**讲稿**

**为什么要使用identityserver3框架？**

当然可以基于OAUTH2写一套SSO服务器，但是Identityserver3是一直致力于研究验证框架和服务的，他们是专业的，在安全性上考虑会更加全面。

也比较适用于我们这样的小项目小团队，只关注逻辑处理。

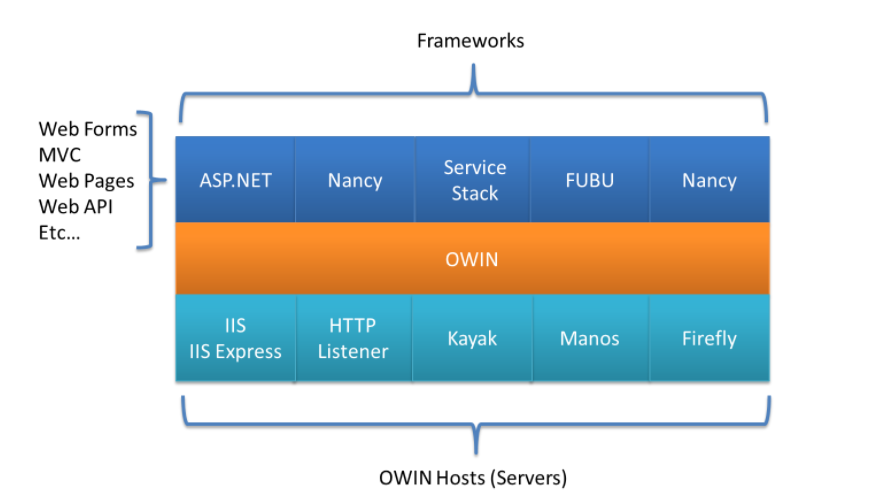
**需要了解的几个概念**

**- Owin**

Owin（open web interface for .net）只是标准，目的是为了解耦web服务器和应用程序。

ASP.NET对IIS的依赖性太强。

Katana是Microsoft对OWIN的实现。



深入：Owin的5个层都分别表示什么，怎么扩展

*参考文档：*

*http://www.cnblogs.com/dudu/p/what-is-owin.html  --owin是什么*

*http://www.cnblogs.com/jesse2013/p/owin-webserver.html --owin和Web服务器*

*http://www.cnblogs.com/jesse2013/p/aspnet-identity-claims-based-authentication-and-owin.html --ClaimBased认证和owin*

**- ClaimBased认证**

这种认证方式最主要的是把认证和授权分成另外的web服务，我不关心怎么认证授权的，我只需要从认证授权服务器拿到令牌，

这个令牌里面包含了用户的信息（用户名，角色等）。例子：在没有账号的网站上使用QQ账号登录。

思考：怎么从token中解析用户的信息，又怎么获取用户信息。

ASP.NET OWIN OAuth（Microsoft.Owin.Security.OAuth）负责对access token 的解密和处理，赋值给HttpContext.Current.User,写入Cookie这些操作（具体的原理可以深入去了解）

**- OAuth,OAuth2**

OAuth是关于授权的网络开放标准

OAuth2是对这个标准的实现

**- OPenID Connect**

OpenID Connect 是一套基于 OAuth 2.0 协议的轻量认证级规范，

OpenID Connect 方式除了认证请求之外，还标明请求的用户身份。

区别

OpenID Connect 侧重认证：用户是谁

OAuth2 侧重授权：用户能做什么，这并不是说OAuth不能用来认证，他是认证成功的人才能授权

如果A网站需要用B网站的用户系统进行登录（学名好像叫federated login），

它可以选择OpenID认证，然后通过attribute exchange获取用户的昵称或其他通过OpenID暴露出来的用户属性，

或者选择OAuth认证，获取到token后再用token获取用户昵称或其他允许被访问的信息（资源或者scope）

**- Claim,ClaimsIdentity,ClaimsPrincipal**

HttpContext 类:封装有关个别 HTTP 请求的所有 HTTP 特定的信息。保持单个用户、单个请求的数据，并且数据只在该请求期间保持

HttpContext.User 属性:获取或设置当前 HTTP 请求的安全信息。

看下定义：

public IPrincipal User {

         get;

         [SecurityPermissionAttribute(SecurityAction.Demand, ControlPrincipal = true)]

         set;

}

转到定义看下 ClaimsPrincipal,和ClaimsIdentity

多个Claim组成ClaimsIdentity,一个ClaimsPrincipal可以包含多个ClaimsIdentity

就像一个人可以同时拥有护照和驾照，护照和驾照都拥有一组表明用户身份的数据。

Microsoft.Owin.Security.OAuth帮我们做的事就是从token中获取用户的信息，并且赋值给HttpContext.Current.User。

*参考：*

*http://openid.net/connect/*

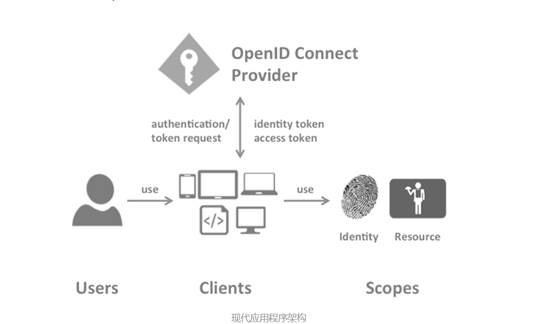
*https://tools.ietf.org/html/rfc6749*

*http://www.ruanyifeng.com/blog/2014/05/oauth\_2\_0.html  阮一峰这篇日志基本诠释了OAuth2，可以说是从官方文档提出来的精华。说明了四种模式的调用方式，和OAuth2解决的问题*

*http://www.cnblogs.com/dudu/p/6367303.html*

*http://www.cnblogs.com/dudu/p/4646922.html*

**identityserver3的概述**



OpenID Connect Provider是负责验证和分发token.

Clients 请求token

Users 具体的用户

Scopes 要访问的资源标识，包括身份作用域（用户的身份信息）和资源作用域（webApi等）

Identity token身份令牌包括用户的身份信息

Access token 访问令牌是拿来访问资源的。

**重点来了IdentityServer3 Demo**

**- Server的实现方式（结合示例代码）**

IdentityServerOption中主要是对一些配置信息：<http://www.jianshu.com/p/b61c598bf311>

重点介绍几个：

**Factory**：IdentityServer3必须的配置信息以及扩展信息

对UserService的扩展实现自定义的登录和用户数据的管理。

对ViewService的扩展可以实现自定义页面展示

**SigningCertificate**：签名证书。

这个属性对于身份令牌和JWT访问令牌是必须设置的。如果用的是参考令牌，可以不设置签名证书。（但是目前项目中使用的是identityserver3的测试证书）

**RequireSsl：**标示IndentityServer是否需要SSL，默认为true.

**AuthenticationOptions：**认证选项

EnablePostSignOutAutoRedirect 配置登出后自动重定向，重定向url是由Client定义的PostLogoutRedirectUris属性决定的。

**EventsOptions**：事件相关设置

IdentityServer在运行时会触发很多事件，例如令牌发放成功等，这些事件默认会发到配置的日志服务

EventsOptions 类包括以下设置(默认全为 false):

 RaiseSuccessEvents

比如:更新令牌刷新了或者授权成功

RaiseFailureEvents

比如:授权失败，授权码赎回失败

RaiseErrorEvents

比如：未处理异常

RaiseInformationEvents

比如：令牌发放和证书验证

**- MVC的接入（结合示例代码）**

假设：所有通过认证的用户都有权访问方法，是否有某个资源的访问权限有代码逻辑来实现。这样我们其实是使identity server只做认证的事情。

**- WebApi的接入（结合示例代码）**

 Api作为Scope资源进行注册

只接受identityserver的令牌

只接受给generalapi的令牌

 可以自己去研究Console或者其他方式接入，APP调用

还有把identityserver3角色和资源权限用起来

*参考：*[*http://identityserver.github.io/Documentation/docsv2/*](http://identityserver.github.io/Documentation/docsv2/)

*http://www.jianshu.com/p/398b226e41ed*