文章
Louis Lu · 九月 26, 2021 阅读大约需 7 分钟

IRIS 2021 技术文档 First Look 10 -- InterSystems .Net 网关

本技术概要（ First Look）通过重点概述和一个基础的、实际操作的示例，向您介绍在 InterSystems IRIS®数据平台上使用 .NET 网关（Gateway）的基础知识。您将了解 InterSystems IRIS 如何与 .NET 程序集进行互操作，在这个示例中，您将创建一个 .NET 网关（Gateway），并从 InterSystems IRIS 中的代理类调用基础 DLL。

本文档设计简单；当您把 .NET 网关（Gateway）引入您的生产系统时，您可能需要做一些不同的事情。本文档末尾的参考资料，提供了有关在生产中使用 .NET 网关（Gateway）的详细和完整的信息。

要浏览所有的技术概要（First Look），包括其他可以在免费的云实例或 web 实例上执行的技术概要（First Look），请参见 InterSystems First Looks（InterSystems 技术概要）。

1. 为什么 .NET 网关（Gateway）很重要

InterSystems IRIS Object Gateway for .NET（也称为".NET 网关（Gateway）"）为InterSystems IRIS 与 Microsoft .NET Framework 组件进行互操作提供了一种简便的方法。在使用网关（Gateway）导入 .NET DLL 后，您可以实例化一个外部 .NET 对象，InterSystems IRIS通过代理类将其作为的一个本地对象来操作。

每个代理对象（proxy object）都与相应的 .NET 对象通信，使您能够从 InterSystems IRIS 和 ObjectScript 中访问您的 .NET 类和方法。调用任何 InterSystems IRIS 代理方法都会将消息发送给 .NET 网关（Gateway）工作线程，该线程会找到合适的方法或构造函数调用。调用的结果发送回代理对象，再将结果返回给 InterSystems IRIS 应用程序。

通常，使用 .NET 网关（Gateway）的最佳方法是在您的应用程序中为第三方 DLL 构建一个小型包装器类（a small wrapper class），然后为包装器生成代理类。包装器类只公开您想要的功能，而不是创建大量应用程序可能不需要的代理类。

2. 探索 .NET 网关（Gateway）

在这个实际操作的示例中，您将：

- 创建一个 DLL，其中包含要从 InterSystems IRIS 调用的示例类
- 定义一个 .NET 网关（Gateway），并启动服务器进程
- 创建一个 ObjectScript 类，从 DLL 生成代理类
- 创建另一个 ObjectScript 类，以连接到网关（Gateway）并通过代理对象操作 DLL

想试试 InterSystems IRIS .NET 开发和互操作性功能的在线视频演示吗？请看.NET QuickStart（.NET 快速入门）!

2.1 用前须知

要运行这个演示，您需要一个运行 Microsoft .NET Framework 4.5 版本 的 Windows 10 系统、Visual Studio，以及一个已安装的 InterSystems IRIS 实例。(有关安装 InterSystems IRIS 的说明，请参见 InterSystems IRIS Basics:Installation [《InterSystems IRIS 基础：安装》]。)

有关配置 Visual Studio 以连接到您的 InterSystems IRIS 实例的说明，请参见 InterSystems IRIS Basics:Connecting an IDE（《InterSystems IRIS 基础: 连接一个 IDE》）中的 InterSystems IRIS Connection Information（InterSystems IRIS 连接信息）和 .Net IDE。

您还将使用InterSystems 的 Studio IDE（一个在 Windows 系统上运行的客户端应用程序）来创建 ObjectScript 代码；更多信息，请参见InterSystems IRIS Basics: Connecting an IDE（《InterSystems IRIS 基础: 连接一个 IDE》）中的 Using Studio（使用 Studio） 和 Studio。）

2.2 创建 DLL

使用 Visual Studio，创建一个名为 Person 的类并复制以下 C# 代码。在本例中使用 .NET 4.5。

```
public class Person {
    public int age;
    public String name;

    //constructor
    public Person (int startAge, String Name) {
        age = startAge;
        name = Name;
    }

    public void setAge(int newAge) {
        age = newAge;
    }

    public String getName() {
       return name;
    }

    public int getAge() {
       return age;
    }

    public static void main(String []args) {
        Person myPerson = new Person (5, "Tom");
        Console.Out.WriteLine(myPerson.getName());
        Console.Out.WriteLine(myPerson.getAge());
    }
}
```

编译 Person 类，并生成一个 Person.dll 文件。注意 DLL 的位置，因为您稍后会需要它。

2.3 创建并启动 .NET 网关

1. 使用 InterSystems IRIS Basics：Connecting an IDE（《InterSystems IRIS 基础: 连接一个 IDE》）中的 InterSystems IRIS Connection Information（InterSystems IRIS 连接信息）描述的 URL，在浏览器中打开您的实例的管理门户（Management Portal）。
2. 导航至 System Administration（系统管理） > Configuration（配置） > Connectivity（连接） > External Language Servers（外部语言服务）。
3. 选择 Create External Language Server（创建外部语言服务）。
4. 输入 Server Name（网关名称）。
5. 在 Server Type （服务类型）中选择 .Net
6. 在 Port（端口）字段中，输入 55000。
7. 对于 .NET Version（.NET 版本），请选择 4.5。
8. 选择 Save（保存）。
9. 在External Language Servers（外部语言服务）页面，找到您刚刚定义的网关（Gateway），并选择 Start（开始）

2.4 生成代理类

使用 Studio, 在实例的 USER 命名空间中创建一个名为 CreateProxyClasses.cls 的新 ObjectScript 类，并粘贴以下代码， 将您的实例的主机标识符替换为 gwyConn.%Connect 中的 server ，并用双引号括起来的 Person.dll 文件的完整文件路径替换 YOUR FILEPATH HERE。

```
Class User.CreateProxyClasses Extends %Persistent
{
ClassMethod run()
{
   // get a connection to the .NET Gateway
   set gwyConn = ##class(%Net.Remote.Gateway).%New()
   set status = gwyConn.%Connect("127.0.0.1", 55000, "USER")
   if $$$ISERR(status) {
       write !,"error: "_$system.OBJ.DisplayError(status)
       quit
   }

   // add the DLL to the classpath
   set classpath = ##class(%ListOfDataTypes).%New()
   do classpath.Insert("YOUR FILEPATH HERE")
   set status = gwyConn.%AddToCurrentClassPath(classpath)
   if $$$ISERR(status) {
      write !,"error: "_$system.OBJ.DisplayError(status)
      quit
   }

   // create the proxy ObjectScript classes corresponding to the .NET classes in the
DLL
   set status = gwyConn.%Import("Person",,,,1)
   if $$$ISERR(status) {
      write !,"error: "_$system.OBJ.DisplayError(status)
      quit
   }

   //close the connection to the .NET Gateway
   set status = gwyConn.%Disconnect()
   if $$$ISERR(status) {
      write !,"error: "_$system.OBJ.DisplayError(status)
      quit
   }
}
}
```

编译并构建该类。然后使用 InterSystems IRIS Basics:Connecting an IDE（InterSystems IRIS 基础: 连接一个 IDE） 中的 instructions for your instance（对您的实例的说明），在 USER 命名空间中打开 InterSystems 终端（InterSystems Terminal）, 并执行以下命令：

```
do ##class(User.CreateProxyClasses).run()
```

2.5 使用ObjectScript 操作 .NET Object

在 USER 命名空间中创建另一个名为 ManipulateObjects.cls 的 ObjectScript 类，并粘贴以下代码
(注意要将您的实例的主机标识符替换 gwyConn.%Connect 中的 第一个参数):

```
Class User.ManipulateObjects Extends %Persistent
{
ClassMethod run()
{
    // get a connection to the .NET gateway
    set gwyConn = ##class(%Net.Remote.Gateway).%New()
    set status = gwyConn.%Connect("127.0.0.1", 55000, "USER")
    if $$$ISERR(status) {
       write !,"error: "_$system.OBJ.DisplayError(status)
       quit
    }

    // manipulate some proxy objects
    set person = ##class(User.Person).%New(gwyConn,5,"Tom")
    write !,"Name: "_person.getName()
    write !,"Age: "_person.getAge()
    do person.setAge(100)
    write !,"Age: "_person.getAge()

    // close the connection to the .NET gateway
    set status = gwyConn.%Disconnect()
    if $$$ISERR(status) {
       write !,"error: "_$system.OBJ.DisplayError(status)
       quit
    }
}
}
```

编译并构建该类，然后在终端（Terminal）中的 USER 命名空间中执行以下命令：

```
do ##class(User.ManipulateObjects).run()
```

您应该可以看到以下输出：

```
Name: Tom Age: 5
setting age to 100
Age: 100
```

现在您已经成功地完成了练习，停止您创建的 .NET 网关（Gateway）。返回到管理门户（Management Portal）中的 External Language Servers（外部语言服务）页面，找到网关（Gateway），并选择 Stop（停止）。

3. 了解更多有关 .NET 网关（Gateway）的信息

从这里，您可以继续探索 .NET 网关（Gateway）和 InterSystems IRIS。请参阅下面的文档和参考资料，了解 .NET、互操作性、应用程序开发等。

- Using the Gateway for .NET（使用 .NET 网关） — 了解更多有关 InterSystems IRIS 和 Microsoft .NET

Framework 组件之间互操作性的信息。
- Skyrocket Your .NET Application Development（Skyrocket Your .NET 应用程序开发） — 这是 InterSystems 关于在 .NET 中作为对象建模和访问数据的演示。
- .NET Documentation（.NET 文档） — Microsoft 在 .NET 上的文档，包括架构概念、教程和开发指南。

#InterSystems IRIS #InterSystems IRIS for Health

源 URL: https://cn.community.intersystems.com/post/iris-2021-%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3-first-look-10-intersystems-net-%E7%BD%91%E5%85%B3