# TES总结

## 布局模板页Layout.cshtml

* 1. 我/你是如何避免前端代码重复的？我/你是如何设计前端的？

在项目中，大多数页面都具有相同的布局，样式，如果简单的复制粘贴，那么当需要更改布局样式时，改动是空前的，这会给项目后期维护带来很多的困难，时间与经济对于企业是非常宝贵的，在ASP.NET MVC中提供了模板页，通过将反复出现需要重用的公共页面提取为布局模板页，再通过在razor视图页更改Layout属性来使用布局模板页，这样就重用了视图，只需更改布局模板页，就可以将布局样式更改，有效的减少了后期维护的困难，并且布局模板页是支持反复嵌套的，一个视图使用的布局页又可以再次标记使用另一个布局页，然而这在过去WebForms所支持的ASPX视图引擎的母版页所不支持的，然而，这还不算什么，在razor中提供了部分视图Html.Partical()和Html.RenderPartical(),出于性能考虑我选择了后者，因为它直接将页面内容输出到HttpContext上下文，而Html.Partical()会将内容编码为MvcHtmlString，性能上有所损耗，尽管它使用起来简单一点，毕竟他不需要花括号，通过Html.RenderPartical()，我将经常常见的导航栏，菜单，消息框等控件封装，实现了前端模块化，这使得代码结构清晰，提高了易读性，重用率，