

State-Based Routing with UI-Router



Brice Wilson

@brice_wilson | www.BriceWilson.net

Overview



Configuring states with `$stateProvider`

Triggering state transitions

Using the `$stateParams` service

State configuration objects

What is a state?

Getting UI-Router

`bower install angular-ui-router#0.2.15`

`npm install angular-ui-router@0.2.15`

`https://cdnjs.cloudflare.com/ajax/libs/angular-ui-router/0.2.15/angular-ui-router.min.js`

`http://angular-ui.github.io/ui-router/release/angular-ui-router.min.js`

Getting Started with UI-Router

Reference
`angular-ui-router.js`

Add `ui.router` as a
module dependency

Add a `ui-view`
directive

Including UI-Router

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>School Buddy</title>
</head>

<body ng-app="app">
<div ui-view></div>
<!-- Angular -->
<script src="app/bower_components/angular/angular.min.js"></script>
<script src="app/bower_components/angular-ui-router/release/angular-ui-router.min.js">
</script>

</body>
</html>
```



ui.router Module Dependency

```
// app.js
```

```
(function() {
```


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

```
    // module configuration code goes here...
```

```
})();
```




Configuring States


```
(function() {  
    var app = angular.module('app', ['ui.router']);  
    app.config(['$stateProvider', function ($stateProvider) {  
        $stateProvider  
        .state('home', {  
            url: '/',  
            controller: 'HomeController',  
            controllerAs: 'home',  
            templateUrl: 'app/templates/home.html'  
        });  
    }]);  
})();
```


Configuring States

```
(function() {  
    var app = angular.module('app', ['ui.router']);  
    app.config(['$stateProvider', function ($stateProvider) {  
        $stateProvider  
            .state('home', {  
                url: '/',  
                controller: 'HomeController',  
                controllerAs: 'home',  
                templateUrl: 'app/templates/home.html'  
            });  
    }]);  
})();
```



Configuring States

```
(function() {  
    var app = angular.module('app', ['ui.router']);  
    app.config(['$stateProvider', function ($stateProvider) {  
        $stateProvider  
            .state('home', {  
                url: '/',  
                controller: 'HomeController',  
                controllerAs: 'home',  
                templateUrl: 'app/templates/home.html'  
            });  
    }]);  
})();
```


Configuring States

```
(function() {  
  
    var app = angular.module('app', ['ui.router']);  
  
    app.config(['$stateProvider', function ($stateProvider) {  
        $stateProvider  
            .state('home', {  
                url: '/',  
                controller: 'HomeController',  
                controllerAs: 'home',  
                templateUrl: 'app/templates/home.html'  
            });  
    }]);  
  
})();
```




controller : 'HomeController as home'

Configuring States

```
(function() {  
    var app = angular.module('app', ['ui.router']);  
    app.config(['$stateProvider', function ($stateProvider) {  
        $stateProvider  
            .state('home', {  
                url: '/',  
                controller: 'HomeController',  
                controllerAs: 'home',  
                templateUrl: 'app/templates/home.html'  
            });  
    }]);  
})();
```

Configuring States

```
(function() {  
    var app = angular.module('app', ['ui.router']);  
    app.config(['$stateProvider', function ($stateProvider) {  
        $stateProvider  
            .state('home', {  
                url: '/',  
                controller: 'HomeController',  
                controllerAs: 'home',  
                templateUrl: 'app/templates/home.html'  
            });  
    }]);  
})();
```



Activating a State

```
$state.go('classrooms');
```

```
<a ui-sref="classrooms">Classrooms</a>
```

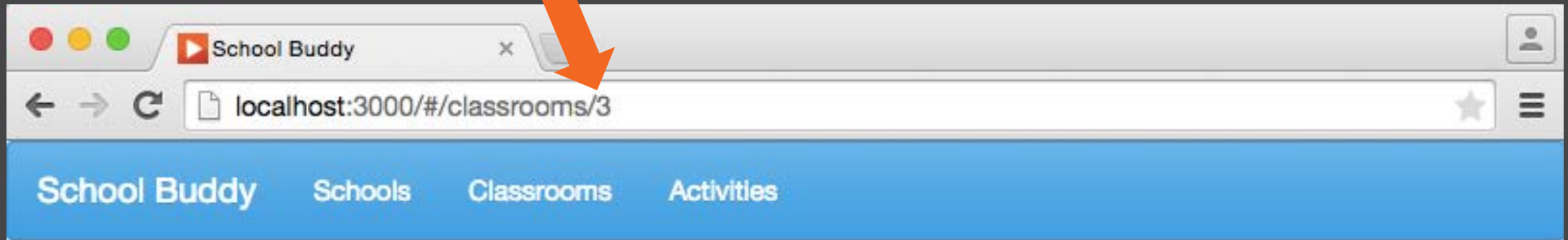
```
<a href="/classrooms">Classrooms</a>
```

Using the \$state Service

- Methods
 - go()
 - reload()
 - get()
- Properties
 - current
 - params
- Events
 - \$stateChangeError
 - \$stateChangeStart
 - \$stateChangeSuccess
 - \$stateNotFound

Working with State Parameters

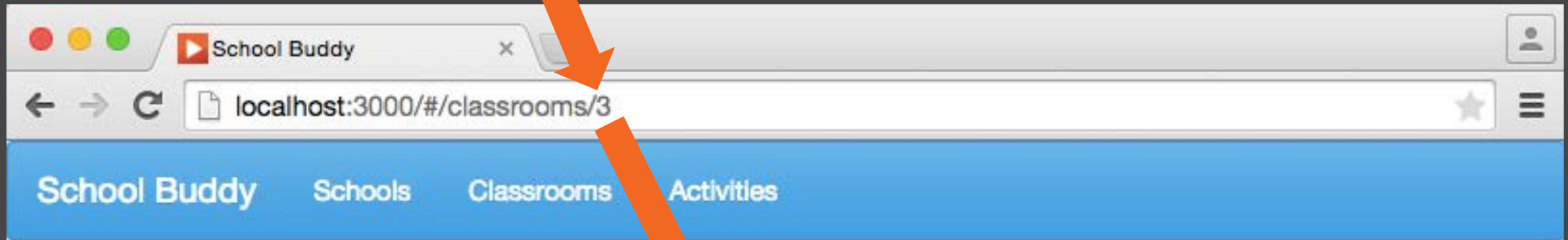
```
.state('classrooms', {  
  url: '/classrooms/:id'  
  // ...  
})
```



```
function ClassroomController($stateParams) {  
  var classroomID = $stateParams.id;  
  // use classroomID to gather data about that classroom  
}
```


Working with State Parameters

```
.state('classrooms', {  
  url: '/classrooms/:id'  
  // ...  
})
```



```
function ClassroomController($stateParams) {  
  var classroomID = $stateParams.id;  
  // use classroomID to gather data about that classroom  
}
```

Declaring Parameters

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


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



Resolve Property

- 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

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



Assigning Data to States

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

Advanced Configuration

- Templates
 - templateUrl Property
 - templateProvider Property
- Controllers
 - controller Property
 - controllerProvider Property
- onEnter and onExit Properties

Summary



Include UI-Router

Configuring states with `$stateProvider`

Pass parameters to controllers

Resolve and data properties