1. 缓存

EMR系统的缓存机制，主要是基于Redis和MemoryCache。

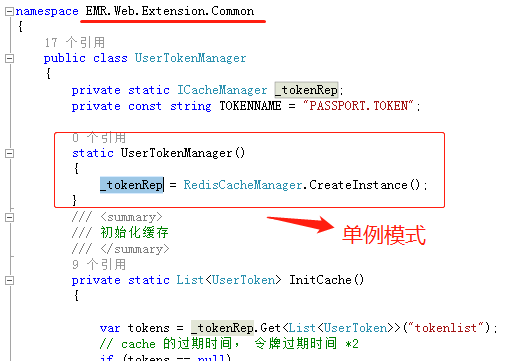
Redis是一个第三方的缓存插件，它不需要安装，但需要但在使用的时候开启Redis服务。

MemoryCache是.Net自带的缓存插件，在命名空间System.Runtime.Caching下。

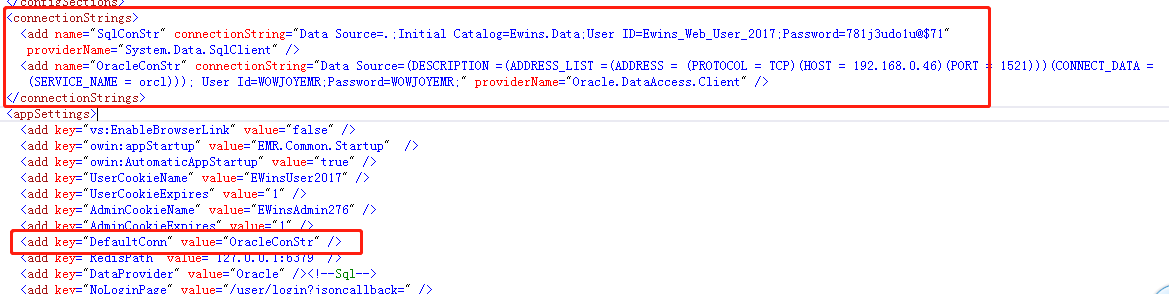
Redis比MemoryCache要更加稳定，更加高效，存储的模式更加丰富。

注意事项：

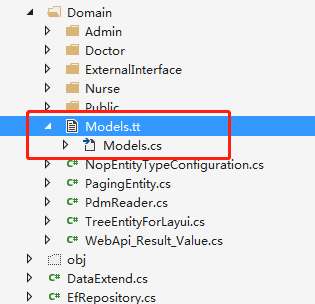
1. 两个缓存之间的切换，需要重启web站点，否则会报错。
2. 前端操作UserToken的时候，由于UserToken的操作非常频繁，所以只能以单例模式创建缓存对象，不能更换缓存机制，否则会让前端访问变慢。



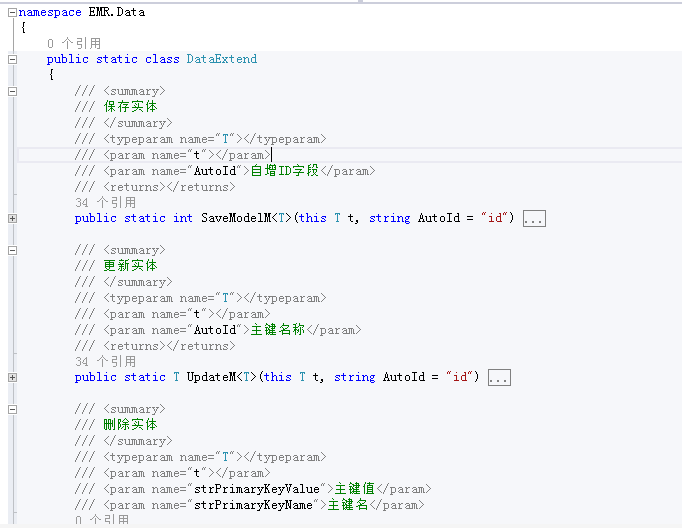
1. 数据操作
2. EMR通过接口IDataProvider及Web.Config来动态配置数据的驱动模式（Oracle/SqlServer）。然后通过EntityOperate 这个类进行数据操作。



1. 实体通过T4模板直接从PDM生成。如果需要扩展实体属性，请在EMR.Data.Domain文件夹下进行相应的扩展。



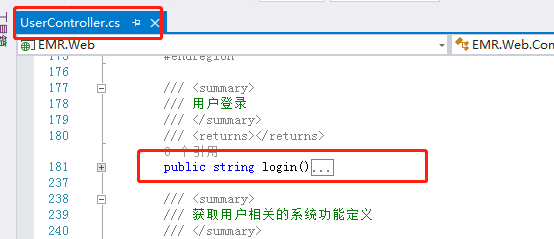
3、单个实体可以直接进行保存（entity.SaveModelM()）、更新（entity.UpdateM()）、删除（entity.DeleteM()）等操作。这些扩展方法都在EMR.Data.DataExtend类中实现：



1. 前端认证机制

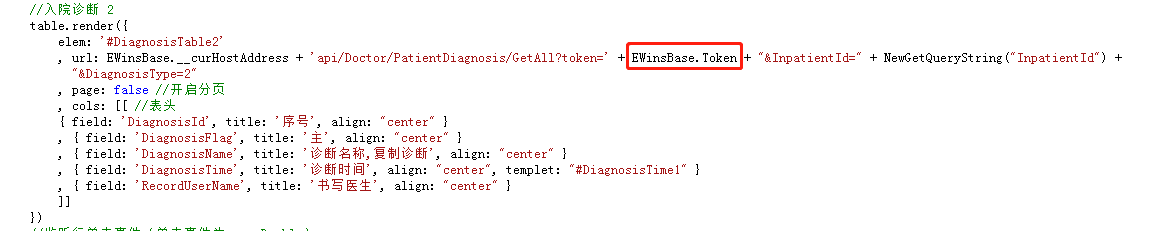
每个用户登录后，都会生成一个专属的TOKEN密钥进行保存。后端保存在缓存中，前端保存在cookie中。

后端用户数据的保存方式可以具体查看：EMR.Web.Controllers.SystemSupport.UserController 中的login方法。



前端每次访问后端数据，都要传入TOKEN进行验证。

前端只能通过AJAX访问接口，获取数据。EMR系统的前端使用AJAX访问后端的方法有多种：通过layui进行访问、通过jquery进行访问、通过EWinsBase进行访问。当然，若有必要，也可以通过你自己封装的JS进行访问。其中，EWinsBase是一个经过封装的对象体，里面封装了很多方法和属性，经常使用的TOKEN就封装在里面，当我们使用EWinsBase访问后端接口的时候，EWinsBase会自动传递TOKEN给后端，无需我们费心。但如果使用layui或者jquery访问后端接口，则需要传入TOKEN：



[page/DoctorPages/Medrecord?EMRIDSIGN=30&InpatientId=1](http://localhost:4773/page/DoctorPages/Medrecord?EMRIDSIGN=30&InpatientId=1)

1. 路由机制

EMR的路由机制比较特殊。它采用的是页面、接口分离机制，重写了MVC的底层方法。

页面方法全部放在EMR.Web.Controllers.Page文件夹下。

接口方法全部放在EMR.Web.Controllers文件夹下。

详细路由机制说明，请参考类：EMR.Web.RouteConfig。里面有做非常详细的解说。