

[首页 \(/\)](#) [AngularJS 教程 \(/tutorial/\)](#) [AngularJS PhoneCat \(/phonecat/\)](#) [AngularJS 下载 \(/download/\)](#)

[AngularJS api \(/api/\)](#) [Ecs服务器 \(https://www.aliyun.com/product/ecs?userCode=iuvvbh9n\)](https://www.aliyun.com/product/ecs?userCode=iuvvbh9n)

[简介 \(Introduction\) \(/tutorial/1.html\)](#)
[概念概述\(Conceptual Overview\) \(/tutorial/18.html\)](#)
[引导程序 \(Bootstrap\) \(/tutorial/16.html\)](#)
[Html编译 \(HTML Compiler\) \(/tutorial/15.html\)](#)
[数据绑定 \(Data Binding\) \(/tutorial/10.html\)](#)
[控制器\(Controllers\) \(/tutorial/2.html\)](#)
[服务 \(Services\) \(/tutorial/19.html\)](#)
[作用域\(Scope\) \(/tutorial/12.html\)](#)
[依赖注入\(Dependency Injection\) \(/tutorial/17.html\)](#)
[模板 \(Templates\) \(/tutorial/13.html\)](#)
[使用css\(Working With CSS\) \(/tutorial/11.html\)](#)
[过滤器 \(Filters\) \(/tutorial/8.html\)](#)
[表单\(Forms\) \(/tutorial/4.html\)](#)
[指令\(Directives\) \(/tutorial/5.html\)](#)
[Components \(/tutorial/20.html\)](#)
[Component Router \(/tutorial/21.html\)](#)
[动画\(Animations\) \(/tutorial/7.html\)](#)
[模块\(Modules\) \(/tutorial/6.html\)](#)
[表达式\(Expressions\) \(/tutorial/3.html\)](#)
[供应者 \(Providers\) \(/tutorial/9.html\)](#)
[\\$location \(/tutorial/14.html\)](#)
[单元测试 \(/tutorial/22.html\)](#)
[端对端测试 \(/tutorial/23.html\)](#)



AngularJS 引导程序（Bootstrap）

VPN接続とい
えばスイカVPN

目
次

这一章讲述 Angular 初始化过程以及必要的时候用户如何能够手动将 Angular 初始化。

VPN接続といえばスイカVPN

専用ソフトのインストールは必要ありません。まずは
無料お試しを。

スイカVPN

Angular <script> 标签

下面例子展示了推荐的在应用中使用 Angular 的自动初始化的代码路径设置。

0元开户 0佣金 新手上路送大礼

广告 易上手 涨跌皆可投 分析师在线互

鑫圣金业

打开

```
1. <!doctype html>
2. <html xmlns:ng="http://angularjs.org" ng-app>
3.   <body>
4.     ...
5.     <script src="angular.js">
6.   </body>
7. </html>
```

- 在HTML页面底部放置 `script` 标签。这样可以优化应用的加载时间，因为避免了HTML加载时被 `angular.js` 脚步的加载阻滞。你可以在这里 (<http://code.angularjs.org>)<http://code.angularjs.org> (<http://code.angularjs.org>) 获取 Angular 的最新版本。注意，别再你的产品代码中连接这个 URL，因为这样会在你的网站中暴露一个安全问题。当然，在开发的时候连接是没有问题的。
 - 选择 `angular-[version].js` 这样一个可读的版本，用来作为开发和调试用
 - 选择 `angular-[version].min.js` 这样一个压缩版本，用来作为产品发布时用
- 如果你想要你的应用自动启动 Angular 的话，那就把 `ng-app` 放在应用的根结点中，通常情况下是 `<html>` 标签中，像下面这样：

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

- 如果你的应用需要支持 IE7，那么加上 `id="ng-app"`

```
<html ng-app id="ng-app">
```

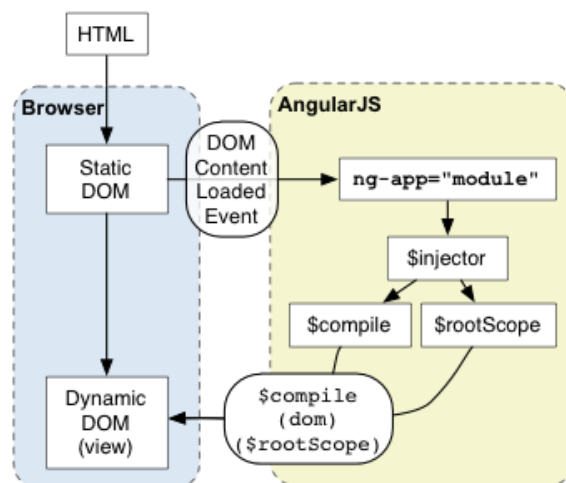
- 如果你要使用 `ng:` 这样的老风格的指令方式，那就在 `html` 中加入 `xml命名空间`，这样也能让 IE 得瑟下。（译注：历史原因，你懂的，所以 `ng:` 这种方式现在已经不推荐了。

```
<html xmlns:ng="http://angularjs.org">
```

自动初始化

Angular 在以下两种情况下自动初始化，一个是在 `DOMContentLoaded` 事件触发时，或者在 `angular.js` 脚本被执行的同时如果 `document.readyState` 被置为 `'complete'` 的话。初始化时，Angular 会去找 `ng-app` (`api/ng.directive:ngApp`) 这个指明应用开始所在的指令。如果 `ng-app` (`api/ng.directive:ngApp`) 指令被找到的话，Angular 会做以下几件事：

- 加载 `ng-app` 指令所指定的 模块 (`guide/module`)
- 创建应用所需的 `injector` (`api/AUTO.$injector`)
- 以 `ng-app` (`api/ng.directive:ngApp`) 所在的节点为根节点，开始遍历并编译DOM树（`ng-app` 指出了应用的哪一部份开始时



Angular 去编译的)

```
1. <!doctype html>
2. <html ng-app="optionalModuleName">
3.   <body>
4.     I can add: 3.
5.     <script src="angular.js"></script>
6.   </body>
7. </html>
```

手动初始化

如果你想在初始化阶段拥有更多的控制权，你可以使用手动方法启动应用。你需要手动启动的可能会是你想要在你的应用中使用脚本加载器，或者你可能想要在 Angular 编译页面之前执行一些别的操作。

下面是一个手动初始化 Angular 的例子：

```
1. <!doctype html>
2. <html xmlns:ng="http://angularjs.org">
3.   <body>
4.     Hello World!
5.     <script src="http://code.angularjs.org/angular.js"></script>
6.     <script>
7.       angular.element(document).ready(function() {
8.         angular.module('myApp', []);
9.         angular.bootstrap(document, ['myApp']);
10.      });
11.    </script>
12.  </body>
13. </html>
```

注意，在上面的例子中，我们提供了我们应用要加载的模块名作为 `api/angular.bootstrap` (`api/angular.bootstrap`) 函数的第二个参数。需要注意的是 `angular.bootstrap` 不会凭空创建模块，在我们将模块作为参数注入之前，必须创建任一自定义的 模块 (guide/module)。

以下是你的 Angular 代码运行时遵循的顺序：

1. 在HTML页面以及所有代码加载完毕后，Angular 会去找到应用的根元素（通常是文档的根节点）
2. 调用 `api/angular.bootstrap` (`api/angular.bootstrap`) 去 编译 (guide/compiler) 各元素成为一个可执行的且双向绑定的应用

延迟启动

这个特色可以让像 Batarang 一样的测试工具横插一杠进入 Angular 的引导进程，并且溜进模块中的DI注册机制中，这样就可以替换或者增强DI提供的服务。（译注：由于后面一句不甚理解其原理，不敢贸然翻译）

当 `api/angular.bootstrap` (`api/angular.bootstrap`) 被调用时，如果 `window.name` 包含 `NG_DEFER_BOOTSTRAP!` 前缀，引导进程会被暂停直到 `angular.resumeBootstrap()` 被调用。

`angular.resumeBootstrap()` 以一个可选的数组作为参数。这个数组是包含了应用启动时需要被注入的模块。



Copyright © 2014 - 2018 AngularJS中文网 (<http://www.angularjs.net.cn>), 粤ICP备15074038号 (<http://beian.miit.gov.cn>), All Rights Reserved. EasyUI中文网 (<http://www.jeasyui.net>)