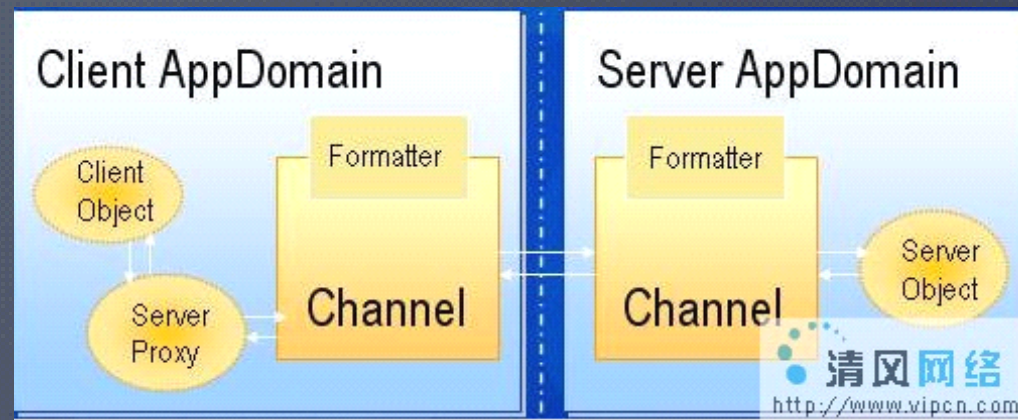




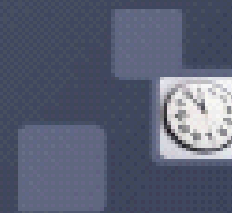
CS框架专题

远程处理

基本原理



程序需求



- .Net framework 4.0及以上框架
- 客户端必须是Winform或wpf程序，不支持控制台程序。
- 客户端所有的窗体必须从cs框架中继承

演示程序



- 见RemoteClient和RemoteServer演示

服务器端启动流程



- 1 检查工作目录，查找配置文件
- 2 启动应用程序域(对应用程序域简单说明)
- 3 搜寻配置中对外提供服务的端口（如果未设置）
- 4 加载工作目录下所有的.net 动态库，搜寻所有对外提供服务的RemoteAction并对其进行注册。
- 运行服务器端初始化方法
"InitRemoteServer"方法。

服务器端介绍



- 服务器启动新服务时会启动一个新的应用程序域，应用程序域是.net程序的一个安全边界，类似于windows系统下的进程
- 服务器在启动新服务时会给该服务分配配置文件(即设置服务的.config文件)，一个服务的工作目录下面只能最多有一个.config文件，该文件会被设置为服务的配置文件。

服务器端介绍



- 服务器端配置完成后会查询该服务的程序中是否含有一个静态的InitRemoteServer方法，如果有则会调用该方法，可以将服务端初始化代码放入该方法。
- 服务器运行方式可能是服务，因此可能没有windows消息机制，不能显示窗体。

客户端介绍



- **RemoteAction**类介绍(说明该类的代码需要在服务端运行)
- **ClientAction**类介绍（说明该类的代码需要在客户端运行,可进行双向调用，暂不演示）
- **RemoteObjectAttrite**类介绍（指定被标记的类是否在服务端运行，以及运行的方式，只有标记在**RemoteAction**类及其派生类时才起作用。

客户端介绍



- 所有的初始化操作在第一个BaseForm实例化过程中进行（及如果不继承BaseForm则所有的远程对象都无法在远程运行）

远程处理的3种类型



- **1 PerInstance** 客户端创建一个实例，服务端创建一个实例（不推荐，难维护状态）
- **2 SingleCall** 客户端每次请求，服务端新建一个实例处理客户端请求，处理完成后该类被销毁（性能稍差，稳定安全,网站，推荐）
- **3 Singleton** 服务器端只有一个实例处理所有的客户端请求（性能最好，但需注意死锁等问题）

演示程序2



见RemoteClient2

应用类介绍



- **Util.ClientOnly** 指示当前所有业务逻辑代码是否只在客户端运行（通过 **Boyoi.Client.Config.ClientConfig.ClientOnly** 属性设置业务逻辑是否只在客户端运行。）
- **Boyoi.Utils.Util.IsServerSide** 指示当前程序是否运行在服务端，可以通过该属性分别设置服务端和客户端运行的代码。

注意事项



- 服务端运行时多线程环境，注意线程安全
- 服务器上存储数据时注意线程安全。
- 注意网络故障，网络故障时，所有调用方法都会抛出异常。
- 有极小的几率在服务器向客户端返回数据时出现网络故障，此时服务器操作已经完成，注意代码兼容性。（比如数据入库）
- 不适合在internet上大量使用（上千客户端）

ASP.NET MVC



- 网站本质
- MVC
- 简单的参数传递
- 示例

网站本质



- 网页的本质是遵循HTML规则的字符串
- 网站本质是处理HTTP请求，返回符合HTML规则的字符串，即客户端向服务端TCP端口发送HTTP请求，服务端处理请求后通过TCP返回HTML格式的字符串
- .Net简化处理过程

MVC介绍



- **M**:模型， 包含数据模型和业务逻辑
- **V**:显示， 包含界面展示的代码
- **C**:控制器， 根据情况调用模型和显示。

ASP.NET MVC



- 服务器受到客户端的请求，则会实例化一个Control类(mvc 中的 c,即控制器)
- 控制器调用对应的业务逻辑(mvc中的m)和对应的展示界面(mvc中的v)并且将业务逻辑的结果传递到展示界面中，生成含有业务的界面。
- 控制器将生成的字符串发送到客户端(暂时这样理解)。

参数传递



- `http://localhost:2865/mycontrol/MyAction?input=my%20input`
- url中对应的mycontrol即调用服务器端的MyController
- url 中对应的MyAction表示调用MyController类对应的MyAction方法（该方法成为Action）
- url 中对应的 `input=my%20input` 即调用方法时给参数input传递值为 `my%20input`

实例

- 见MVCDEMO

MVC总结



- 控制器(C)处理请求，调用正确的业务逻辑和对应的界面。
- 模型(M) 负责业务逻辑代码
- 界面(V) 根据控制器传递来的数据生成界面布局。