细节：（可以考虑插入到原有的文章中，这样比较合理哈）

Return View(“NotIndex”);命名视图

Return View(“~/Views/Example/Index.cshtml”);

简单请款下推荐使用Dynamic的ViewBag，复杂推荐强类型的ViewModel，在使用强类型时，为了方便，推荐将其加入配置文件中，如下所示。

|  |
| --- |
| <pages pageBaseType="System.Web.Mvc.WebViewPage">  <namespaces>  <add namespace="Sory.Entertainment.WebAPI.Models"/>  </namespaces>  </pages> |

此外，比较有意思的是，可以通过Install-Package XXX命令安装自己的程序集。

视图模型部分推荐大家使用MVVM模式，尽可能的使数据扁平化，便于强类型的数据管理。

在（设定好Model后）通过基架模板Scaffold template（T4模板）新建视图时，支持Empty、Create、Delete、Details、Edit、List等类型的视图。通过设置RefenceScriptLib添加JQuery（包括非侵入版）验证等相关js。

Razor引擎简单介绍：

通过“@”进行代码和标记的转换符，@@表示原有的@，@（表达式）

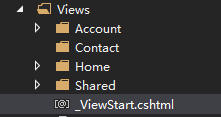
|  |  |
| --- | --- |
| 表达式 | 值（只是示例，勿深究） |
| @ProductList.Length | 3 |
| @@ | @，转义 |
| Item\_@(item.length) | Item\_3 |
| @{ xxx } | 代码块 |
| @{ Html.RenderPartial(“SomePartial”); } | 代码行 |
| @foreach(var item in items){<span>Item @item.Name</span>} | 循环 |
| @if(showMessage){<text>xxx<text>} | 判断 |
| @\* ………………..\*@ | **注释**，这个需要注意哦，不然可能会出错 |
| @(Html.SomeMethod<Type>) | 泛型需要加括号 |

所有后台数据的呈现默认使用HTML编码（这个html标记类型为System.Web.IHtmlString），如果遇到特殊字符，如<script>alert(‘hacked!’)<alert>等情况，会编码为&lt;script&gt;….，如果需要原样输出，[需要使用@Html.Raw(message)](mailto:需要使用@Html.Raw(message))。这样就够了么？不然，在前端如果使用js处理后台数据时，为了防止跨站脚本攻击，还需要使用’@Ajax.JavaScriptStringEncode()’防止XSS攻击。

布局，使用在Layout页面中使用@RenderBody可以为主体占位，也可以用的@RenderSection(“Footer”)为Footer占位，这时需要在页面中通过@section Footer{ html内容 }设置，否则会抛出异常，一个比较合理的例子如下。

|  |
| --- |
| <footer>  @if(IsSectionDefined(“Footer”)){RenderSection(“Footer”);}  Else{<span>Default Footer</span>}  </footer> |

此外，还可以使用\_ViewStart.cshtml页面设置布局页，它可以递归到相应的子目录中，它先于所有页面的渲染。



最后一点是，可以通过部分视图的方式，来满足Ajax调用的需要，这个需要时html文件，而仅仅是json要注意。

|  |
| --- |
| <script>  $(function()){  $(‘#partialView’).Load(‘/home/partial’);  });  </ script > |