

# Paul Beresuita

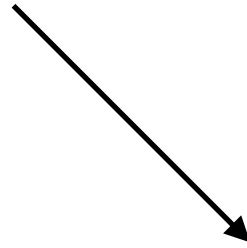


<https://github.com/paulberesuita>

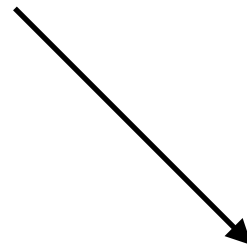


JavaScript

Mocha



LiveScript



JavaScript



JavaScript

2 weeks



METEOR





<http://todomvc.com/>



Why Angular?

<demo>

**<helloworld>**

</demo>



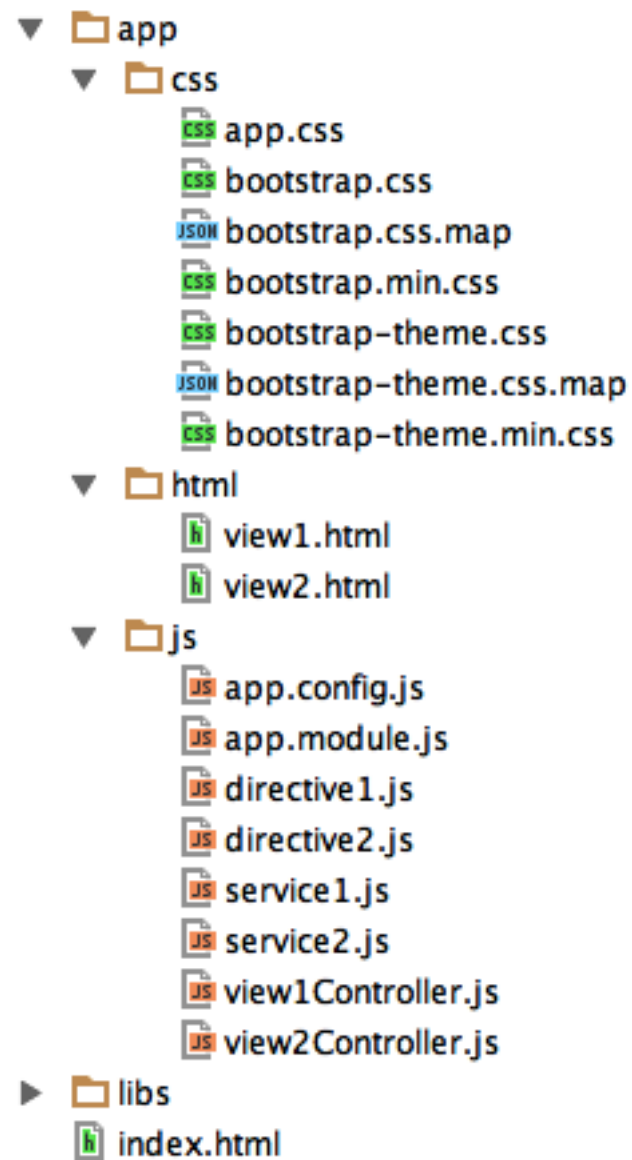


# Best Practices

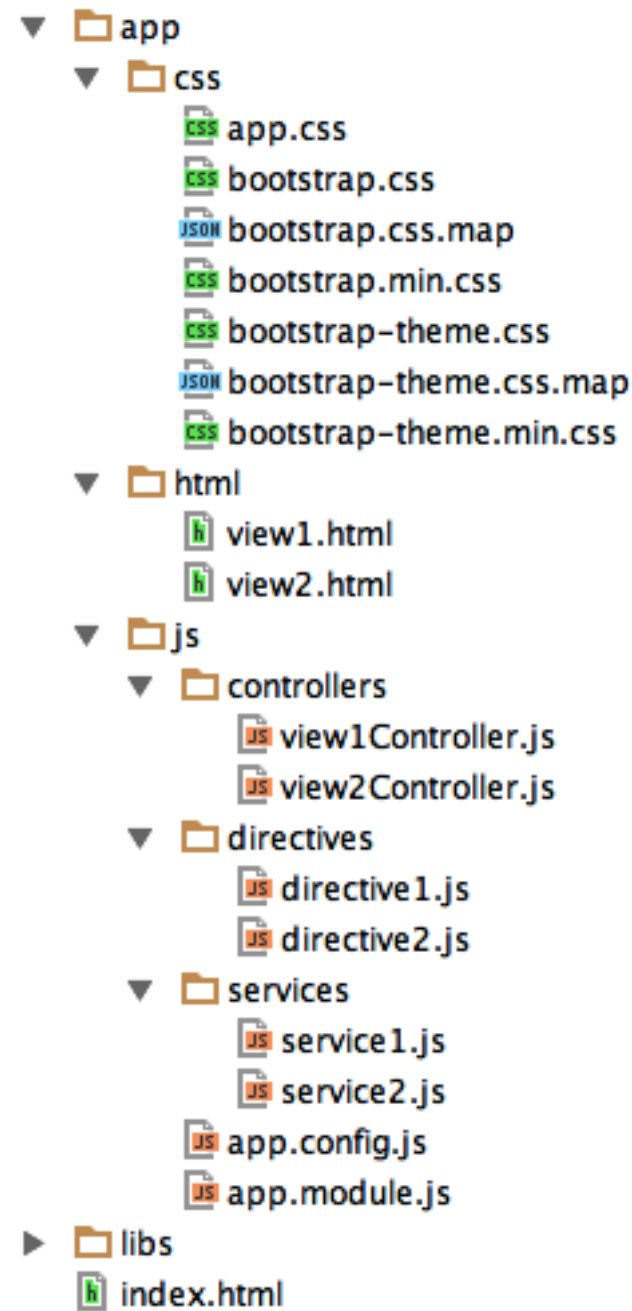


# Project Structure

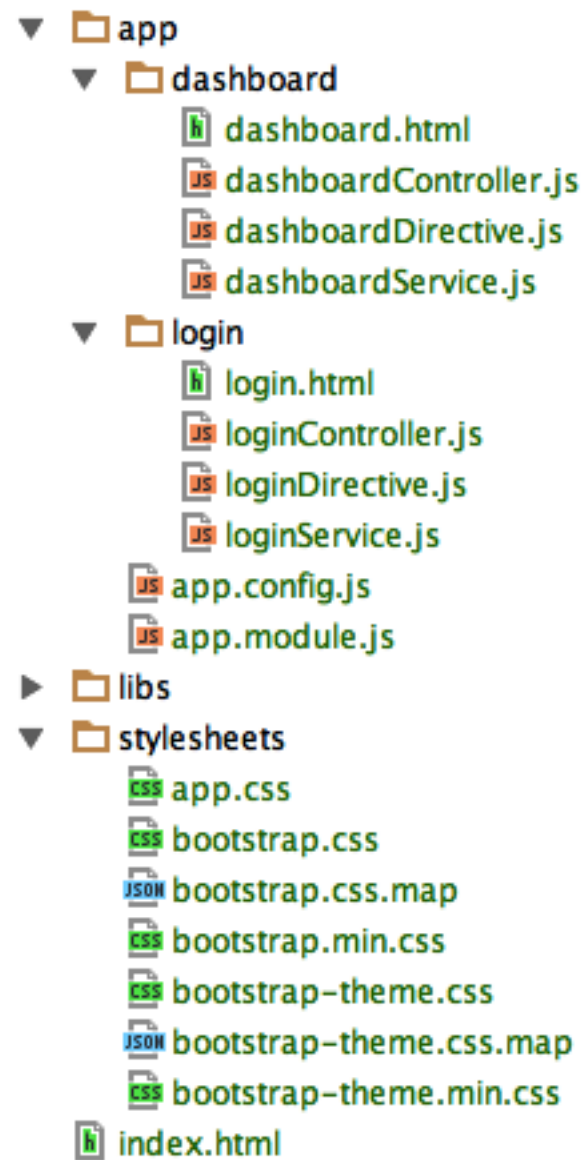
# File Type



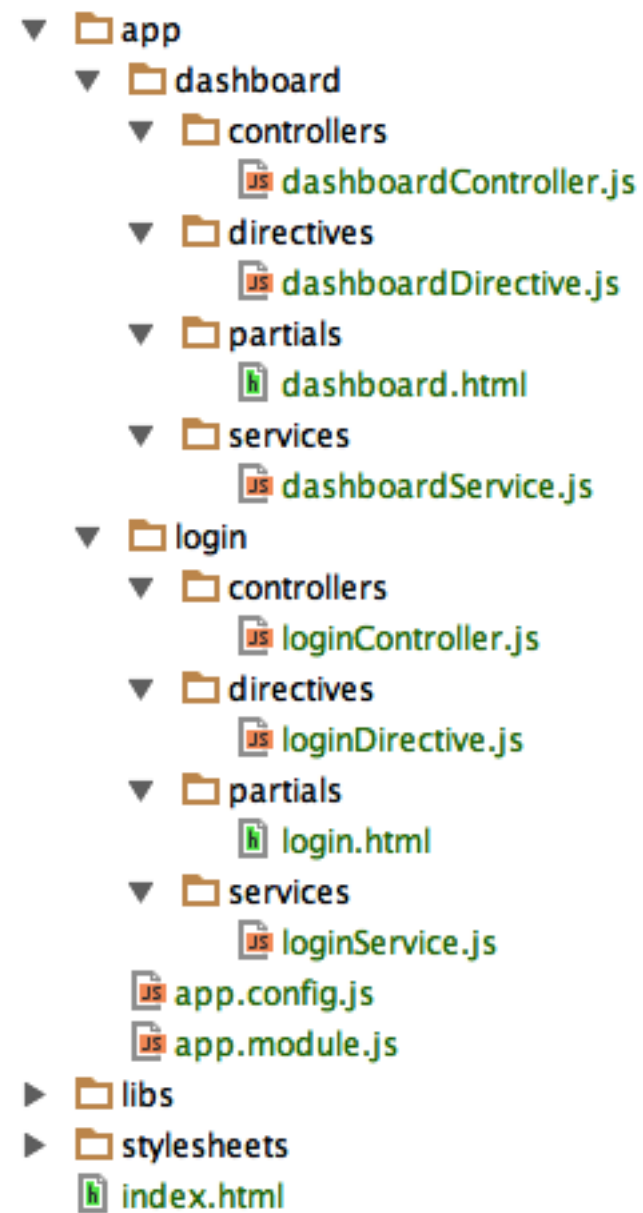
# File Type - Specific



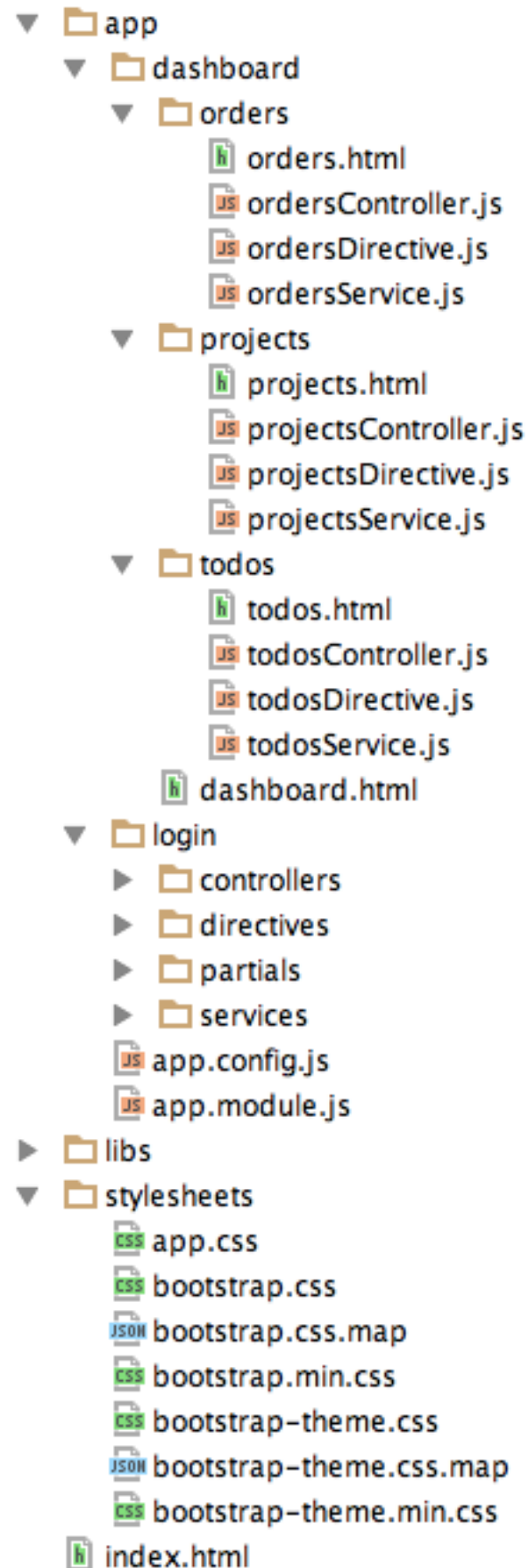
# File Type - Feature



# File Type - Feature & Type



# File Type - Feature & Sub Features





Minification



<demo>

**<minification>**

</demo>



Immediately Invoked Function Expression (IIFE)

```
(function() {  
    'use strict';  
    angular.module('app').config(function($stateProvider, $urlRouterProvider) {  
        $urlRouterProvider.otherwise("/main");  
        $stateProvider  
            .state('login', {  
                url: "/login",  
                templateUrl: "app/login/login.html",  
                controller: 'LoginController'  
            })  
            .state('main', {  
                url: "/main",  
                templateUrl: "app/main/main.html",  
                controller: 'MainController'  
            })  
    });  
})();
```

# Controllers

Use `controllerAs`  
Syntax

## View

```
<!-- avoid -->
<div ng-controller="Customer">
  {{ name }}
</div>
```

```
<!-- recommended -->
<div ng-controller="Customer as customer">
  {{ customer.name }}
</div>
```

## Controller

```
/* avoid */
function Customer($scope) {
  $scope.name = {};
  $scope.sendMessage = function() { };
}
```

```
/* recommended - but see next section */
function Customer() {
  this.name = {};
  this.sendMessage = function() { };
}
```

```
/* avoid */  
function Customer() {  
    this.name = {};  
    this.sendMessage = function() { };  
}
```

```
/* recommended */  
function Customer() {  
    var vm = this;  
    vm.name = {};  
    vm.sendMessage = function() { };  
}
```

Bindable  
members on top



```

/* avoid */
function Sessions() {
    var vm = this;

    vm.gotoSession = function() {
        /* ... */
    };
    vm.refresh = function() {
        /* ... */
    };
    vm.search = function() {
        /* ... */
    };
    vm.sessions = [];
    vm.title = 'Sessions';
}

```

```

/* recommended */
function Sessions() {
    var vm = this;

    vm.gotoSession = gotoSession;
    vm.refresh = refresh;
    vm.search = search;
    vm.sessions = [];
    vm.title = 'Sessions';

    ///////////

    function gotoSession() {
        /* */
    }

    function refresh() {
        /* */
    }

    function search() {
        /* */
    }
}

```

Don't pollute the  
controller

<demo>

**<pollutedController>**

</demo>

# 5 Directives

Use a prefix

1 file per  
directive

Manipulate the  
DOM in a directive



# Angular 2.0





Angular 2.0 is written in  
TypeScript



Angular 2.0 has no controllers

No 2-way data binding (sort of)

# Resources



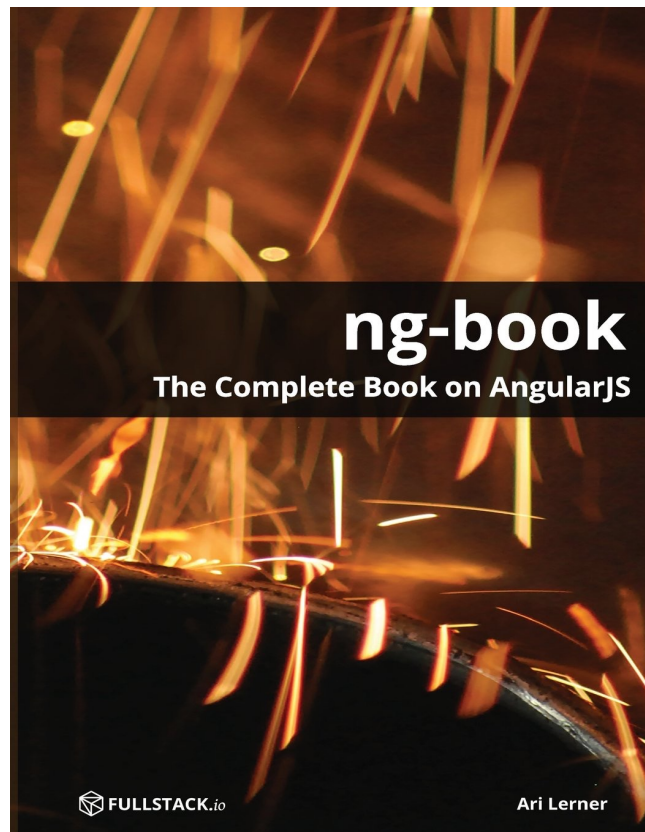
**Docs**



**Angular Best Practices**

- <https://www.airpair.com/angularjs/posts/preparing-for-the-future-of-angularjs>
- <http://blog.mgechev.com/2015/04/06/angular2-first-impressions/>
- <http://angularjs.blogspot.com/2015/03/forms-in-angular-2.html> (angular 2.0 - forms)
- <https://github.com/johnpapa/angular-styleguide#controllers>

# Learning Materials



egghead.io