.2.18	2016-02-07
0.2.17	2016-01-25
0.2.16	2016-01-24
0.2.15	2016-05-19
0.2.14	2016-04-23
0.2.13	2016-11-20
0.2.12	2016-11-13
0.2.11	2016-08-26
0.2.10	2016-03-12
0.2.9	2014-01-17
0.2.8	2014-01-16

Examples

Hola mundo ejemplo

PASO 1: Instalación

Antes de poder utilizar el enrutador Angular-UI, debe incluir AngularJS en su proyecto. Para una guía detallada al respecto, vea esta documentación .

Puede descargar Angular-UI Router desde su página GitHub o desde NuGet, NPM, Bower respectivamente.

Una vez que haya incluido el archivo JS en su página web, puede `ui.router` módulo `ui.router` dentro de su aplicación. En tu archivo de script deberías tener algo como esto:

```
var app = angular.module('app', []);
```

y ahora vamos a inyectar Angular-UI Router en nuestra propia aplicación como esta:

```
var app = angular.module('app', ['ui.router']);
```

Ahora Angular-UI Router se cargará con nuestra aplicación. Los siguientes pasos explicarán los conceptos básicos de Angular-UI Router y mostrarán algunas de las funciones básicas.

PASO 2: Definir estados simples

Puede configurar el UI-Router dentro de la función de `config` angular. Use el `$stateProvider` para definir sus estados. En el siguiente ejemplo, cada estado tiene una URL, un controlador y una plantilla.

```
(function() {
  var app = angular.module('app', ['ui.router']);

  app.config(['$stateProvider', function($stateProvider) {
      $stateProvider
        .state('home', {
          url: "/home",
          templateUrl: "home.html",
          controller: "homeCtrl"
        })
        .state('kitchen', {
          url: "/kitchen",
          templateUrl: "kitchen.html",
          controller: "kitchenCtrl"
        })
        .state('den', {
          url: "/denCtrl",
          templateUrl: "den.html",
          controller: "denCtrl"
        })
        .state('itemDetail', {
          url: "/items/:itemName",
          templateUrl: "item.html",
          controller: "itemDetailCtrl"
```

```
        })

    }])
})();
```

en su HTML, necesitará la directiva `ui-view` para que las vistas de estado se puedan rellenar en su interior.

```
<div ui-view></div>
```

PASO 3: Acceso a los estados

Existen en conjunto 3 formas de acceder a un estado que se define en `$stateProvider` .

1. Vía la directiva `ui-sref`

Puede acceder a los estados dentro de su HTML, usando la directiva `ui-sref`

```
<li ui-sref-active="active">
    <a ui-sref="kitchen">Go to the Kitchen</a>
</li>
<li ui-sref-active="active">
    <a ui-sref="den">Enter the den</a>
</li>
<li ui-sref-active="active">
    <a ui-sref="itemDetail({itemName:'keyboard'})">Key Board</a>
</li>
```

2. Vía `$state` servicio `$state` en el controlador.

También puede navegar a otros estados dentro de su controlador utilizando el `$state` proporcionado al controlador con el método `.go` .

```
.controller(function($scope, $state) {
    // ...
    $scope.navigateTo = function(stateName) {
        $state.go(stateName); // i.e. $state.go('den');
    };
})
```

3. A través de la url en el navegador.

Asumiendo que tienes un estado llamado `kitchen` definido de esta manera:

```
$stateProvider
  .state("kitchen", {
    url: "/kitchenUrl",
    ...
  });
```

Luego, al acceder a appdomain / kitchenUrl, la URL de su navegador irá a su estado de `kitchen` , asumiendo que no hay estados anidados y `appdomain` es el servidor que aloja su aplicación.

Si todavía estás confundido, aquí hay un Plnkr completamente funcional .

Vista basica

index.html

```
<html>
    <head>
        <title>Angular-UI Router Example</title>
        <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.9/angular.js"></script>
        <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/angular-ui-
router/0.3.1/angular-ui-router.js"></script>
        <script type="text/javascript" src="../js/script.js"></script>
    </head>
    <body ui-view="mainView"> <!-- Defining a container for our view -->
    </body>
</html>
```

script.js

```
var app = angular.module('app', ['ui.router']);
app.config(['$stateProvider', function($stateProvider){
    $stateProvider.state('home', {                          // Creating a state called 'home'
        url: '',                                            // An empty URL means that this
state will be loaded on the main URL when no other state is called
        views: {
            'mainView': {                                   // Section for our view-container
that we defined in the HTML
                template: '<h1>It works!</h1>'              // Setting a template for this view
                /*templateUrl: '../templates/home.html'*/ //templateUrl would load the file
and uses it's content as the template
            }
        }
    });
}])
```

Definiendo un estado con vista múltiple

En ui-router, un estado puede tener múltiples vistas, cada una con su propio controlador y una plantilla.

```
.state('dashboard', {
     name: 'dashboard',
     url: '/dashboard',
     views: {
         "view1": {
             templateUrl: "path/to/view1.html",
             controller: "view1Controller"
         },
         "view2": {
             templateUrl: "path/to/view2.html",
             controller: "view2Controller"
         }
     }
 })
```

Luego, dentro del HTML de su estado, puede vincular estas vistas

```
<div ui-view="view1"></div>
<div ui-view="view2"></div>
```

Resolviendo datos a un estado

Puede `resolve` datos en su estado cuando hace la transición a ellos, por lo general es útil cuando el estado necesita usar esos datos, o para resolver en un estado en el que es necesario autenticar alguna entrada proporcionada.

Cuando defina sus estados, deberá proporcionar un mapa de valores a resolver en la propiedad `.resolve` , cada valor resuelto debe tener una función que devuelva una `promise`

```
.state('main', {
     url: "/main",
     templateUrl: "path/to/main.html",
     controller: 'mainCtrl',
     resolve: {
         serverData: function ($http) {
             return $http.get('some/url');
         }
     }
});
```

Ahora, dentro de `mainCtrl` puede acceder a los datos (es decir, si la llamada `$http` resolvió correctamente).

```
.controller("mainCtrl", function($scope, serverData) {
    $scope.resolvedData = serverData.then(resp=> resp.data);
    ....
})
```

Usando eventos de transición

UI-Router expone eventos de transición que pueden ser útiles para manejar errores de transición, manejar / bloquear transiciones basadas en ciertos valores de parámetros, autenticación personalizada, etc.

Estos eventos pueden vincularse a `$rootScope` para un efecto global o a `$scope` para un efecto por controlador.

`$stateChangeError` : este evento se difunde cuando un intento de cambiar el estado ha fallado y se produjo un error, este evento activa una función de devolución de llamada con la siguiente firma:

> devolución de llamada (evento, toState, toParams, fromState, fromParams, error)

evento : el objeto evento

toState : el estado objetivo

toParams : los parámetros pasados al estado objetivo

fromState : estado actual

FromParams : los parámetros pasados al estado actual

error : el objeto de error

`$stateChangeStart` : este evento se transmite cuando se inicia una transición de estado, este evento activa una función de devolución de llamada con la siguiente firma:

devolución de llamada (evento, toState, toParams, fromState, fromParams, opciones)

opciones : el objeto de opciones de estado

`$stateChangeSuccess` : este evento se transmite cuando se completa una transición de estado, este evento activa una función de devolución de llamada con la siguiente firma:

devolución de llamada (evento, toState, toParams, fromState, fromParams, opciones)

`$stateNotFound` : este evento se transmite cuando no se encontró un estado al que solicitó la transición, este evento activa una función de devolución de llamada con la siguiente firma:

devolución de llamada (evento, unfoundState, fromParams, fromState)

unfoundState : un objeto que representa el estado que no se encontró

Ejemplo:

```
$rootScope.$on('$stateChangeSuccess', function (event, toState, toParams, fromState,
fromParams, options) {
    $log.debug("$stateChangeSuccess: event: %o toState: %o, toParams: %o, fromState: %o,
fromParams: %o, options: %o", event, toState, toParams, fromState, fromParams, options);
    // runs when the state has successfully changed
});

$rootScope.$on('$stateChangeStart', function (event, toState, toParams, fromState, fromParams,
options) {
    $log.debug("$stateChangeStart: event: %o toState: %o, toParams: %o, fromState: %o,
fromParams: %o, options: %o", event, toState, toParams, fromState, fromParams, options);
    // runs when the state has just started to transition
});

$rootScope.$on('$stateNotFound', function (event, unfoundState, fromParams, fromState) {
    $log.debug("$stateNotFound: event: %o unfoundState: %o, fromParams: %o, fromState: %o",
event, unfoundState, fromParams, fromState);
    // runs when the state wsa not found
});

$rootScope.$on('$stateChangeError', function (event, toState, toParams, fromState, fromParams,
error) {
    $log.debug("$stateChangeError: event: %o toState: %o, toParams: %o, fromState: %o,
```

```
fromParams: %o, error: %o", event, toState, toParams, fromState, fromParams, error);
    // runs when there was an error while attempting to transition
});
```

Lea Empezando con angular-ui-router en línea: https://riptutorial.com/es/angular-ui-router/topic/1869/empezando-con-angular-ui-router

Capítulo 2: Tipos de parametros personalizados

Parámetros

Parámetro	Detalles
`decode`	Convierte el valor de la URL (cadena) al valor disponible en `$stateParams`
`encode`	Convierte un valor a la cadena que se utilizará en la URL
`equals`	Verifica si dos valores son iguales desde el punto de vista del tipo.
`is`	Comprueba si el valor se puede utilizar como tipo de parámetro definido
`pattern`	Asegura que los valores de la URL coincidan con este patrón cuando se resuelve la ruta

Examples

Parámetro de número de página

Similar a `int` pero acepta solo enteros positivos (útil para paginación cuando hay un parámetro de `page` .

Definir:

```
module.config(['$urlMatcherFactoryProvider', function($urlMatcherFactory) {
  $urlMatcherFactory.type('page', {
    decode: function(val) { return +val; },
    encode: function(val) { return Math.floor(val); },
    equals: function(a, b) { return this.is(a) && +a == +b; },
    is: function(val) { return angular.isNumber(val) && val >= 1; },
    pattern: /\d+/
  })
}]);
```

Y use:

```
$stateProvider.state({
  url: '/my-route/{page:page}'
  template: '<my-page></my-page>'
});
```

Plunker y la respuesta de SO relacionada .

Parámetro booleano

Definir:

```
module.config(['$urlMatcherFactoryProvider', function($urlMatcherFactory) {
  $urlMatcherFactory.type('boolean', {
    decode: function(val) { return val == true || val == "true" },
    encode: function(val) { return val ? 1 : 0; },
    equals: function(a, b) { return this.is(a) && a == b; },
    is: function(val) { return [true, false, 0, 1].indexOf(val) >= 0 },
    pattern: /false|true|0|1/
  })
}]);
```

Y use:

```
$stateProvider.state({
  url: '/my-route/{showSidebar:boolean}'
  template: '<my-page></my-page>'
});
```

Plunker y la respuesta de SO relacionada .

Parámetro de ruta (con una barra no codificada en el interior)

Por defecto, ui-router codifica los parámetros de barra `/` interior. Si desea enviar una ruta en la URL, debe definir un tipo de parámetro personalizado.

Definir:

```
module.config(['$urlMatcherFactoryProvider', function($urlMatcherFactory) {
  $urlMatcherFactory.type('path', {
    decode: function(val) { return val != null ? val.toString() : val; },
    encode: function(val) { return val != null ? val.toString() : val; },
    is: function(val) { return this.pattern.test(val); },
    pattern: /[^/]+\/[^/]+/
  })
}]);
```

Y use:

```
$stateProvider.state({
  url: '/my-route/{directory:path}'
  template: '<my-page></my-page>'
});
```

Pregunta relacionada .

Lea Tipos de parametros personalizados en línea: https://riptutorial.com/es/angular-ui-router/topic/2031/tipos-de-parametros-personalizados

Capítulo 3: Transición de estado

Examples

Recargar estado actual

Puede recargar el estado actual usando el método `$state.reload` de su controlador

```
$state.reload()
```

Esta es una abreviatura de (código tomado de los documentos oficiales)

```
$state.transitionTo($state.current, $stateParams, {
  reload: true, inherit: false, notify: false
});
```

Ejecutar una recarga en su estado también reiniciará su controlador / s y volverá a resolver todos sus valores resueltos.

Use $ state.go para la transición entre estados

`$state.go` es un método abreviado para `$state.transitionTo`

> $ state.go (toState [, toParams] [, opciones])

Este método establece automáticamente sus opciones en `{ location: true, inherit: true, relative: $state.$current, notify: true }` (a menos que las anule) y le permite realizar la transición con menos código.

Ejemplos:

Digamos que tenemos una aplicación con un estado "principal", con 2 estados secundarios: "tablero" y "ayuda", y "tablero" también tiene un hijo llamado "acerca de".

Transición a otro estado

```
$state.go("main.dashboard") // from anywhere to 'main.dashboard'
```

Transición al estado padre

```
$state.go("^") // from 'main.dashboard' to 'main'
```

También puede pasar a otro hijo del estado principal (hermano)

```
$state.go("^.help") // from 'main.dashboard' to main.help
```

Colocando un . te permitirá la transición a estados infantiles

```
$state.go(".about") // from 'main.dashboard' to 'main.dashboard.about'
```

Use $ state.transitionTo para la transición entre estados

Use `$state.transitionTo` para pasar de un estado a otro. Este es un método de bajo nivel para la transición y `$state.go` es la forma recomendada para los casos de uso más comunes, ya que utiliza este método internamente.

    $ state.transitionTo (toState [, toParams] [, opciones])

toState - el estado para la transición a

toParams (opcional): un mapa de parámetros para enviar el estado objetivo

opciones (opcional) - las opciones de transición de estado

Ejemplos:

```
$state.transitionTo("dashboard.history", {period: "week"})
// transitions to the history child state with a state parameter
```

Lea Transición de estado en línea: https://riptutorial.com/es/angular-ui-router/topic/3608/transicion-de-estado

Creditos

S. No	Capítulos	Contributors
1	Empezando con angular-ui-router	Ajeet Lakhani, Community, CozyAzure, Gaara, svarog, ThermalCube
2	Tipos de parametros personalizados	fracz, lujcon
3	Transición de estado	Matt Tester, svarog