核心概念 编辑页面

英文原文：http://emberjs.com/guides/concepts/core-concepts/

要开始学习Ember.js,首先要了解一些核心概念。

Ember.js的设计目标是能帮助广大开发者构建能与本地应用相颉美的大型Web应用。要实现这个目标需要新的工具和新的概念。我们花了很大的功夫从Cocoa、Smalltalk等本地应用框架引入了其优秀的理念。

然而，记住Web的特殊性非常重要。很多人认为一个应用是Web应用是因为其使用了像HTML、CSS和Javascript这些技术。实际上，这只是实现的细节问题。

相反，Web应用是通过能收藏和分享链接来凸显它的作用的。URL是Web应用的一个最核心的特性，正是URL使得Web应用有了卓越的可共享性和可协作性。现今，很多Javascript框架都事后才考虑URL，没有考虑这个让Web成功的主要因素。

Ember.js将本地GUI框架中的工具和概念与使得Web应用如此强大的URL嫁接在一起。

概念

模板

模板，用Handlebars模板语言来编写，它描述了一个应用程序的用户接口。每个模板背后都有一个模型，当模型发生改变时，模板将自动进行更新。

此外，相对于纯HTML，模板还提供了：

- 表达式，例如 `{{firstName}}` ，它从模板对应的模型获取信息并将信息添加到HTML中。
- 出口（Outlets），它是其他模板的占位符。当用户使用应用时，不 同的模板会通过路由插入到出口中。你可以使用 `{{outlet}}` 助手将出口放到 模板中去。
- 组件，自定义的HTML元素，可以用来清理重复的模板或创建可重用的控件。

路由器

路由器将URL转换为一系列内嵌的有模型数据支撑的模板。当显示给用户的模板和模型发生改变时，Ember自动更新浏览器地址栏中的URL。

这意味着用户可以在任意点分享应用的URL。当某个用户点击了这个链接时，将看到与分享链接的用户看到的相同内容。

组件

组件是一个自定义的HTML标签，其行为用Javascript来实现，而显示使用Handlebars模板来描述。组件可以用来定义可重用的控件来简化应用的模板。

模型

模型是一个存储 持久化状态 的对象。它是应用将操作的数据，也是用来返回值给用户的数据。这些对象通常从服务器端加载，并当其在客户端发生改变后又保存到服务器端。

路由

路由是 负责管理应用程序状态 的对象。

控制器

控制器是存放应用状态的对象。模板除模型之外还可以有一个控制器与之对应，使其可以从这两者获取属性。

以上这些是在开发Ember.js应用时需要了解的核心概念。Ember.js设计为可以弹性的处理复杂的问题，因此需要为应用增加新功能、新特性时只需要改变很小的部分。

现在你已经理解了这些对象各自的角色，可以开始深入到Ember.js 的世界中，进一步了解这些部分如何工作的细节。

← 获取Ember: 获取Ember 命名惯例 →