



AngularJS – The Big Picture



© Wahlin Consulting – All Rights Reserved



The Big Picture

Views, Directives and Filters

Controllers, Scope and Modules

Routing

Services and Factories

Animation



Dan Wahlin



Blog

<http://weblogs.asp.net/dwahlin>



Twitter

@DanWahlin

Target Audience

- **Developers with experience using the following:**
 - HTML
 - CSS
 - JavaScript
- **Expect to see a lot of JavaScript throughout the course!**

SPA Overview

Jumping into AngularJS

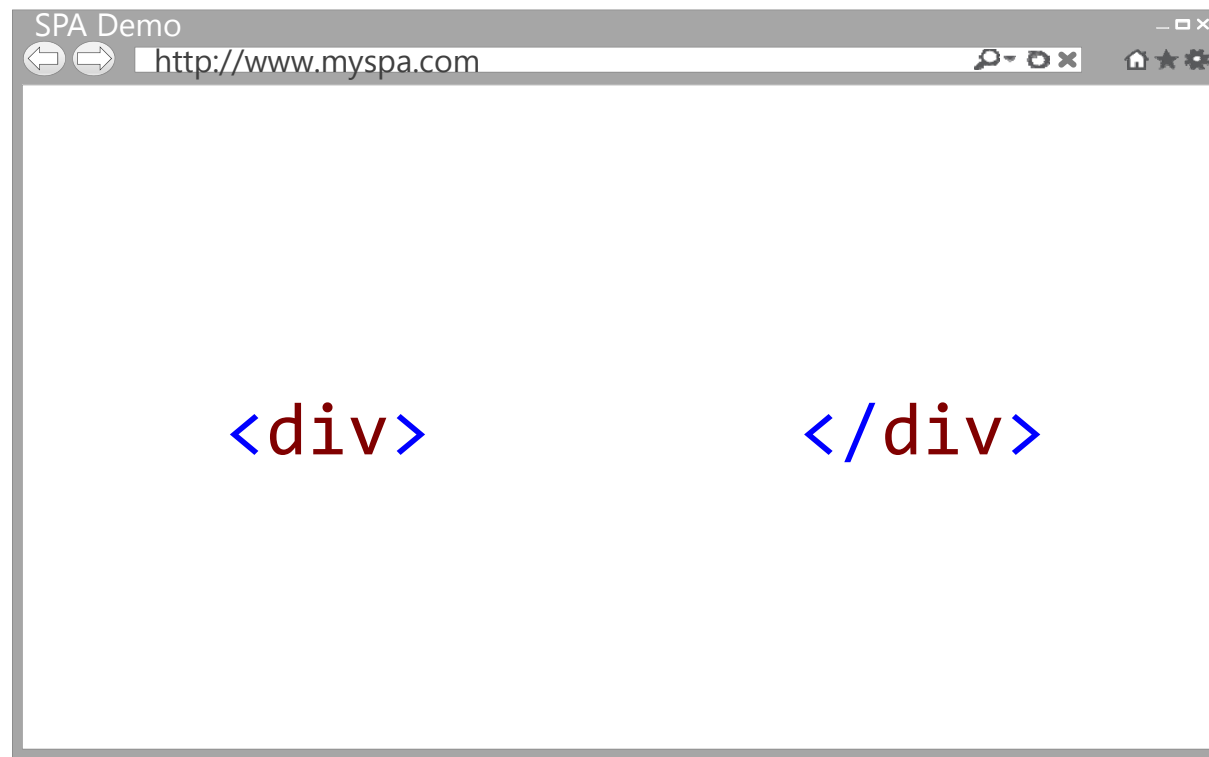
Key Players in AngularJS

AngularJS Documentation

SPA Overview

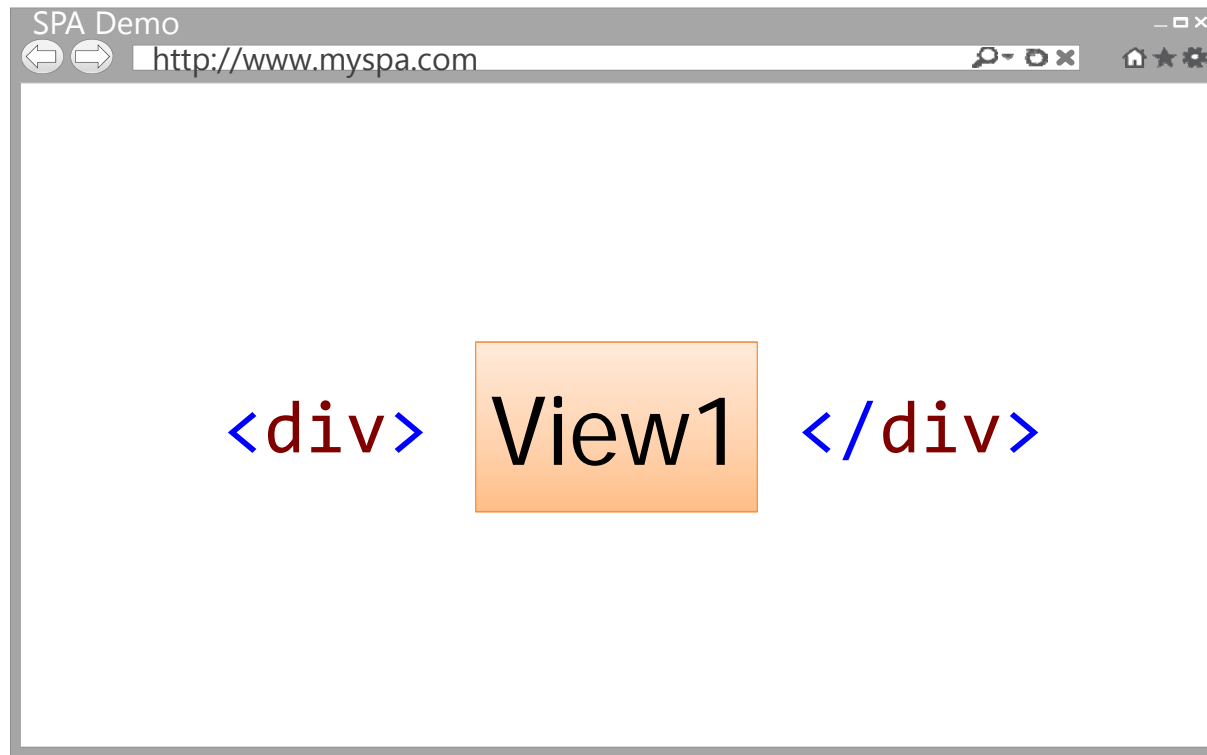
Single Page Applications

SPAs allow different views (screens) to be loaded into a shell page as the user interacts with the page



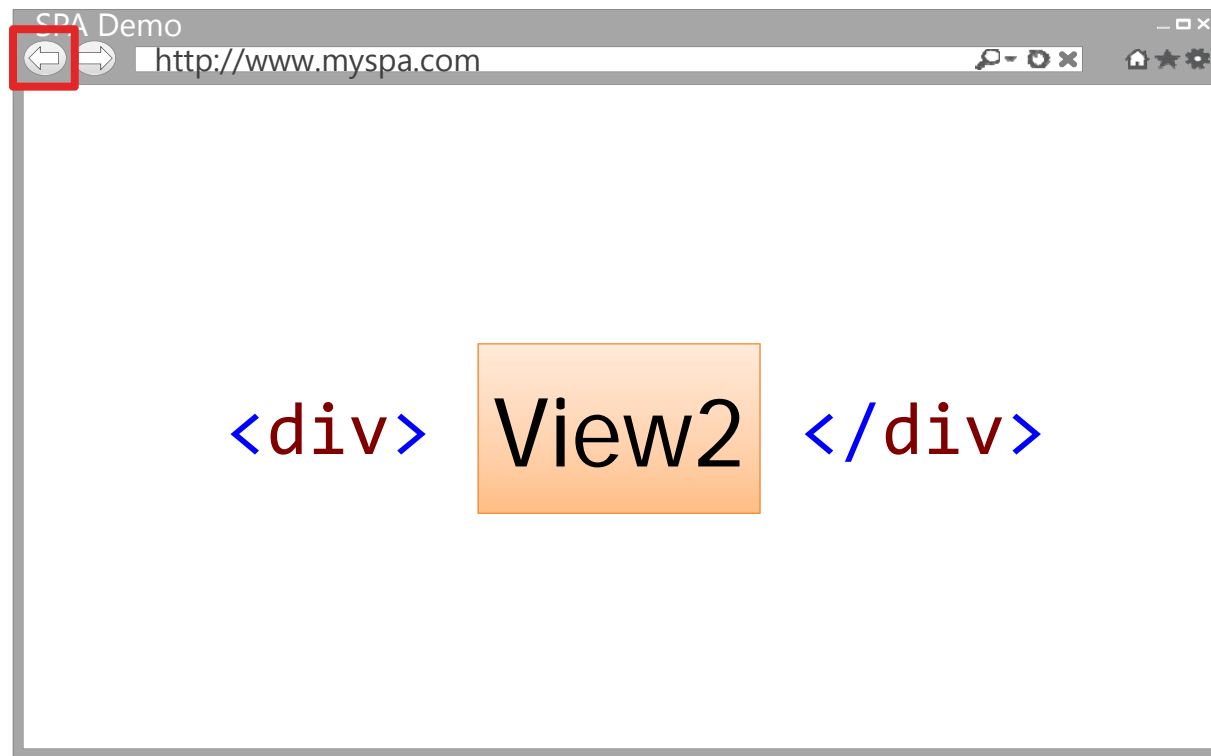
Single Page Application Views

Views can be replaced with other views



Single Page Applications

SPAs maintain a history of views that have been displayed



The Challenge with SPAs

- **SPAs rely on many different technologies:**
 - DOM manipulation
 - History
 - Routing
 - Ajax
 - Data Binding
 - More...





<http://angularjs.org>

Full-Featured SPA Framework

Data Binding

MVC

Routing

Testing

jqLite

Templates

History

Factories



AngularJS is a full-featured
SPA framework

ViewModel

Controllers

Views

Directives

Services

Dependency Injection

Validation

Jumping into AngularJS

Getting Started with AngularJS

- **Steps to get started with AngularJS:**
 1. Reference the AngularJS script in an HTML page
 2. Add an **ng-app** directive
 3. Bind data using directives



Step 1: Reference the AngularJS Script

- Add a reference to a local or hosted version of AngularJS:

```
<script src="/Scripts/angular.js"></script>
```

OR

```
<script  
  src="https://ajax.googleapis.com/ajax/libs/angularjs/  
    [version]/angular.min.js"></script>
```

Step 2: Add the ng-app Directive

- Add the *ng-app* directive into the HTML page:

```
<!DOCTYPE html>
```

```
<html ng-app>
```

```
...
```

```
</html>
```


Step 3: Bind Data using Directives

- Use AngularJS directives to add data binding functionality into a page:

Data Binding
Expression

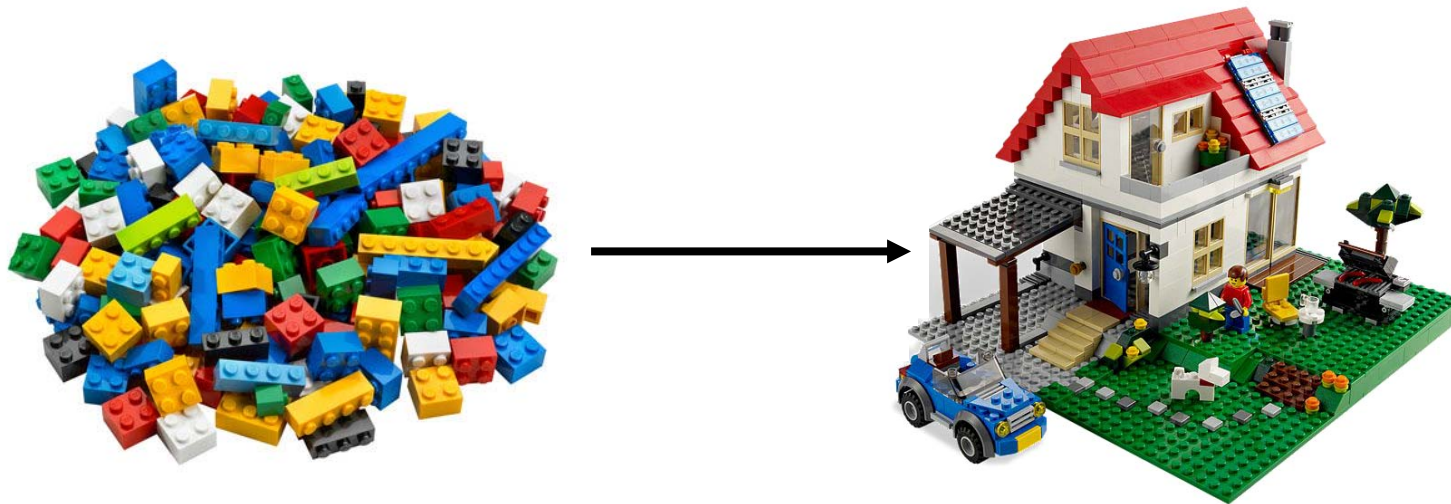
Name: `<input type="text" ng-model="name" />{{ name }}`

AngularJS Directive

Key Players in AngularJS

The Key to Learning AngularJS

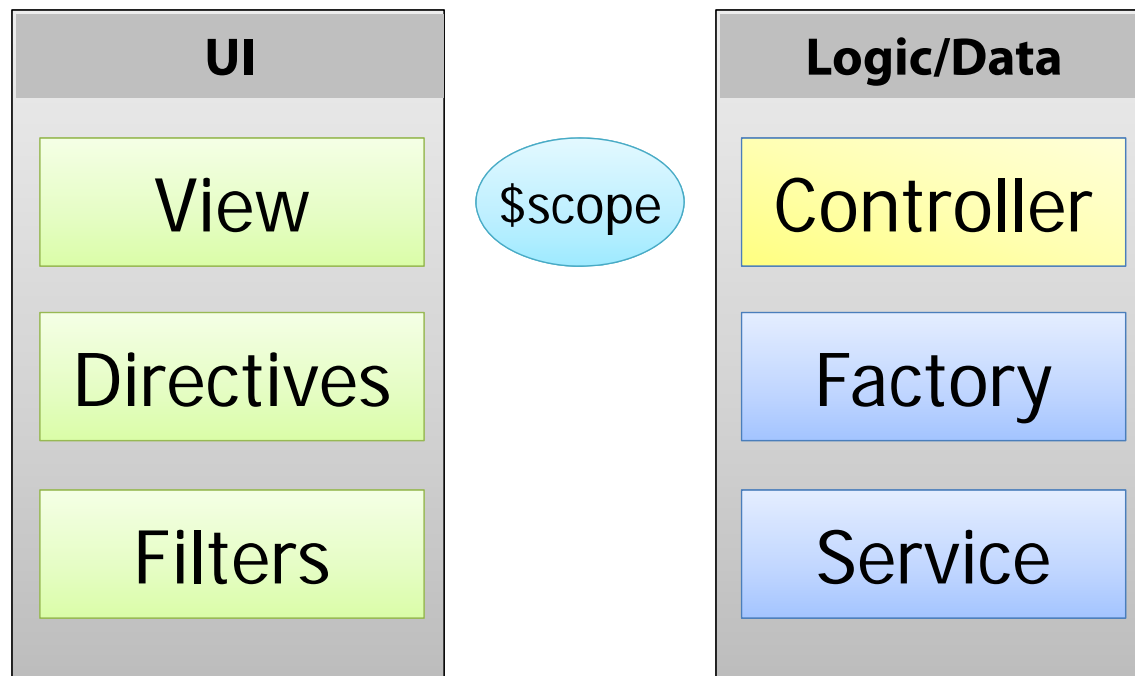
1. Understand the available parts
2. Understand how the parts fit together
3. Understand how to organize parts



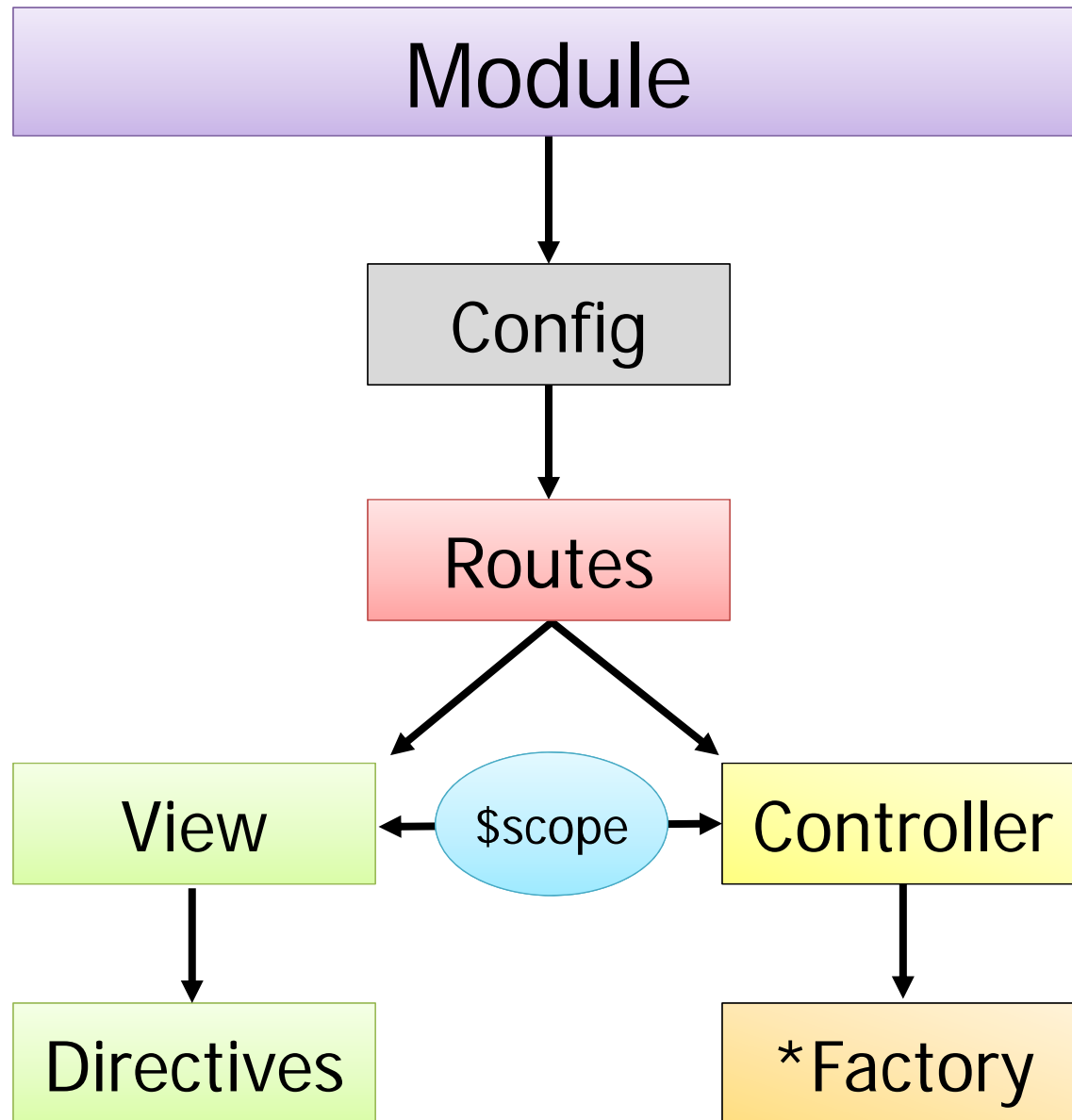
Key Players in AngularJS

Module

Routes



The Big Picture



What are Modules?

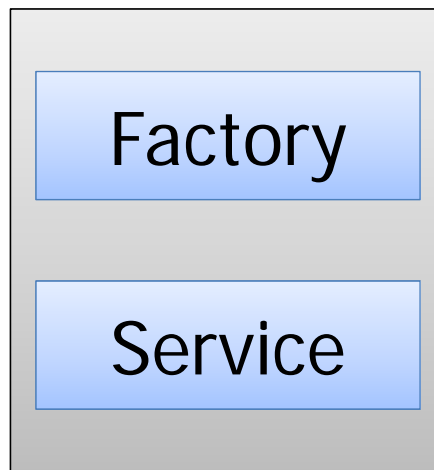
Module

Modules are containers for:

- **Controllers**
- **Routes**
- **Factories/Services**
- **Directives**
- **Filters**



What are Factories/Services?

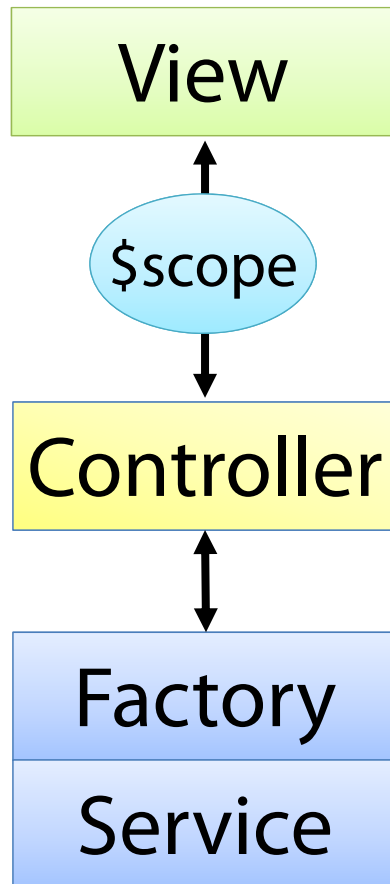


Factories and Services:

- Used to make RESTful calls
- Used to share data between controllers
- Used to handle custom logic
- Are Singletons



What are Controllers?



Controllers act as the "brain" for a view:

- Retrieve data from a factory/service and store it
- Handle events triggered by the view
- Know how to handle custom logic
- Rely on the \$scope object to interact with the view



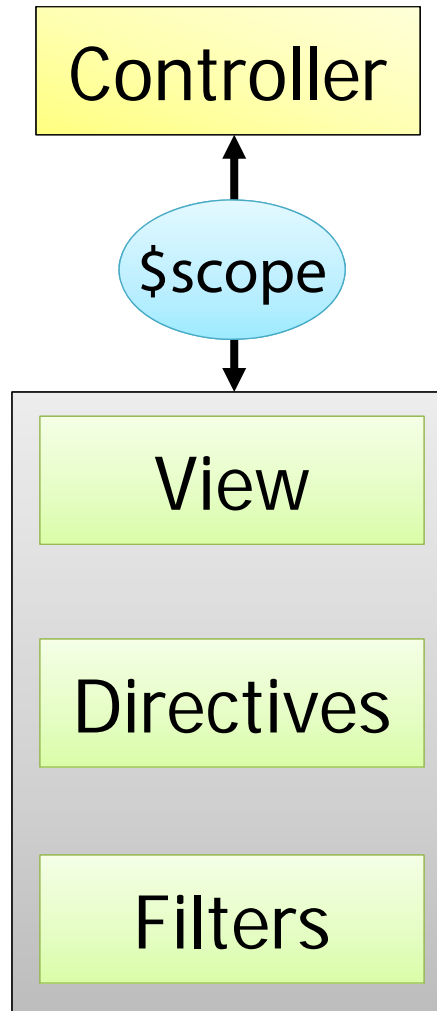
What is Scope?



**\$scope is the "glue" (ViewModel)
between a controller and a view**



What are Views?



Views render the user interface:

- Contain HTML and CSS
- Bind to data provided by a controller via the \$scope object
- Use directives to enhance HTML and render data

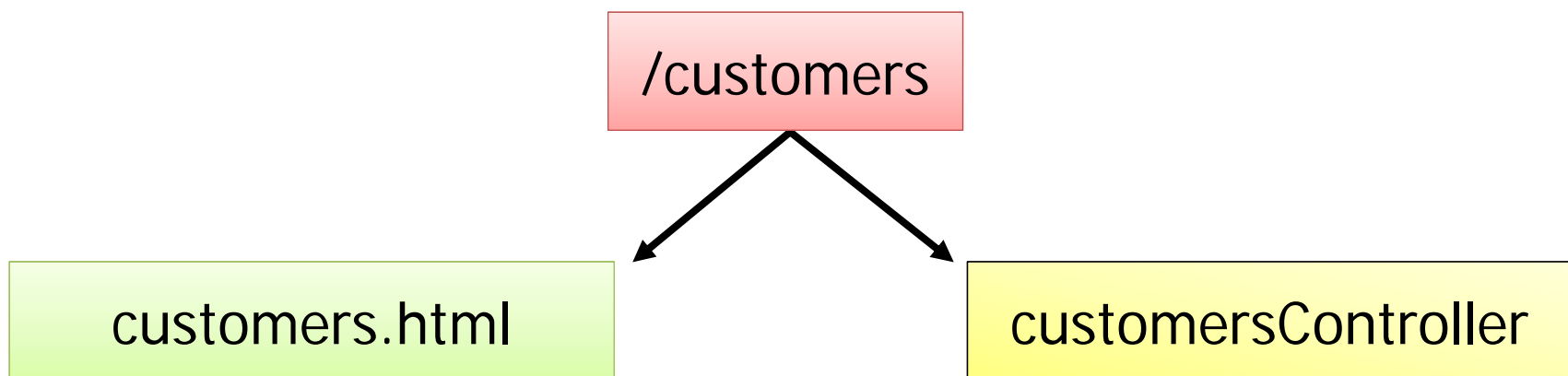


What are Routes?

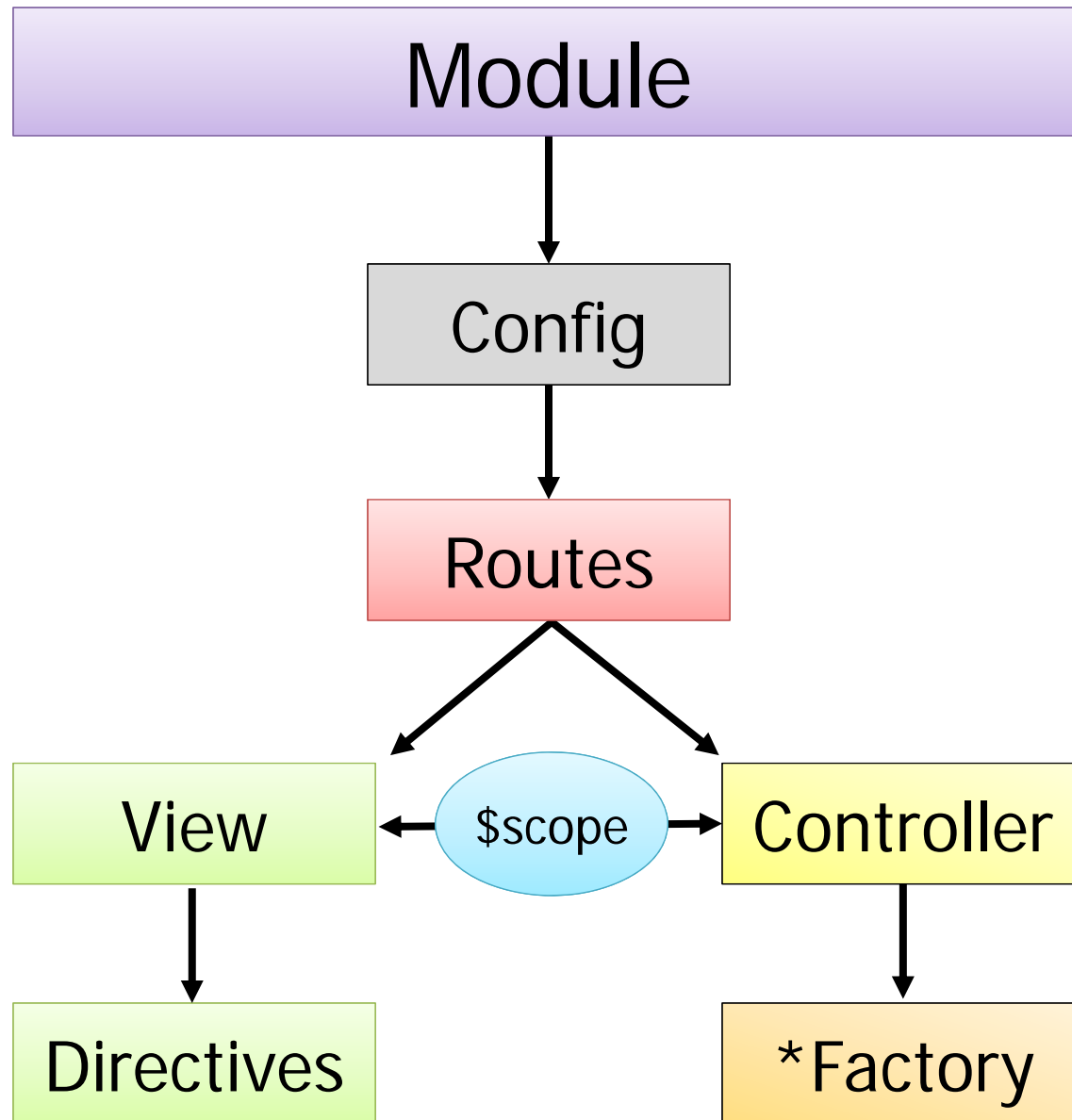
Routes

Each route has a unique path:

- Reference a controller and view
- Can include route parameters:
`/customers/:customerId`



The Big Picture



AngularJS Documentation

Summary

Summary

- **AngularJS is a full-featured SPA framework**
- **Applications are organized into one or more containers (modules)**
- **Different parts can be associated with a module:**
 - Controllers
 - Factory/Service
 - Directives
 - Filters