AngularJS

WHAT A LOVELY DAY.
MAX
FURY ROAD

Course Contents

Course Contents

Introduction

AngularJS is an open-source web application framework maintained by Google and a community of developers to address many of the challenges encountered in developing single-page applications.

AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

SPA Challenges

DOM Manipulation

Module Loading

Caching

Data Binding

View Loading

History

Routing

Object Modeling

AJAX / Promises

SPA Framework

Data Binding	MVC	Routing
Testing	jqLite	Templates
History	Factories	ViewModel
Controllers	Views	Directives
Services	Dependency Injection	Validation

MAD MAX
FURY ROAD

Directives

A directive is really a way to teach HTML new tricks. Angular allows us to extend HTML very easily by simply adding attributes, elements or comments.

ng-app: This directive defines an AngularJS application. We use this directive to auto-bootstrap an AngularJS application. It designates the root element of the application.

ng-model: The ngModel directive binds an input, select, textarea (or custom form control) to a property on the scope using NgModelController, which is created and exposed by this directive.

Data-Binding

Data-Binding is the automatic synchronization of data between the model and view components.

Treat the model as the single-source-of-truth in your application. The view is a projection of the model at all times.

When the model changes, the view reflects the change, and vice versa. This form of data-binding is referred to as "Two-Way Data-Binding".

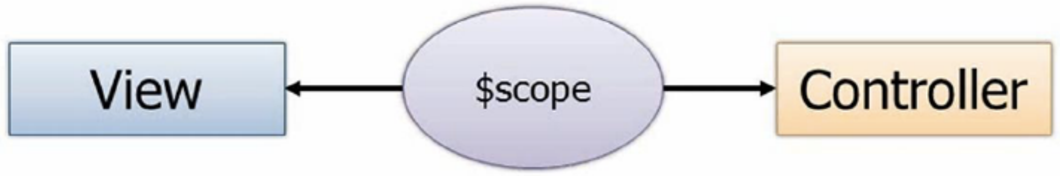
Iterators

The ngRepeat directive instantiates a template once per item from a collection.

Each template instance gets its own scope, where the given loop variable is set to the current item, and $index is set to the item index or key.

Views

View, Controllers and Scope



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

Controllers

The Controller will drive things. It will ultimately control what data gets bound into the view.

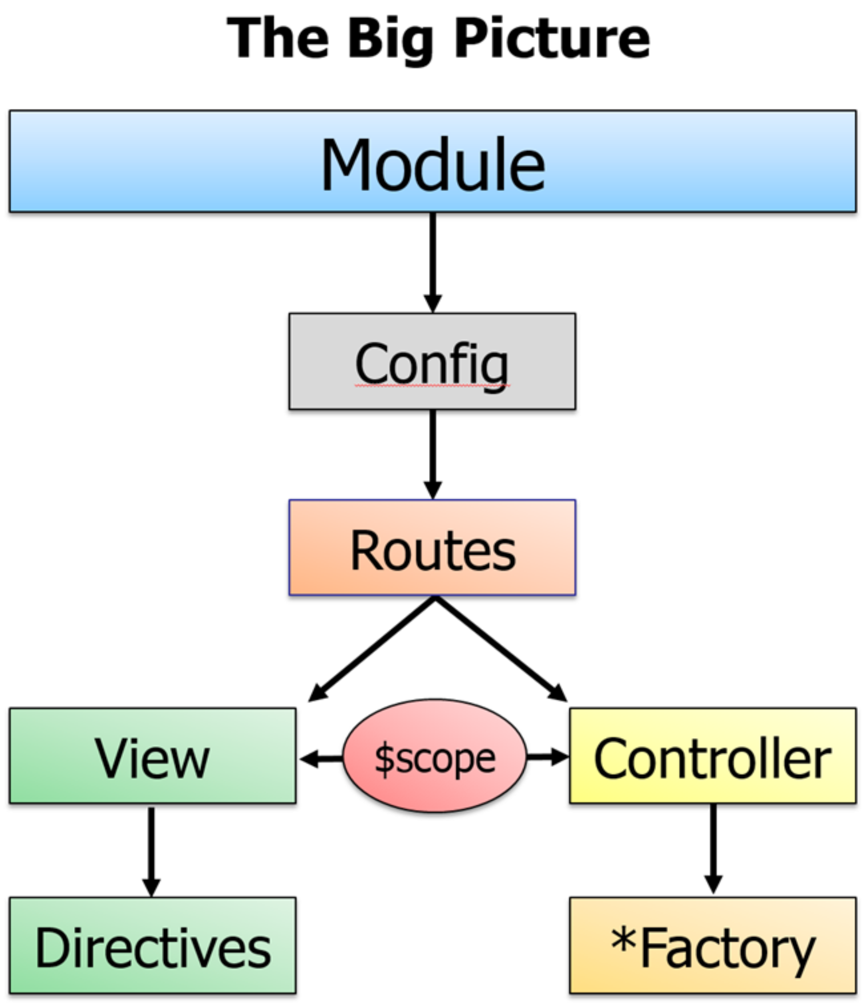
If the view passes data to the controller, it may handle passing control to a service which may then update a back-end data store.

Scope

The glue between the view and the controller is called the scope. In Angular, the scope object is represented by $scope.

A ViewModel literally is the model - the data - for the view. The scope is our ViewModel and it is the glue between the view and the controller.

Modules



The Big Picture
Module
Config
Routes
View
$scope
Controller
Directives
*Factory

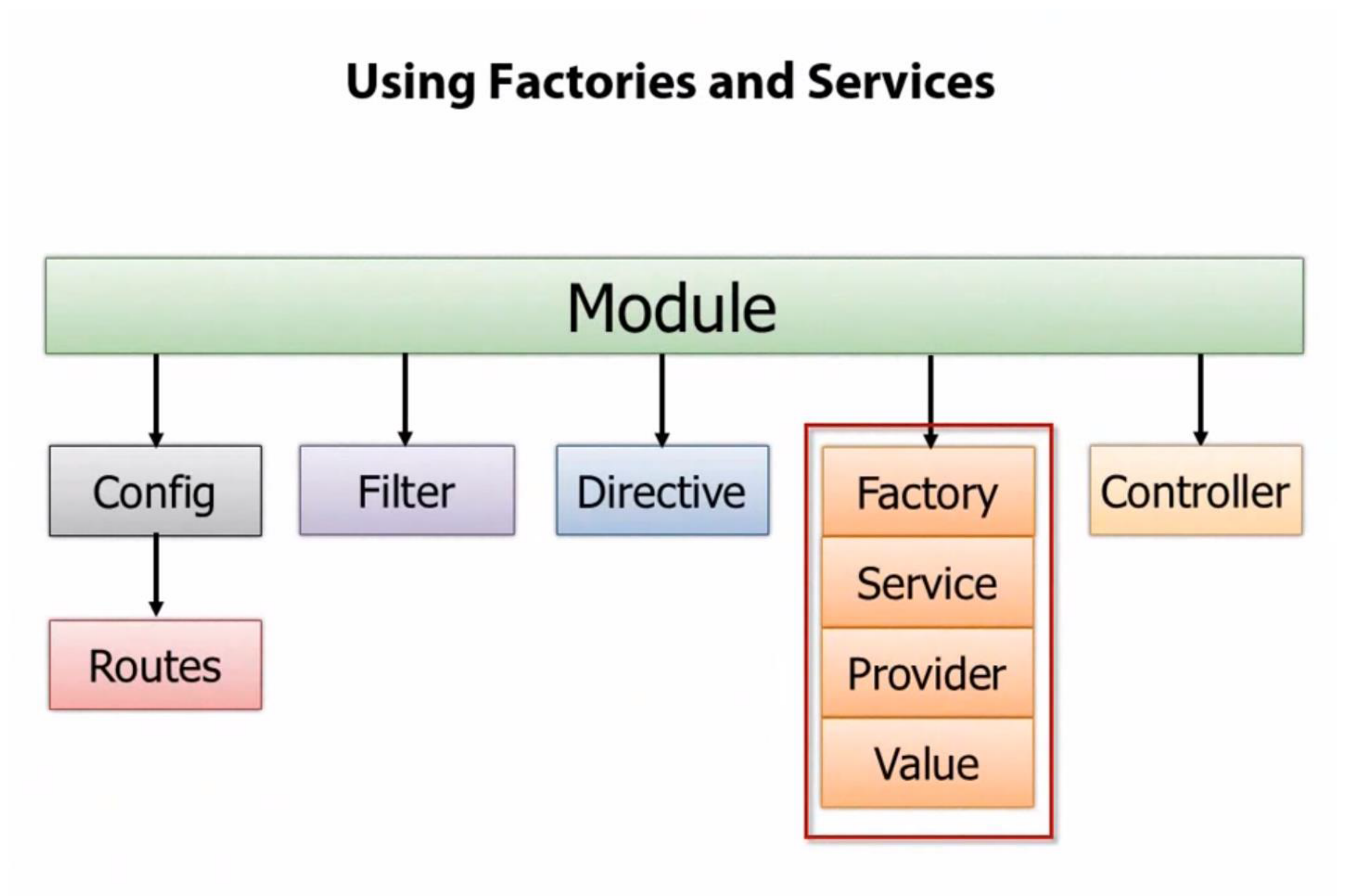
Routes

Routing helps you divide your application into logical views and bind different views to controllers.

In Angular, the "magic" of routing is taken care by a service provider called $routeProvider.

Note: The ng-view directive is a placeholder for views. Each view referred by the route is loaded in the ng-view section of the page.

Factories


Using Factories and Services
Module
Config
Filter
Directive
Factory
Service
Provider
Value
Controller
Routes

Factory vs Service vs Provider

Factory: Create an object, add properties to it, then return that same object. When you pass this service into your controller, those properties on the object will now be available in that controller through your factory.

Service: Instantiate service with the ‘new’ keyword. Add properties to ‘this’ and return ‘this’. When you pass the service into your controller, those properties will be available on that controller through your service.

Provider: Providers are the only service you can pass into your .config() function. Use a provider when you want to provide module-wide configuration for your service object before making it available.

Unit Testing

Jasmine: Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks. It does not require a DOM. And it has a clean, obvious syntax so that you can easily write tests.

Karma: Karma is a tool which spawns a web server that executes source code against test code for each of the connected browsers. The results for each test against each browser are examined and displayed via the command line.

Code Coverage

Karma can generate code coverage using awesome Istanbul. If you want to generate the coverage, you need to configure up to three parts:

- Preprocessor Coverage (Required)
- Reporter Coverage (Required)
- Reporter Options (Optional)

References

https://angularjs.org

http://fastandfluid.com/publicdownloads/AngularJSIn60MinutesIsh_DanWahlin_May2013.pdf

http://jasmine.github.io

http://karma-runner.github.io

https://github.com/vchandnani/rangoli