

Angular UI Router in depth

Table of Contents

Introduction	0
Configuring State	1
Activating State	2
Named Views	3
Nested State	4
onEnter onExit	5
Passing Data	6
Resolve Property	7
State Params	8
State Service	9
State Events	10
Abstract State	11

Angular UI Router

State - some place in ui (may not have template url)

Setup

```
<script src="http://angular-ui.github.io/ui-router/release/angular-ui-router.js"></script>
```



```
angular.module("myApp", ['ui.router']);

angular.module("myApp")
  .config(function($stateProvider, $urlRouterProvider){
    $stateProvider.state('home', {
      url: "/home",
      controller: "HomeCtrl", // or "HomeCtrl as h"
      controllerAs: "h",
      templateUrl: "views/home.html"
    });
  });
```

ui-view - where to place your template

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

Configuration

```
angular.module("myApp")
  .config(function($stateProvider, $urlRouterProvider){
    $stateProvider.state(stateName, stateConfig);
  });
```

- stateName - string
- stateConfig - object

State name

```
$stateProvider.state('home', {});
$stateProvider.state('home.child', {});
// state() returns $stateProvider
$stateProvider.state('home', {})
  .state("about", {})
  .state("contact", {})
```

stateConfig - object

```
template
templateUrl
templateProvider

controller
controllerProvider

resolve
url
params
views
abstract

onEnter ( state )
onExit ( state )

reloadOnSearch // on query parameter change state woun't reload
data
```

\$urlRouterProvider

```
$urlRouterProvider.when(fromPath, toPath);
$urlRouterProvider.otherwise(path);
$urlRouterProvider.rule(handler)
// handler should return dynamically constructed path string
```

Activating State

3 ways

```
$state.go("classroom");  
<a ui-sref="classroom">Class Room</a>  
<a href="/classroom">Class Room</a>
```

\$state

```
$state.go(to, [toParams], [options]);  
  
to - required string (state name, use ^, . for relative paths )  
toParams - optional parameter object  
options (optional object )  
  - location ('replace/true/false' default true)  
  - inherit // true (default)  
  - relative // $state.current  
  - notify // true (if false stateChangeStart & stateChangeSuccess events are not fired)  
  - reload // false ( if true force reload of state even parameters are the same )
```

Demo

```
$state.go("photos.detail")  
  
// now current state is "photos.detail"  
$state.go("^") // goes to state "photos"  
$state.go("^.list") // goes to sibling state ("photos.list")  
$state.go("^.detail.comment") // goes to "photos.detail.comment"
```

ui-sref (same as \$state.go but used in html)

```
ui-sref="stateName"  
ui-sref="stateName({param1: val1, param2: val2})"
```

Un-named view

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

```
$stateProvider.state('home', {  
  template: "<h1>Home page<h1>"  
});  
  
// OR ( not useful if have single view)  
  
$stateProvider.state('home', {  
  views: {  
    "": {  
      template: "<h1>Home page<h1>"  
    }  
  }  
});
```

Multiple Named Views

```
<div ui-view></div>  
<div ui-view="tab"></div>  
<div ui-view="accordian"></div>
```



```
$stateProvider.state('home', {  
  views: {  
    "": {  
      template: "<h1>Home page<h1>"  
    },  
    "tab": {  
      template: "<h1>Tab<h1>"  
    },  
    "accordion": {  
      template: "<h1>Accordion<h1>"  
    }  
  }  
});
```

Nested state and views

```
// Parent
.state('activities', {
  url: '/activities',
  controller: 'AllActivitiesController',
  controllerAs: 'activities',
  templateUrl: '/app/templates/allActivities.html',
  resolve: {
    user: function(){
      return { name: "Hemant"}
    }
  },
  myData: "Hello Fool"
})

// Child
.state('activities.details', {
  url: '/details',
  templateUrl: '/app/templates/allActivities.html',
  // resolve & myData from parent inherited
})

// OR
.state('details', {
  url: '/details',
  parent: "activities",
  templateUrl: '/app/templates/allActivities.html',
  // resolve & myData from parent inherited
})

// OR
.state('details', {
  url: '/details',
  parent: activities, // activities not string here as previous
  templateUrl: '/app/templates/allActivities.html',
  // resolve & myData from parent inherited
})
```

onEnter & onExit state

Usefull if want to show alert before leaving the page

```
.state('activities', {  
  url: '/activities',  
  controller: 'AllActivitiesController',  
  controllerAs: 'activities',  
  templateUrl: '/app/templates/allActivities.html',  
  onEnter: function(){  
    console.log("Entering into state");  
  },  
  onExit: function(){  
    console.log("Leaving the state");  
  }  
})
```

Assigning data to state

```
.state('activities', {  
  url: '/activities',  
  controller: 'AllActivitiesController',  
  controllerAs: 'activities',  
  templateUrl: '/app/templates/allActivities.html',  
  data: {  
    name: 'My Activity',  
    desc: 'Fun'  
  },  
  foo: "my foooo"  
})
```

- Arbitrary property added to configuration object
- Prototypically inherited by child states

Note

"data" could be anything

Access data/foo property in controller

```
.controller('ActivitiesCtrl', function($state){  
  console.log($state.current.data); // object  
  console.log($state.current.foo); // my foooo  
})
```

resolve

```
.state('activities', {  
  url: '/activities',  
  controller: 'AllActivitiesController',  
  controllerAs: 'activities',  
  templateUrl: '/app/templates/allActivities.html',  
  resolve: {  
    activities: function (dataService) {  
      return dataService.getAllActivities();  
    }  
  }  
})
```

- List of dependencies to inject into the controller
- Promises must resolve before the state change occurs
- Result is injected into controller with the name specified in resolve

Working with parameters

```
$stateProvider
  .state('schools', {
    url: '/schools/:id'
  })
  .state('classrooms', {
    url: '/classrooms/{id}'
  })
  .state('activities', {
    url: '/activities',
    params: {id: { value: 42 }}
  })

// from browser
http://localhost:8080/#/classrooms/3

// access from controller
angular.module("myApp")
  .controller('MainCtrl', function($routeParams){
    var id = $routeParams.id; // :id same as in state config url
  })
```

Using \$state Service


```
// $state methods
$state.go("home");
$state.reoad() // reload current state
$state.get() // returns array of states in application

// $state properties
$state.current // reference to currently active state
$state.params // params with current active state

// Events
$stateChangeStart
$stateChangeSuccess
$stateChangeError
$stateNotFound
```


State Events

```
angular.module("myApp")
  .run(function($rootScope){
    $rootScope.$on('$stateChangeStart', function(event, toState, fromState){
      // ...
    });
    $rootScope.$on('$stateChangeSuccess', function(event, toState, fromState){
      // ...
    });
    $rootScope.$on('$stateChangeError', function(event, toState, fromState, error){
      // ...
    });
    $rootScope.$on('$stateNotFound', function(event, unfoundState, fromState){
      // ...
    });
  });
```



Abstract State

- Can't be activated
- url perpended to child state
- templateUrl must have where child view injected

```
// Parent
.state('activities', {
  url: '/activities',
  controller: 'AllActivitiesController',
  controllerAs: 'activities',
  abstract: true,
  templateUrl: '<h1>Activities </h1><div ui-view></div>',
  resolve: {
    user: function(){
      return { name: "Hemant"}
    }
  },
  myData: "Hello Fool"
})

// Child
.state('activities.details', {
  url: '/details',
  templateUrl: '/app/templates/allActivities.html',
  // resolve & myData from parent inherited
})
```