**我要如何完成此项目？**

1. 复习之前的 JavaScript 设计模式课程，并仔细阅读**[街区地图项目评估指南](https://review.udacity.com/" \l "!/rubrics/501/view" \t "_blank)**.
2. 下载 [**Knockout 框架**](http://knockoutjs.com/)。

<script type='text/javascript' src='knockout-3.4.2.js'></script>

* 必须**使用** Knockout：处理列表、筛选器和状态会不断更改的页面的任何其他相关信息。
* **不应**通过 knockout 处理的工作：地图 api 适用的任何工作、 创建标记、跟踪其点击事件、创建地图、刷新地图。

1. 错误处理：

如果出错（例如，在第三方 api 未返回预期结果时），我们预计你的网页会执行以下操作之一：

* 显示一则**消息**，通知用户数据无法加载，
* **或者**不会对 UI 产生负面影响。

**注意：** 请注意，如果浏览器在最初**访问第三方站点**时遇到问题，我们也期望学生处理错误。

* 例如，假设用户使用你的区域地图，但她的防火墙阻止她访问 Instagram 服务器。这里是有关使用 hosts 文件 [**如何阻止网站**](http://www.digitaltrends.com/computing/how-to-block-a-website/) 的参考文章。

务必处理错误，以使用户始终在网页中获得良好的体验。请阅读 [**此博文**](http://ruben.verborgh.org/blog/2012/12/31/asynchronous-error-handling-in-javascript/) 以了解更多信息。

某些 JavaScript **库**提供了处理错误的特殊方法。

* 例如：如果你使用 jQuery 的 ajax() 方法，请参考 [**此处**](http://api.jquery.com/jquery.ajax/#jqXHR) 讨论的 .fail() 方法。我们强烈鼓励你在创建 API 调用时使用的库中探索处理错误的方法。

1. 编写显示地图**标记**（标识你感兴趣的至少 **5 个周边地点**）所需的代码。

加载页面时，你的应用应该会**默认显示**这些地点。

1. 如果系统提示你这样做，你可能需要获取 [**Google Maps API key**](https://developers.google.com/maps/documentation/javascript/get-api-key)，并将其添加在**index.html**中，替换下面的[YOUR\_API\_KEY]： <script src="http://maps.googleapis.com/maps/api/js?libraries=places&key=[YOUR\_API\_KEY]"></script>
2. 使用 [**Google 地图 API**](https://developers.google.com/maps/) 编写在页面上添加全屏地图所需的代码。出于效率原因，地图 API 仅应**调用一次**。
3. 实现在步骤 5 中定义的地点集的**列表视图**
4. 提供一个**筛选器**选项，使用输入字段来筛选加载时，默认显示的**列表视图**和地图**标记**。

列表视图和标记应该**实时更新**。通过第三方 API 提供搜索函数不足以符合规格要求。

1. 使用**第三方** API 添加功能，以在单击地图标记或列表视图条目时提供信息

* 例如，Yelp 评论、Wikipedia、Flickr 图片等。

请注意，StreetView 和 Places 并不算作其他第三方 API，因为它们是包括在 Google 地图 API 中的库。

如果你需要复习如何对**第三方服务器**发出 AJAX 请求，请查看我们的 AJAX 简介课程。

1. 添加功能，以在**选定**与地图标记关联的**列表**项目或地图**标记**本身时，将其**动画化**。
2. 添加功能，以在选定列表视图中的地点或直接选定其地图标记时，打开包含步骤 9 中介绍的信息的**信息窗口**。
3. 应用的界面应该直观易用。

* 例如，筛选地点的输入文本区域应该易于定位。
* 应该能够轻松了解要筛选的地点集。

通过列表项目或地图标记选择地点应该会导致地图标记退回或以某种其他方式显示动画，以指明该地点**已选定**，并且应该在地图标记上方**打开**包含其他信息的关联**信息窗口**。

* 这个应用在移动端上应该是**响应式**的：所有应用组件均具有响应性，均可在桌面、平板电脑和手机浏览器上使用

|  |  |
| --- | --- |
| 地点筛选 | 设置一个**文本输入栏**或**下拉菜单**，使其可以   * **筛选**地图标记 * 列出与输入文本或选项相**匹配**的列表项。 |
| 列表视图 | 设有地点名称的列表视图，默认显示所有地点，在添加筛选条件后显示筛选后的地点子集。  点击列表上的地点可显示该地点的相关信息，同时，其相应的地图标记会有所反映（如上下跳动或颜色变化）。  列表功能具有响应性，运行无差错。 |
| 地图和标记 | 地图默认显示所有地点标记，在添加筛选条件后显示筛选后的地点子集。  点击一个标记，在 infoWindow 或 DOM 元素中显示该地点的信息。  标记应在点击时有所反映（如上下跳动或颜色变化）。  应用中的所有其他功能运行无误。 |

|  |  |
| --- | --- |
| 异步数据请求 | 应用使用 Google Maps API 和至少一个**非谷歌的第三方 API**，详情参见[此文件](https://developers.google.com/maps/documentation/javascript/tutorial)。  所有数据要求均以**异步**形式取回。 |
| 错误处理 | 出错的数据要求均应利用常见的应变技术（即 AJAX 错误或错误方法）进行妥善处理。“妥善”意味着用户不会对不运行的组件感到困惑。  如果 API 无法加载，页面上需要有明显的指示（**警告框**即可），说明其无法加载。你无需处理用户下线时的问题。 |

|  |  |
| --- | --- |
| 更多地点数据 | 设有来自第三方的功能，以提供有关某地点的其他数据。该信息可以在标志的 infoWindow 或 DOM （侧栏或列表视图等）的 HTML 元素中显示。  为其他数据源提供属性。例如，如果使用 Foursquare，在 UI 或 README 文件中说明你在使用 Foursquare 数据。 |

|  |  |
| --- | --- |
| 注释 | 设有有效解释较长代码程序的注释。 |
| README 文档 | README 文档含有成功运行应用所需的所有详细步骤。 |
| 代码质量 | 代码格式应保持前后一致，代码应易读、具有逻辑性并符合[优达学城 JavaScript 样式指南](http://udacity.github.io/frontend-nanodegree-styleguide/javascript.html)的规定。  若使用了构建工具（如 Gulp 或 Grunt ），须将源代码和生产代码通过不同的目录提交在同一储存库中，这两个目录通常被命名为 src 和 dist。 |

