所谓自定义授权管理器，就是从 ServiceAuthorizationManager 类派生出一个类，并重写其 CheckAccessCore 方法，如果检查通过，允许调用者访问服务操作，就返回true，表示通过，如果不通过就返回false。

为什么要重写这个方法呢，你看看.NET源代码就明白了：

protected virtual bool CheckAccessCore(OperationContext operationContext)

{

return true;

}

直接返回true，万能授权。

授权检查是以服务操作为单位的，即Operation，因为服务的每次调用，实际上你只能调用一个操作方法，因为WCF是基于消息的，跟HTTP一样，一问一答（当然也可以只问不答，比如单工模式），故授权是基于操作为单位的，CheckAccessCore方法就一个参数，OperationContext，这个东东是跟当前要调用的操作相关的信息。

下面我们来实现一下。

[IMG_256](https://www.cnblogs.com/tcjiaan/p/javascript:void(0);)

class MyAuthorManager : ServiceAuthorizationManager

{

protected override bool CheckAccessCore(OperationContext operationContext)

{

ServiceSecurityContext sccontext = operationContext.ServiceSecurityContext;

// 检查声明集

foreach (ClaimSet clset in sccontext.AuthorizationContext.ClaimSets)

{

foreach (Claim c in clset.FindClaims("city", Rights.Identity))

{

if (c.Resource.ToString() == "佛山")

{

return true;

}

}

}

return false;

}

}

[IMG_257](https://www.cnblogs.com/tcjiaan/p/javascript:void(0);)

其实这个授权检查会失败的，还记得吗，我们刚刚在自定义授权策略时，在Evaluate方法中，添加的声明集里面，其中有一个type为city，resource为“广州”，但是，你看看此处，我检测的值是“佛山”，所以，这个授权检测会返回false，表示授权不通过，想调用服务，没门！

接着，不要忘了，把刚定义的授权管理器类添加到授权的ServceBehavior中，下面是配置文件写法：

[IMG_258](https://www.cnblogs.com/tcjiaan/p/javascript:void(0);)

<serviceAuthorization **serviceAuthorizationManagerType="Coc.MyAuthorManager, SomeApp771"**>

<authorizationPolicies>

<clear/>

<add policyType="Coc.CustPolicy, SomeApp771"/>

</authorizationPolicies>

</serviceAuthorization>