

# AngularJS 2 学习指南

岚剑 译

2016/5/11

# 目录

ANGULARJS 2 学习指南 .....	1
目录 .....	2
修订 .....	11
预发布版本 .....	11
错误指正 .....	11
参与社区 .....	11
通过 TWITTER 订阅更新 .....	11
1. 第一个 ANGULARJS 2 WEB 应用 .....	12
1.1. 简单网站克隆 .....	12
1.2. 开始 .....	12
1.2.1. TypeScript .....	12
1.2.2. 项目示例 .....	12
1.2.3. 依赖 .....	12
1.2.4. 依赖加载 .....	12
1.2.5. 添加 CSS .....	12
1.3. 第一个 TYPESCRIPT 文件 .....	12
1.3.1. 创建组件 .....	12
1.3.2. 添加模板 .....	12
1.3.3. 启动应用 .....	12
1.3.4. 系统导出 .....	12
1.4. 运行应用 .....	12
1.4.1. 编译 TypeScript .....	13
1.4.2. 使用 npm .....	13
1.4.3. 通过 server 启动 app .....	13
1.4.4. 热启动 .....	13
1.5. 给组件添加数据 .....	13
1.6. 使用数组 .....	13
1.7. 扩展应用 .....	13
1.7.1. 应用组件 .....	13
1.7.2. 添加注入 .....	13

1.7.3. 添加一个组件.....	13
1.8. 渲染多行 .....	13
1.8.1. 创建一个文章类.....	13
1.8.2. 文章列表类.....	13
1.8.3. 配置组件输入.....	13
1.8.4. 渲染文章列表.....	14
1.9. 添加文章 .....	14
1.10. 结尾 .....	14
1.10.1. 显示文章主题 .....	14
1.10.2. 排序 .....	14
1.11. 代码 .....	14
1.12. 总结 .....	14
1.13. 提问 .....	14
<b>2. TYPESCRIPT .....</b>	<b>15</b>
2.1. ANGULAR 2 内置 TYPESCRIPT .....	15
2.2. 为什么 TYPESCRIPT.....	15
2.3. 类型.....	15
2.3.1. 测试一下.....	15
2.4. 内置类型 .....	15
2.5. 类.....	15
2.5.1. 属性.....	15
2.5.2. 方法.....	15
2.5.3. 构造函数.....	15
2.5.4. 继承.....	15
2.6. 通用性 .....	15
2.6.1. 富数组函数.....	15
2.6.2. 模板字符串.....	15
2.7. 总结 .....	16
<b>3. ANGULARJS 原理 .....</b>	<b>17</b>
3.1. 应用示例 .....	17
3.1.1. 导航组件.....	17
3.1.2. 面包屑组件.....	17

3.1.3. 列表组件.....	17
3.2. 商品模型 .....	17
3.3. 组件.....	17
3.4. 组件装饰器 .....	17
3.4.1. 选择器.....	17
3.4.2. 模板.....	17
3.4.3. 添加一个商品.....	17
3.4.4. 绑定模板.....	17
3.4.5. 添加多个商品.....	17
3.4.6. 使用<product-list>显示列表.....	17
3.5. PRODUCTSList 组件 .....	18
3.5.1. @Component 配置项.....	18
3.5.2. 组件输入.....	18
3.5.3. 组件输出.....	18
3.5.4. 触发自定义事件.....	18
3.5.5. ProductsList 控制类.....	18
3.5.6. ProductsList 视图模板.....	18
3.5.7. ProductsList 全部代码.....	18
3.6. PRODUCTRow 组件.....	18
3.6.1. ProductRow 配置.....	18
3.6.2. ProductRow 定义类.....	18
3.6.3. ProductRow 模板.....	18
3.6.4. ProductRow 代码.....	18
3.7. PRODUCTImage 组件 .....	18
3.8. PRICEDisplay 组件 .....	19
3.9. PRODUCTDepartment 组件 .....	19
3.10. 项目全部 .....	19
3.11. 数据结构概要.....	19
<b>4. 内置组件 .....</b>	<b>20</b>
4.1. 概述.....	20
4.2. NgIf .....	20
4.3. NgSwitch .....	20

4.4.	NGSTYLE .....	20
4.5.	NGCLASS .....	20
4.6.	NGFOR .....	20
4.6.1.	获取索引 .....	20
4.7.	NGNONBINDABLE .....	20
4.8.	总结 .....	20
5.	表单 .....	21
5.1.	表单的复杂严格性 .....	21
5.2.	CONTROLS && CONTROL GROUPS .....	21
5.2.1.	Control .....	21
5.2.2.	ControlGroup .....	21
5.3.	第一个表单例子 .....	21
5.3.1.	SKU 表单: @Component 注解 .....	21
5.3.2.	SKU 表单: 模板 .....	21
5.3.3.	SKU 表单: 组件定义类 .....	21
5.4.	使用 FORMBUILDER .....	21
5.4.1.	FormBuilder .....	21
5.4.2.	视图中使用 myForm .....	21
5.4.3.	尝试一下 .....	21
5.5.	添加校验 .....	21
5.5.1.	sku Control 作为实例变量 1 .....	22
5.5.2.	sku Control 作为实例变量 2 .....	22
5.5.3.	自定义校验 .....	22
5.6.	检测变化 .....	22
5.7.	NGMODEL .....	22
5.8.	小结 .....	22
6.	数据结构 .....	23
6.1.	数据结构概述 .....	23
6.1.1.	数据结构 .....	23
7.	数据结构之观察者 1 - SERVICES .....	24
7.1.	观察者与 RxJS .....	24
7.1.1.	注: RxJS 至少要求 .....	24

7.1.2. 学习响应式变成及 RxJS.....	24
7.2. 聊天 APP.....	24
7.2.1. Components.....	24
7.2.2. Models.....	24
7.2.3. Services.....	24
7.2.4. 小结.....	24
7.3. MODELS 实现.....	24
7.3.1. User.....	24
7.3.2. Thread.....	24
7.3.3. Message.....	24
7.4. USERSERVICE 实现.....	24
7.4.1. currentUser 流.....	25
7.4.2. 设置新用户.....	25
7.4.3. UserService.ts.....	25
7.5. MESSAGESERVICE.....	25
7.5.1. newMessages 流.....	25
7.5.2. message 流.....	25
7.5.3. 操作流.....	25
7.5.4. 共享流.....	25
7.5.5. message 流添加 Messages.....	25
7.5.6. MessagesService 代码.....	25
7.5.7. 尝试一下.....	25
7.6. THREADSERVICE.....	25
7.6.1. Threads 集合.....	25
7.7. 数据模型小结.....	25
<b>8. 数据结构之观察者 2 - VIEW.....</b>	<b>26</b>
8.1. CHATAPP 顶层组件.....	26
8.2. CHATTHREADS 组件.....	26
8.3. 单一 CHATTHREAD 组件.....	26
8.4. CHATWINDOW 组件.....	26
8.5. CHATMESSAGE 组件.....	26
8.6. CHATNAVBAR 组件.....	26

8.7. 小结 .....	26
8.8. 下一步 .....	26
<b>9. HTTP 请求 .....</b>	<b>27</b>
9.1. 简介 .....	27
9.2. 使用 ANGULAR2 HTTP .....	27
9.2.1. 导入 http .....	27
9.3. 简单请求 .....	27
9.3.1. SimpleHTTPComponent 添加注解 .....	27
9.3.2. SimpleHTTPComponent 模板 .....	27
9.3.3. SimpleHTTPComponent 控制器 .....	27
9.3.4. SimpleHTTPComponent 代码 .....	27
9.4. 实现 YOUTUBESEARCHCOMPONENT .....	27
9.4.1. SearchResult .....	27
9.4.2. YouTubeService .....	27
9.4.3. SearchBox .....	27
9.4.4. SearchResultComponent .....	27
9.4.5. YouTubSearchComponent .....	28
9.5. API .....	28
9.5.1. POST 请求 .....	28
9.5.2. PUT / PATCH / DELETE / HEAD .....	28
9.5.3. 配置项 .....	28
9.5.4. 小结 .....	28
<b>10. 路由 .....</b>	<b>29</b>
10.1. 为什么路由 .....	29
10.1.1. 初级: 链接标记 .....	29
10.1.2. 进阶: htm5 客户端路由 .....	29
10.2. 客户端路由原理 .....	29
10.3. 第一个路由例子 .....	29
10.4. 路由的组成部分 .....	29
10.4.1. 路由配置 RouterConfig .....	29
10.4.2. 路由出口 RouterOutlet .....	29
10.4.3. 路由链接 RouterLink .....	29

10.5. 组合在一起 .....	29
10.6. 路由策略 .....	29
10.7. 路径策略 .....	29
10.8. 结合应用运行 .....	29
10.9. 路由参数 .....	30
10.10. 音乐搜索 APP .....	30
10.11. 路由生命周期 HOOK .....	30
10.12. 嵌套路由 .....	30
10.12.1. 配置路由 .....	30
10.12.2. <i>ProductsComponent</i> .....	30
10.13. 小结 .....	30
<b>11. 高级组件 .....</b>	<b>31</b>
11.1. 样式 .....	31
11.1.1. 视图(样式)封装 .....	31
11.1.2. <i>Shadow DOM</i> 封装 .....	31
11.1.3. 不封装 .....	31
11.2. 创建 POPUP - .....	31
11.2.1. 弹出组件构成 .....	31
11.2.2. 使用 <i>ElementRef</i> .....	31
11.2.3. 绑定主体 .....	31
11.2.4. 使用 <i>exportsAs</i> 增加按钮 .....	31
11.3. 使用 TRANSLUCION 构建消息嵌套 .....	31
11.3.1. 更改主体的 CSS .....	31
11.3.2. 使用 <i>ng-content</i> .....	31
11.4. 相邻 DIRECTIVE 通信 - TABS .....	31
11.4.1. <i>Tab</i> 组件 .....	32
11.4.2. <i>Tabset</i> 组件 .....	32
11.4.3. 使用 <i>Tabset</i> .....	32
11.5. 生命周期添加 HOOK .....	32
11.5.1. <i>Oninit</i> 和 <i>OnDestroy</i> .....	32
11.5.2. <i>OnChanges</i> .....	32
11.5.3. <i>DoCheck</i> .....	32



11.5.4.	<i>AfterContentInit, AfterViewInit, AfterContentChecked, AfterViewChecked....</i>	32
11.6.	高级模板 .....	32
11.6.1.	重写 <i>ngIf – ngBookIf</i> .....	32
11.6.2.	重写 <i>ngFor – ngBookRepeat</i> .....	32
11.7.	脏值检查 .....	32
11.7.1.	自定义脏值检查 .....	32
11.7.2.	<i>Zone JS</i> .....	33
11.7.3.	观察与推送 .....	33
11.8.	小结 .....	33
12.	<b>ANGULAR 1 如何升级</b> .....	34
12.1.	外围理念 .....	34
12.2.	怎么生成 APP .....	34
12.3.	NG1 VS NG2 .....	34
12.4.	通用特性知识要求 .....	34
12.5.	ANGULAR 1 APP .....	34
12.6.	生成一个 HYBRID .....	34
12.7.	小结 .....	34
12.8.	引用 .....	34
13.	<b>测试</b> .....	35
13.1.	测试驱动 .....	35
13.2.	端到端测试 vs 单元测试 .....	35
13.3.	测试工具 .....	35
13.3.1.	<i>Jasmine</i> .....	35
13.3.2.	<i>Karma</i> .....	35
13.4.	写单元测试 .....	35
13.5.	NG2 单元测试框架 .....	35
13.6.	配置测试环境 .....	35
13.7.	SERVICES 测试 及 HTTP .....	35
13.7.1.	<i>HTTP 思考</i> .....	35
13.7.2.	<i>Stubs</i> .....	35
13.7.3.	<i>Mocks</i> .....	35
13.7.4.	<i>HTTP MockBackend</i> .....	35

13.7.5. <i>BeforeEachProviders</i> 钩子.....	36
13.7.6. 测试跟踪.....	36
13.8. 组件测试.....	36
13.8.1. <i>mock</i> 依赖.....	36
13.8.2. <i>mock</i> 路由类.....	36
13.9. 表单测试.....	36
13.10. HTTP 测试.....	36
13.11. 小结.....	36
<b>14. 修改记录.....</b>	<b>37</b>

修订

预发布版本

错误指正

参与社区

通过 **Twitter** 订阅更新

# **1. 第一个 AngularJS 2 Web 应用**

## **1.1. 简单网站克隆**

## **1.2. 开始**

### **1.2.1. TypeScript**

### **1.2.2. 项目示例**

### **1.2.3. 依赖**

### **1.2.4. 依赖加载**

### **1.2.5. 添加 CSS**

## **1.3. 第一个 TypeScript 文件**

### **1.3.1. 创建组件**

### **1.3.2. 添加模板**

### **1.3.3. 启动应用**

### **1.3.4. 系统导出**

## **1.4. 运行应用**

### **1.4.1. 编译 TypeScript**

### **1.4.2. 使用 npm**

### **1.4.3. 通过 server 启动 app**

### **1.4.4. 热启动**

## **1.5. 给组件添加数据**

## **1.6. 使用数组**

## **1.7. 扩展应用**

### **1.7.1. 应用组件**

### **1.7.2. 添加注入**

### **1.7.3. 添加一个组件**

## **1.8. 渲染多行**

### **1.8.1. 创建一个文章类**

### **1.8.2. 文章列表类**

### **1.8.3. 配置组件输入**

#### **1.8.4. 渲染文章列表**

### **1.9. 添加文章**

### **1.10. 结尾**

#### **1.10.1. 显示文章主题**

#### **1.10.2. 排序**

### **1.11. 代码**

### **1.12. 总结**

### **1.13. 提问**

## **2. TypeScript**

### **2.1. Angular 2 内置 TypeScript**

### **2.2. 为什么 TypeScript**

### **2.3. 类型**

#### **2.3.1. 测试一下**

### **2.4. 内置类型**

### **2.5. 类**

#### **2.5.1. 属性**

#### **2.5.2. 方法**

#### **2.5.3. 构造函数**

#### **2.5.4. 继承**

### **2.6. 通用性**

#### **2.6.1. 富数组函数**

#### **2.6.2. 模板字符串**

## 2.7. 总结



## **3. AngularJS 原理**

### **3.1. 应用示例**

#### **3.1.1. 导航组件**

#### **3.1.2. 面包屑组件**

#### **3.1.3. 列表组件**

### **3.2. 商品模型**

### **3.3. 组件**

### **3.4. 组件装饰器**

#### **3.4.1. 选择器**

#### **3.4.2. 模板**

#### **3.4.3. 添加一个商品**

#### **3.4.4. 绑定模板**

#### **3.4.5. 添加多个商品**

#### **3.4.6. 使用<product-list>显示列表**

## **3.5. ProductsList 组件**

### **3.5.1. @Component 配置项**

### **3.5.2. 组件输入**

### **3.5.3. 组件输出**

### **3.5.4. 触发自定义事件**

### **3.5.5. ProductsList 控制类**

### **3.5.6. ProductsList 视图模板**

### **3.5.7. ProductsList 全部代码**

## **3.6. ProductRow 组件**

### **3.6.1. ProductRow 配置**

### **3.6.2. ProductRow 定义类**

### **3.6.3. ProductRow 模板**

### **3.6.4. ProductRow 代码**

## **3.7. ProductImage 组件**

**3.8. PriceDisplay 组件**

**3.9. ProductDepartment 组件**

**3.10. 项目全部**

**3.11. 数据结构概要**

## 4. 内置组件

### 4.1. 概述

### 4.2. NgIf

### 4.3. NgSwitch

### 4.4. NgStyle

### 4.5. NgClass

### 4.6. NgFor

#### 4.6.1. 获取索引

### 4.7. NgNonBindable

### 4.8. 总结

## 5. 表单

### 5.1. 表单的复杂严格性

### 5.2. Controls && Control Groups

#### 5.2.1. Control

#### 5.2.2. ControlGroup

### 5.3. 第一个表单例子

#### 5.3.1. SKU 表单: @Component 注解

#### 5.3.2. SKU 表单: 模板

#### 5.3.3. SKU 表单: 组件定义类

### 5.4. 使用 FormBuilder

#### 5.4.1. FormBuilder

#### 5.4.2. 视图中使用 myForm

#### 5.4.3. 尝试一下

### 5.5. 添加校验

### **5.5.1. sku Control 作为实例变量 1**

### **5.5.2. sku Control 作为实例变量 2**

### **5.5.3. 自定义校验**

## **5.6. 检测变化**

## **5.7. ngModel**

## **5.8. 小结**

## **6. 数据结构**

### **6.1. 数据结构概述**

#### **6.1.1. 数据结构**

## **7. 数据结构之观察者 1 - Services**

### **7.1. 观察者与 RxJS**

#### **7.1.1. 注：RxJS 至少要求**

#### **7.1.2. 学习响应式变成及 RxJS**

### **7.2. 聊天 app**

#### **7.2.1. Components**

#### **7.2.2. Models**

#### **7.2.3. Services**

#### **7.2.4. 小结**

### **7.3. Models 实现**

#### **7.3.1. User**

#### **7.3.2. Thread**

#### **7.3.3. Message**

### **7.4. UserService 实现**



### **7.4.1.currentUser 流**

### **7.4.2. 设置新用户**

### **7.4.3. UserService.ts**

## **7.5. MessageService**

### **7.5.1.newMessages 流**

### **7.5.2.message 流**

### **7.5.3. 操作流**

### **7.5.4. 共享流**

### **7.5.5.message 流添加 Messages**

### **7.5.6.MessagesService 代码**

### **7.5.7. 尝试一下**

## **7.6. ThreadsService**

### **7.6.1.Threads 集合**

## **7.7. 数据模型小结**

## 8. 数据结构之观察者 2 - View

8.1. **ChatApp** 顶层组件

8.2. **ChatThreads** 组件

8.3. 单一 **ChatThread** 组件

8.4. **ChatWindow** 组件

8.5. **ChatMessage** 组件

8.6. **ChatNavBar** 组件

8.7. 小结

8.8. 下一步

## **9. HTTP 请求**

### **9.1. 简介**

### **9.2. 使用 angular2 http**

#### **9.2.1. 导入 http**

### **9.3. 简单请求**

#### **9.3.1. SimpleHTTPComponent 添加注解**

#### **9.3.2. SimpleHTTPComponent 模板**

#### **9.3.3. SimpleHTTPComponent 控制器**

#### **9.3.4. SimpleHTTPComponent 代码**

### **9.4. 实现 YouTubeSearchComponent**

#### **9.4.1. SearchResult**

#### **9.4.2. YouTubeService**

#### **9.4.3. SearchBox**

#### **9.4.4. SearchResultComponent**

## **9.4.5.YouTubSearchComponent**

### **9.5. API**

#### **9.5.1.POST 请求**

#### **9.5.2.PUT / PATCH/ DELETE/ HEAD**

#### **9.5.3.配置项**

#### **9.5.4.小结**

## 10. 路由

### 10.1. 为什么路由

#### 10.1.1. 初级：链接标记

#### 10.1.2. 进阶：html5 客户端路由

### 10.2. 客户端路由原理

### 10.3. 第一个路由例子

### 10.4. 路由的组成部分

#### 10.4.1. 路由配置 RouterConfig

#### 10.4.2. 路由出口 RouterOutlet

#### 10.4.3. 路由链接 RouterLink

### 10.5. 组合在一起

### 10.6. 路由策略

### 10.7. 路径策略

### 10.8. 结合应用运行

## **10.9. 路由参数**

## **10.10. 音乐搜索 App**

## **10.11. 路由生命周期 hook**

## **10.12. 嵌套路由**

### **10.12.1. 配置路由**

### **10.12.2. ProductsComponent**

## **10.13. 小结**

# **11. 高级组件**

## **11.1. 样式**

### **11.1.1. 视图(样式)封装**

### **11.1.2. Shadow DOM 封装**

### **11.1.3. 不封装**

## **11.2. 创建 Popup –**

### **11.2.1. 弹出组件构成**

### **11.2.2. 使用 ElementRef**

### **11.2.3. 绑定主体**

### **11.2.4. 使用 exportsAs 增加按钮**

## **11.3. 使用 Transclusion 构建消息嵌套**

### **11.3.1. 更改主体的 CSS**

### **11.3.2. 使用 ng-content**

## **11.4. 相邻 Directive 通信 – Tabs**

#### **11.4.1. Tab 组件**

#### **11.4.2. Tabset 组件**

#### **11.4.3. 使用 Tabset**

### **11.5. 生命周期添加 hook**

#### **11.5.1. Oninit 和 OnDestroy**

#### **11.5.2. OnChanges**

#### **11.5.3. DoCheck**

#### **11.5.4. AfterContentInit,**

**AfterViewInit,AfterContentChecked,AfterViewChecked**

### **11.6. 高级模板**

#### **11.6.1. 重写 ngif – ngBookIf**

#### **11.6.2. 重写 nfFor – ngBookRepeat**

### **11.7. 脏值检查**

#### **11.7.1. 自定义脏值检查**



### **11.7.2. Zone JS**

### **11.7.3. 观察与推送**

## **11.8. 小结**

## **12. Angular 1 如何升级**

### **12.1. 外围理念**

### **12.2. 怎么生成 app**

### **12.3. ng1 VS ng2**

### **12.4. 通用特性知识要求**

### **12.5. Angular 1 App**

### **12.6. 生成一个 Hybrid**

### **12.7. 小结**

### **12.8. 引用**

## **13. 测试**

### **13.1. 测试驱动**

### **13.2. 端到端测试 vs 单元测试**

### **13.3. 测试工具**

#### **13.3.1. Jasmine**

#### **13.3.2. Karma**

### **13.4. 写单元测试**

### **13.5. ng2 单元测试框架**

### **13.6. 配置测试环境**

### **13.7. Services 测试 及 HTTP**

#### **13.7.1. HTTP 思考**

#### **13.7.2. Stubs**

#### **13.7.3. Mocks**

#### **13.7.4. HTTP MockBackend**

### **13.7.5. BeforeEachProviders 钩子**

### **13.7.6. 测试跟踪**

## **13.8. 组件测试**

### **13.8.1. mock 依赖**

### **13.8.2. mock 路由类**

## **13.9. 表单测试**

## **13.10.HTTP 测试**

## **13.11.小结**

## 14. 修改记录

略