

Sql 拦截配置及操作说明书

功能说明:

Sql 拦截主要是为了特定租户按需显示内容而所做的功能。主要分为两大部分。一部分为：以配置文件形式来过滤数据，分为按特定表对应的特定字段来过滤。另一部分为：根据租户特定需求，在 sql 配置管理详细页面来配置 sql。

配置说明:

1. TableConfig.xml 配置

TableConfig.xml 文件的配置格式示例如下：

```
<Config>
  <Tables>
    <Table Name="sys_role">
      <Parameters>
        <Parameter Name="sys_data_type" Value="1"></Parameter>
        <Parameter Name="isdefault" Value="1"></Parameter>
      </Parameters>
    </Table>-->
  </Tables>
  <SystemConfigs>
    <Config Name="Ten_Id" Value="1e4e3cc4d09347a18caf95c6005d4d5e"></Config>
  </SystemConfigs>
</Config>
```

Config 底下分为两大节点：

1. Tables

Tables 底下为具体的 table，这些 table 的属性 Name 对应的是数据库中的具体表名

Parameter 为参数：Name 对应的表下的字段，Value 为该字段对应的值。

这部分主要是过滤特定表下的特定字段。配置字段来满足特定的需求。

2. SystemConfigs

SystemConfigs 节点下的 Config 配置，从全局出发，来过滤数据。Name 对应的字段

为数据库中所有表都有的字段，Value 为对应的值。以此来全局过滤数据。

3. 位置：

TableConfig.xml 放在程序的 Config 文件夹下，在不用 TableConfig.xml 来过滤数据时，需要把 Tables 和 SystemConfigs 下的节点删除或是注释。

2. DB.config 配置

DB.config 在原有配置基础上，需要修改两部分。

1. 需要添加鉴权的数据库，这是为了在框架启动时，获取用户的 TokenID 和获取鉴权数据库中的 sql 配置数据。

需要添加：

```
<add name="conn1" connectionString="server=10.95.3.112; port=3306;
uid=ctuni_oauth; pwd=ctuni_oauth; database=ctuni_tenant; Pooling=true;
charset=utf8"
providerName="Tescomm.Framework.DB.MySql.MySqlDataProvider,Tescomm.Framework.DB.MySql"/>
```

为固定写法

2. 在 sql 拦截中，我们用到了 GetCurrentUserInfo 和 GetSqlCommandList 这两个 sql 配置方法，所以需要配置这两个方法所在的 config 文件。进而在 DB.config 的<files>节点下配置添加：

```
<file name="OAuth" src="\MySql\OAuth_db.config"
mappingType="Tescomm.Framework.DB.CommandTemplateCollection,Tescomm.Framework.DB" />
```

```
<file name="OAuthModel" src="\MySql\OAuth_db.config"
mappingType="Tescomm.Framework.DB.ModelTemplateCollection,Tescomm.Framework.DB" />
```

为固定写法

3. OAuth_db.config 配置

在我们框架中，OAuth_db.config 配置，主要服务于 Open 服务，里面配置的 sql 都为读取鉴权数据库的一些数据，业务涉及到登陆用户的信息及权限等等。在 sql 配置拦截中，我们获得 sql 配置数据，也配置到了 OAuth_db.config 这个文件中。

在该文件中，需要添加：

```
<dbCommand id="GetSqlCommandList" db="conn1">

    <text>

        <![CDATA[

            select  * from sys_sql_parameterinfo where app_id={App_Id:String} and
Ten_Id={Ten_Id:String}

        ]]>

    </text>

</dbCommand>
```

为固定写法。

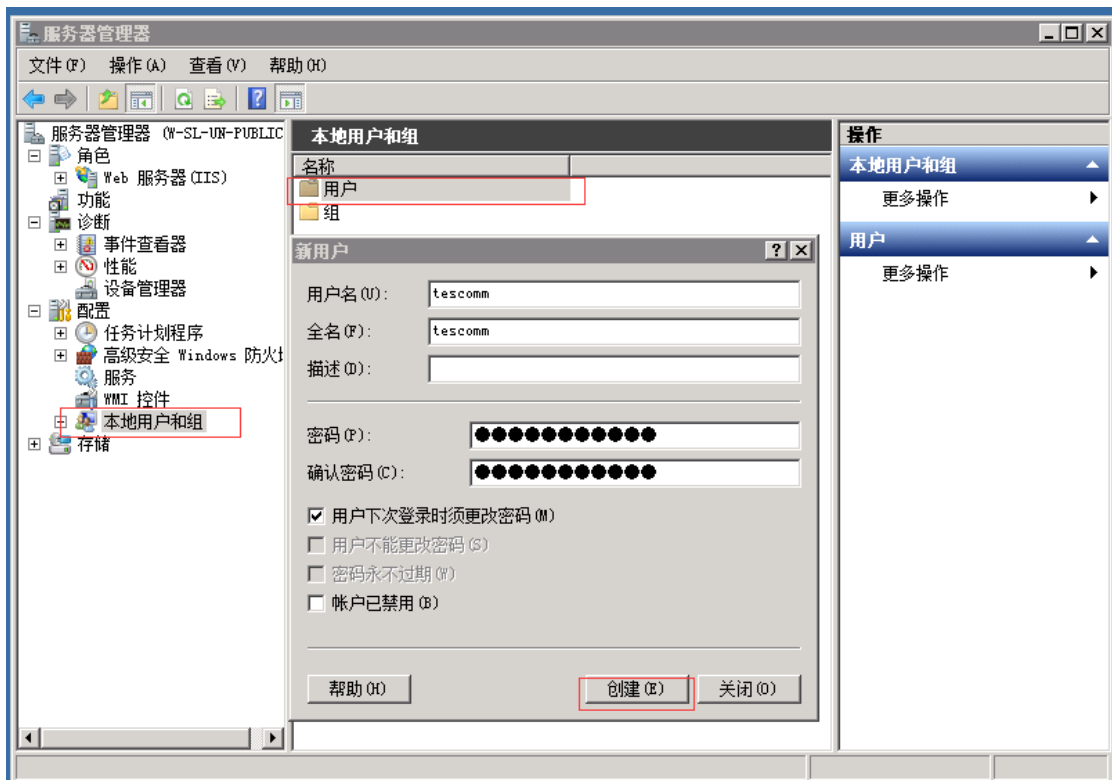
4. 建立 ftp 服务器

在配置文件前，我们需要在一台 windows 服务器上建立一个 ftp 服务，把各个项目需要检测的 sql 配置文件放到 ftp 下。

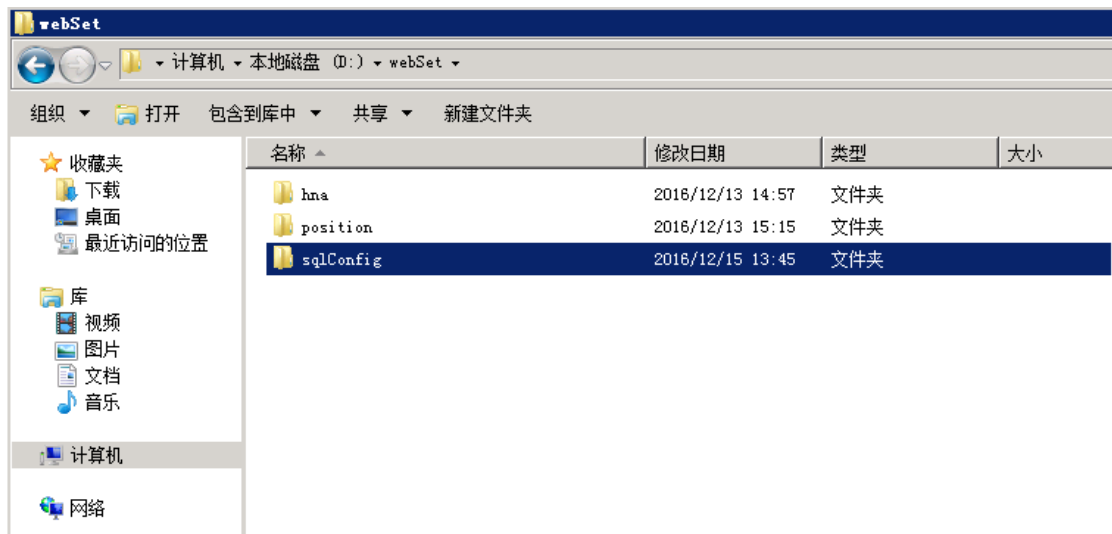
建立过程如下：

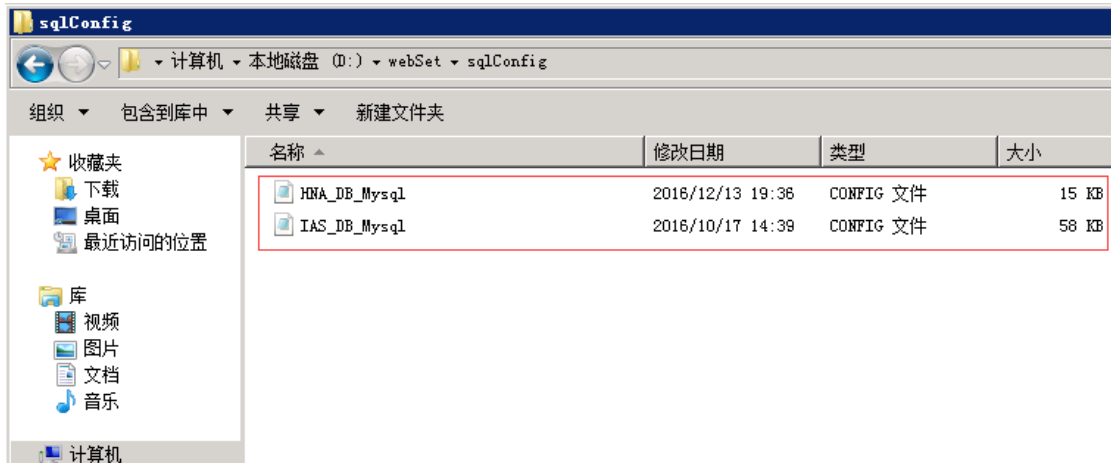
<1>登陆远程 windows 服务器

<2>在服务器上创建一个用户。在计算机右键→管理→本地用户和组→用户→右键→新建用户→输入用户名和密码创建。

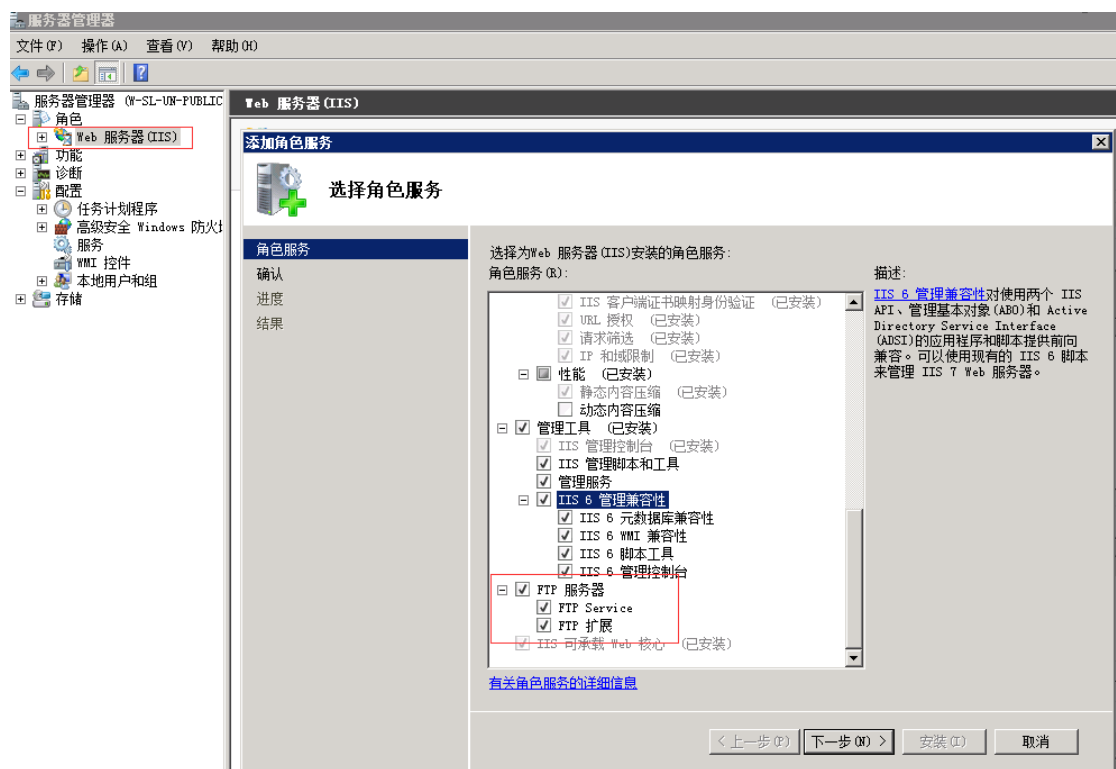


<3>在项目的根目录下，建立文件夹 sqlConfig，用来存放项目的 sql 配置文件





<4>添加 FTP 服务器，安装



<5>在 IIS 上新建 FTP 站点

添加 FTP 站点

站点信息

FTP 站点名称 (T):
sqlFtp

内容目录
物理路径 (H):
D:\webSet\sqlConfig

上一页 (P) 下一步 (N) 完成 (F) 取消

添加 FTP 站点

绑定和 SSL 设置

绑定

IP 地址 (A): 10.95.3.55 端口 (P): 545

☐ 启用虚拟主机名 (E):
虚拟主机 (示例: ftp.contoso.com) (V):

☒ 自动启动 FTP 站点 (S)

SSL

☒ 无
☐ 允许
☐ 需要

SSL 证书 (C): 未选定 查看 (V)...

上一页 (P) 下一步 (N) 完成 (F) 取消

<6>在授权中选中指定用户，输入我们刚才建的用户，勾选其读写权限

添加 FTP 站点

身份验证和授权信息

身份验证

☐ 匿名 (A)

☒ 基本 (B)

授权

允许访问 (C):

指定用户

tescomm1

权限

☒ 读取 (R)

☐ 写入 (W)

上一页 (P) 下一步 (N) **完成 (F)** 取消

<7>ftp 站点建立完成，在地址栏中输入 **ftp://10.95.3.55:545** 测试我们的站点是否新建成功，出现输入用户名和密码输入框，输入用户名和密码，则可以看到我们 **ftp** 下的文件。则代表新建 **ftp** 站点成功。

5. 文件配置

设置完 **ftp** 站点后，便可以在鉴权系统的文件配置里配置我们的 **sql** 配置文件。

在鉴权系统→文件配置下，点击新增按钮：

增加文件配置

项目：

DAAS

文件名称：

DAAS_DB_MySQL

文件路径：

用户名称：

密码：

保存

取消

增加配置文件，选择项目，文件名称，文件路径输入 ftp 的地址，用户名称输入 ftp 的用户名称，密码输入 ftp 的密码。点击保存。

6. sql 配置

在鉴权系统中，打开用户管理，选择一个用户，为该用户配置 sql 拦截。

SQL配置管理

项目

HNA

配置文件

HNA_DB_Mysql

Sql命令

原生SQL

动态参数解析

生成SQL

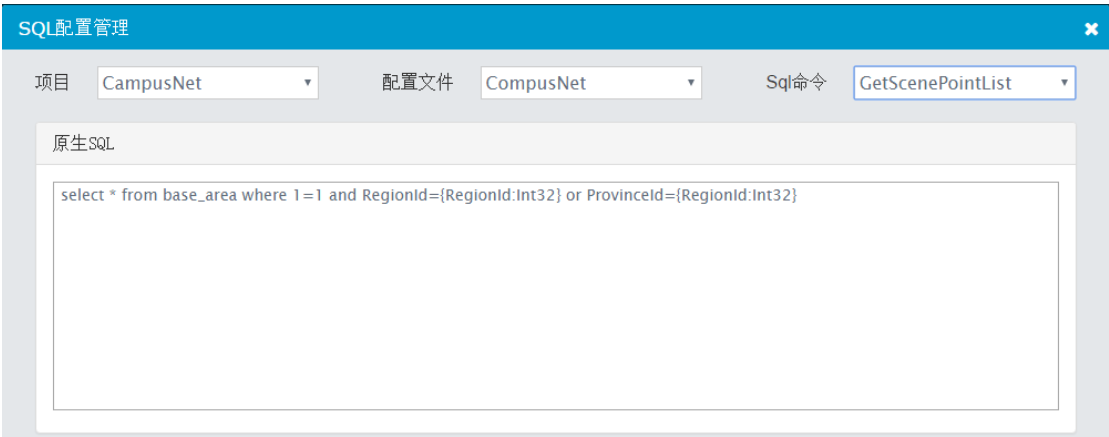
生成SQL

执行结果

在 sql 配置管理界面中，分为五部分。

第一部分为：项目，配置文件，Sql 命令。

选择项目→选择该项目的配置文件→选择该配置文件的 sql 命令，

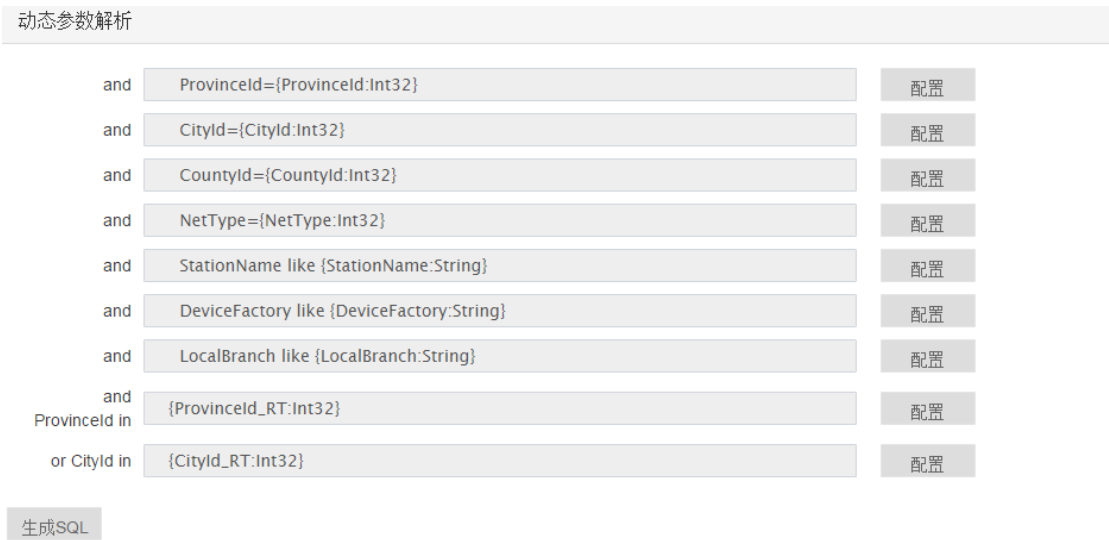


第二部分为：原生 SQL。

此部分为 sql 模板中非动态加载参数的固定的 sql

第三部分为：动态参数解析

在 sql 配置中，有动态需要添加的参数，这部分显示动态添加的参数，点击配置，我们可以设置参数的值给予添加。



第四部分为：生成 SQL

此部分显示我们按照自己意愿设置好的完整 sql

第五部分为：执行结果

执行设置好的完整 sql，显示执行结果

原生 sql 框中出现了：

1. 没有动态参数情况下：

在没有动态参数的情况下，呈现的是我们具体的完整的 sql。这个 sql 可以按照我们的意愿去修改它。可以给它添加新的参数和值，也可以修改参数值，例如修改为：

```
select * from base_area where 1=1 and RegionId=101 or ProvinceId=101
```

SQL配置管理

项目 CampusNet 配置文件 CompusNet Sql命令 GetScenePointList

原生SQL

select * from base_area where 1=1 and RegionId=101 or ProvinceId=101

动态参数解析

生成SQL

生成SQL

select * from base_area where 1=1 and RegionId=101 or ProvinceId=101

执行SQL 保存SQL

执行结果

ten_id	RegionId	RegionName	RegionPid	RegionType	ProvinceId	Lng	La
1e4e3cc4d0934...	101	北京	0	1	101		
1e4e3cc4d0934...	10100	北京市	101	2	101		
1e4e3cc4d0934...	1010000	市辖区	10100	3	101		
1e4e3cc4d0934...	1010001	东城区	10100	3	101		
1e4e3cc4d0934...	1010002	西城区	10100	3	101		
1e4e3cc4d0934...	1010003	朝阳区	10100	3	101		
1e4e3cc4d0934...	1010004	丰台区	10100	3	101		
1e4e3cc4d0934...	1010005	海淀区	10100	3	101		
1e4e3cc4d0934...	1010006	昌平区	10100	3	101		
1e4e3cc4d0934...	1010007	顺义区	10100	3	101		
1e4e3cc4d0934...	1010008	通州区	10100	3	101		
1e4e3cc4d0934...	1010009	大兴区	10100	3	101		
1e4e3cc4d0934...	1010010	门头沟区	10100	3	101		
1e4e3cc4d0934...	1010011	石景山区	10100	3	101		
1e4e3cc4d0934...	1010012	房山区	10100	3	101		
1e4e3cc4d0934...	1010013	平谷区	10100	3	101		
1e4e3cc4d0934...	1010014	密云区	10100	3	101		
1e4e3cc4d0934...	1010015	怀柔区	10100	3	101		
1e4e3cc4d0934...	1010016	延庆区	10100	3	101		
1e4e3cc4d0934...	1010017	昌平区	10100	3	101		
1e4e3cc4d0934...	1010018	昌平区	10100	3	101		
1e4e3cc4d0934...	1010019	昌平区	10100	3	101		
1e4e3cc4d0934...	1010020	昌平区	10100	3	101		

点击生成 SQL→执行 SQL→执行结果会显示你修改后的 sql 的执行结果

1. 如果这就是你想要的结果，想要这个 sql 替代程序中这条 sql 执行，则点击保存 SQL，保存到数据库中。
2. 如果想在不变它原来的参数的情况下添加条件，则可以在 sql 的后面填上自己想要的条件，点击生成 SQL，然后保存 SQL。

在具体项目中，该用户登录时，运行到这条 SQL 的时候，则会你配置后的这条 sql 会替换原先的 sql，来满足用户的特定需求。

2. 存在动态 sql 参数情况下：

<1>存在{:dn:}{n=1,2,3,.....)情况下

SQL配置管理

项目

CampusNet

配置文件

CompusNet

Sql命令

GetStation2G

原生SQL

select * from (select * from Base_Station_2G where 1=1 {d1:} {d2:} {d3:} {d4:} {d5:} {d6:} {d7:})as AA where 1=1 {d8:} {d9:}

动态参数解析

and

ProvinceId={ProvinceId:Int32}

配置

and

CityId={CityId:Int32}

配置

and

CountyId={CountyId:Int32}

配置

and

NetType={NetType:Int32}

配置

and

StationName like {StationName:String}

配置

and

DeviceFactory like {DeviceFactory:String}

配置

and

LocalBranch like {LocalBranch:String}

配置

and

ProvinceId in {ProvinceId_RT:Int32}

配置

or CityId in {CityId_RT:Int32}

配置

生成SQL

可以把{:dn:}(n=1,2.....)替换为自己想要添加的条件

在下面的参数解析中，可以点击配置，设置参数具体值。设置完成后，点击生成 SQL

原生SQL

select * from (select * from Base_Station_2G where 1=1)as AA where 1=1

动态参数解析

and

ProvinceId=101

配置

and

CityId={CityId:Int32}

配置

and

CountyId={CountyId:Int32}

配置

and

NetType={NetType:Int32}

配置

and

StationName like {StationName:String}

配置

and

DeviceFactory like {DeviceFactory:String}

配置

and

LocalBranch like {LocalBranch:String}

配置

and

ProvinceId in {ProvinceId_RT:Int32}

配置

or CityId in

{CityId_RT:Int32}

配置

生成SQL

生成SQL

select * from (select * from Base_Station_2G where 1=1)as AA where 1=1 and ProvinceId=101

执行SQL














保存SQL

执行结果

点击执行 SQL，如果是想要的结果，则可以保存 SQL 到数据库中。

7. 替换 dll

在项目中，因为要进行 sql 拦截，框架修改后，需要替换 dll。在项目的 framework 下替换相应的 dll

	Tescomm.Framework.Auth.IProvider.dll	2016/12/13 10:58	应用程序扩展	16 KB
	Tescomm.Framework.Auth.Provider.dll	2016/12/13 10:58	应用程序扩展	15 KB
	Tescomm.Framework.Common.dll	2016/12/13 10:58	应用程序扩展	61 KB
	Tescomm.Framework.DB.dll	2016/12/13 10:58	应用程序扩展	72 KB
	Tescomm.Framework.DB.IProvider.dll	2016/12/13 10:58	应用程序扩展	9 KB
	Tescomm.Framework.DB.MySql.dll	2016/10/17 11:51	应用程序扩展	8 KB
	Tescomm.Framework.DB.Oracle.dll	2016/10/17 11:51	应用程序扩展	8 KB
	Tescomm.Framework.Message.Provi...	2016/12/13 10:58	应用程序扩展	10 KB
	Tescomm.Framework.Session.IProvid...	2016/12/13 10:58	应用程序扩展	5 KB
	Tescomm.Framework.Session.MemPr...	2016/12/13 10:58	应用程序扩展	7 KB
	Tescomm.ServiceEngine.Commonom.dll	2016/12/13 10:58	应用程序扩展	12 KB
	Tescomm.ServiceEngine.dll	2016/11/10 11:16	应用程序扩展	30 KB
	Tescomm.ServiceEngine.Owin.dll	2016/11/10 11:17	应用程序扩展	37 KB

Dll 链接 dll