

单导系统数据接收端服务发布规范

单导系统接收端文件服务器接收到数据以后，需转发至对应的应用系统，因此应用系统需发布一个 webservice 服务来接收数据，为方便统一调用，需按照一定的要求来发布服务。服务需提供一个参数 sContent,接收数据格式为 json 字符串，格式如下：

```
{
  "oriSysName": "制造执行管理系统",
  "oriSysNum": "MES",
  "fileData": [{"fileName": "xxxxx", "fileData": "xxxxxx"}, {"fileName": "xxx", "fileData": "xxxx"}],
  "uniqueFlag": "1000001",
  "timestamp": "85653652"
}
```

各 key 值含义如下

| 参数名 | 参数说明 | 备注 |
|------------|---------------|---------------------------------------------------------|
| oriSysName | 发送系统名称 | |
| oriSysNum | 发送系统唯一编号 | |
| fileData | 文件数据（一个或多个文件） | fileName 为文件名，fileData 为文件数据（文件转 byte 数组再转为 Base64 字符串） |
| uniqueFlag | 接口对应唯一业务编号 | 根据该编号来调用对应处理方法 |
| timestamp | 发送时间戳 | |

下面以 JAVA 和 C#为例说明如何发布所需的服务。

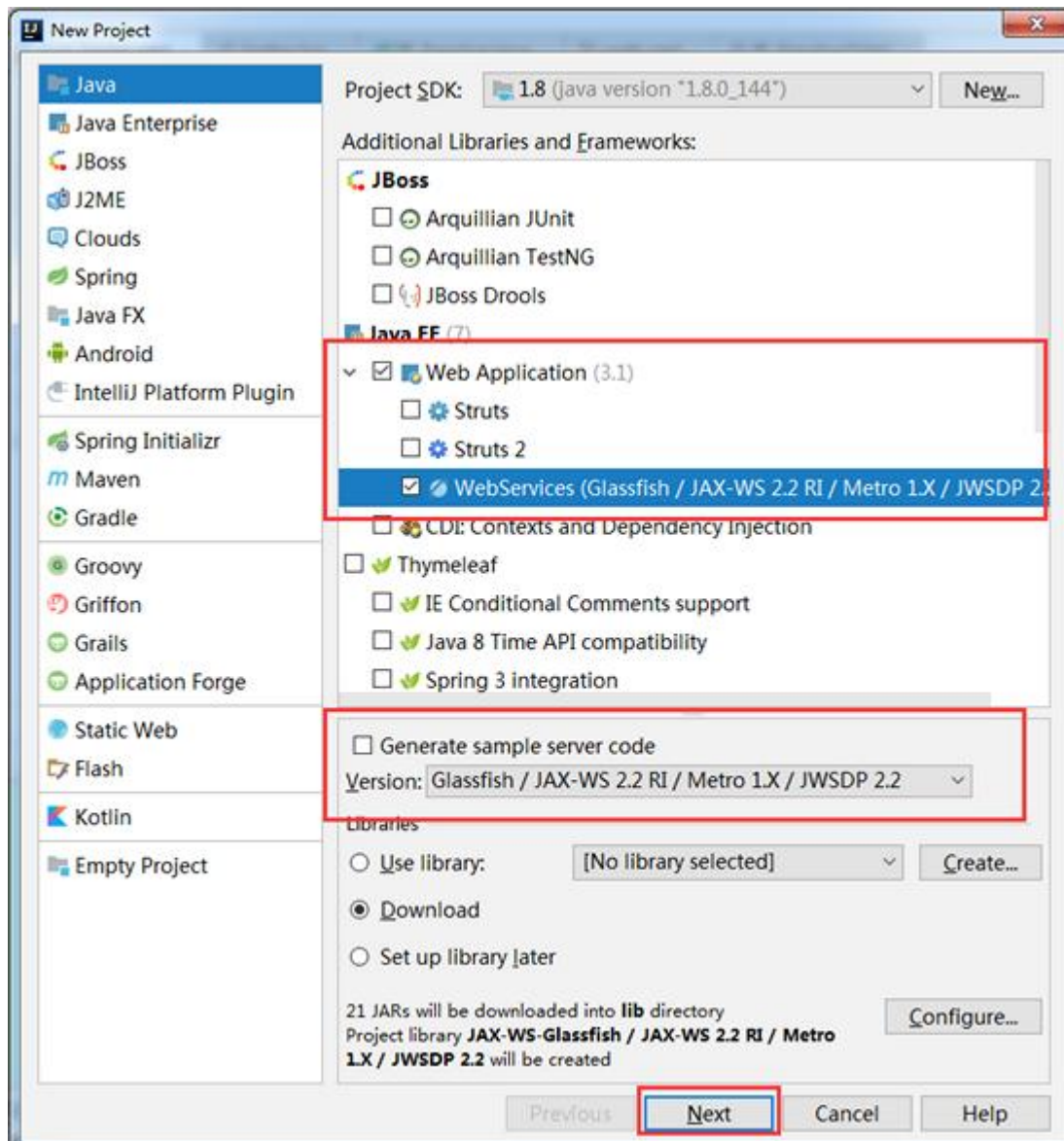
1. Java 代码服务端发布

1.1 IDEA 工具生成服务端

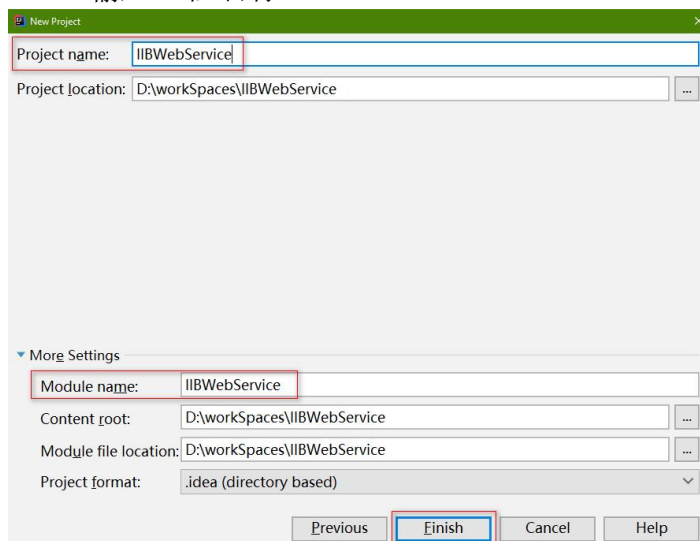
1.1.1 新建项目工程



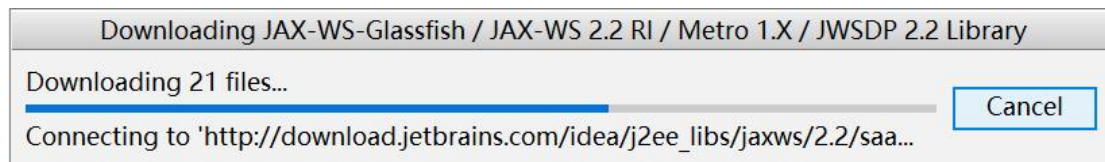
1.1.2 选择 Web Application 工程 创建基于 Java EE 的 Web Service 框架



1.1.3 输入工程名称 IIBWebService

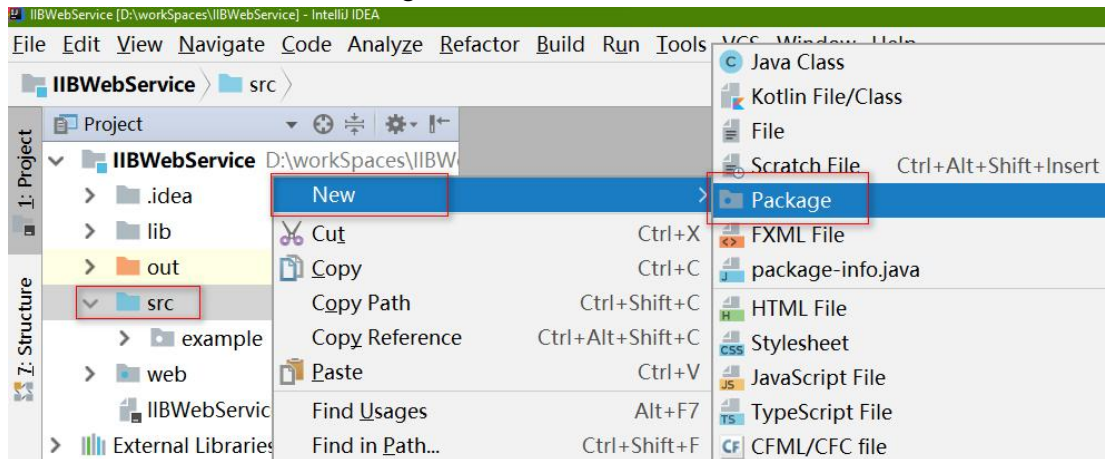


点击完成后，IDEA 会在联网状态下自动下载相关的工程依赖包。

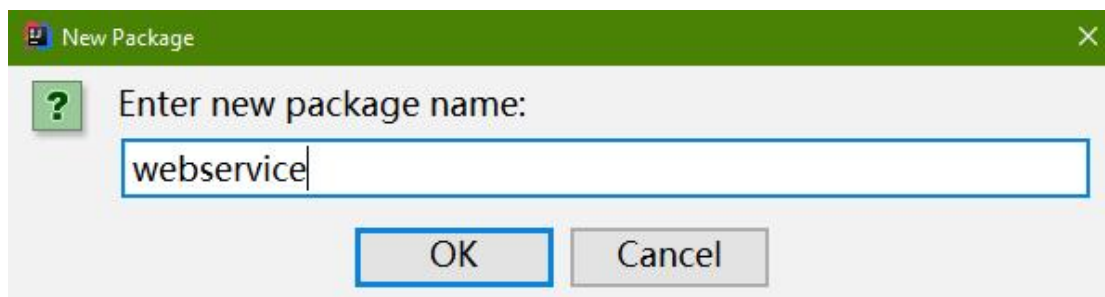


1.1.4 创建 webservice 工程包

单击 src, 右键 New, 选择 Package

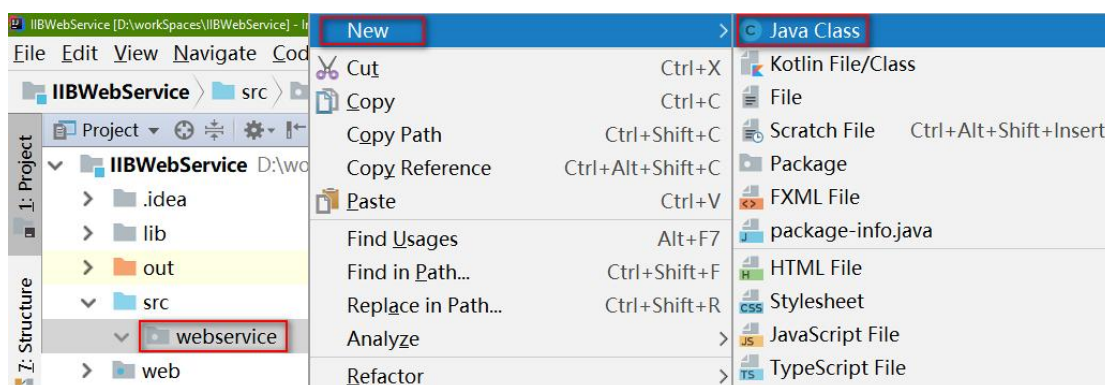


输入 webservice

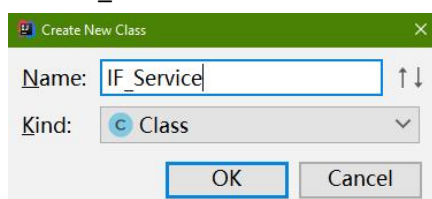


1.1.5 创建 IF_Service 类文件

单击 webservice, 右键 New, 选择 Java Class



输入 IF_Service 并确定



1.1.6 编写 IF_Service 类代码

```
@WebService(serviceName = "IF_ServiceService", targetNamespace = "http://www.pxxygc.com/IF_Service")
public class IF_Service {
    @WebMethod(operationName = "IFService")
    @WebResult(name="IFServiceReturn")
    public String IFService(@WebParam(name="sContent", targetNamespace = "http://www.pxxygc.com/IF_Service") String sContent) {
        String str = ("status\":\"1\", \"msg\":\"发送成功\");
        System.out.println(str);
        return str;
    }
}

public static void main(String[] argv) {
    Object implementor = new IF_Service();
    String address = "http://localhost:8080/services/IF_Service";
    Endpoint.publish(address, implementor);
}
```

1.1.7 修改 sun-jaxws.xml 文件

依次选择 web，WEB-INF，打开 sun-jaxws.xml 文件



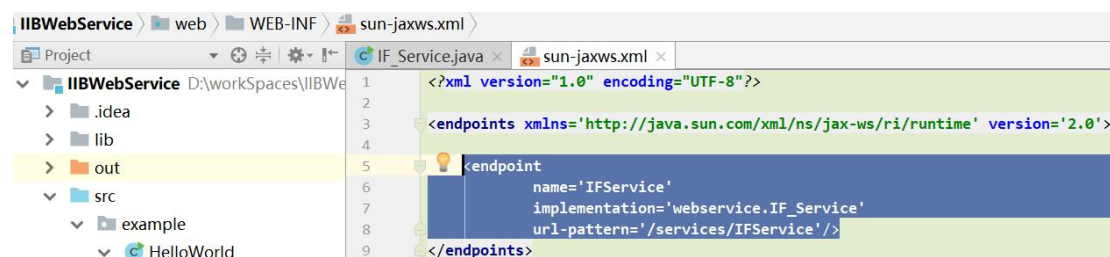
将如下内容：

```
<endpoint name="HelloWorld"
    implementation="example.HelloWorld"
    url-pattern="/services/HelloWorld"/>
```

替换为：

```
<endpoint
    name='IFService'
    implementation='webservice.IF_Service'
    url-pattern='/services/IFService'/>
```

修改后



1.1.8 运行测试工程

右键运行 IF_Service.main()方法，打开浏览器，访问：http://localhost:8080/services/IF_Service 见如下图例，说明 WebService 已经发布成功。

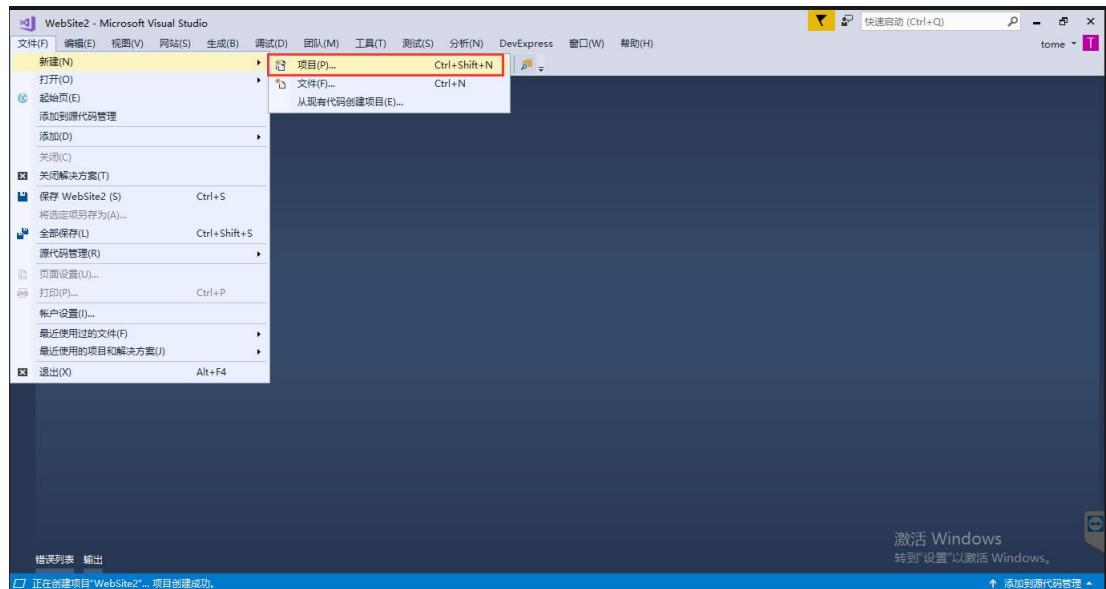
Web Services

| Information | |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Address: | http://localhost:8080/services/IF_Service |
| WSDL: | http://localhost:8080/services/IF_Service?wsdl |
| Implementation class: webservice.IF_Service | |

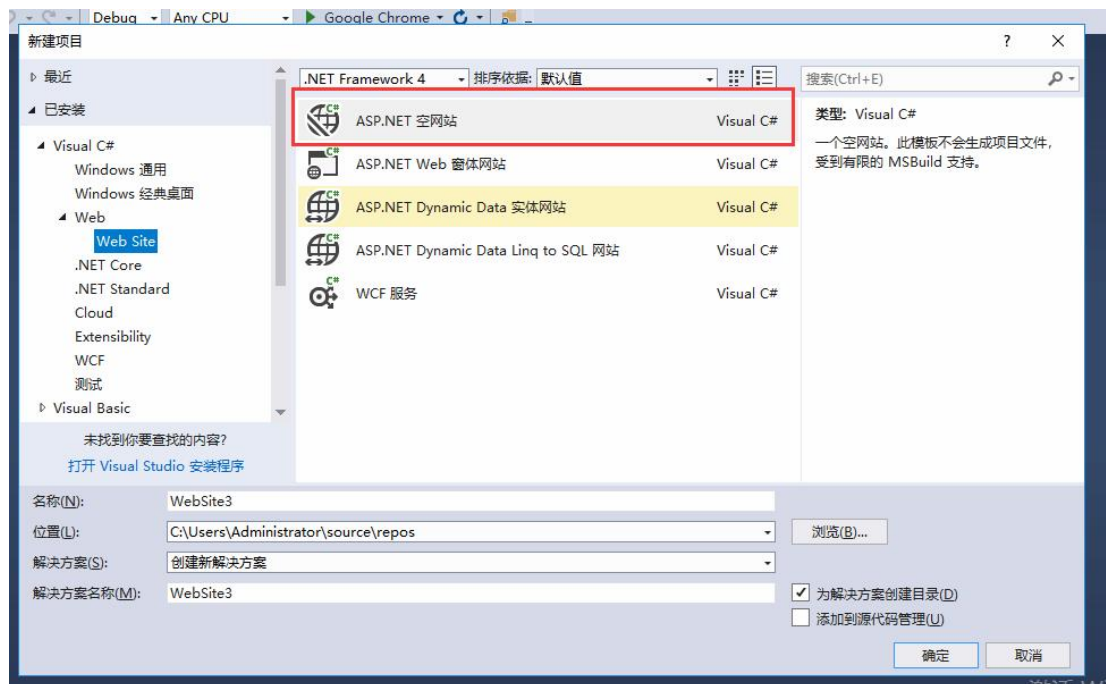
2. C#代码服务端发布

2.1 MS Visual Studio 工具生成服务端

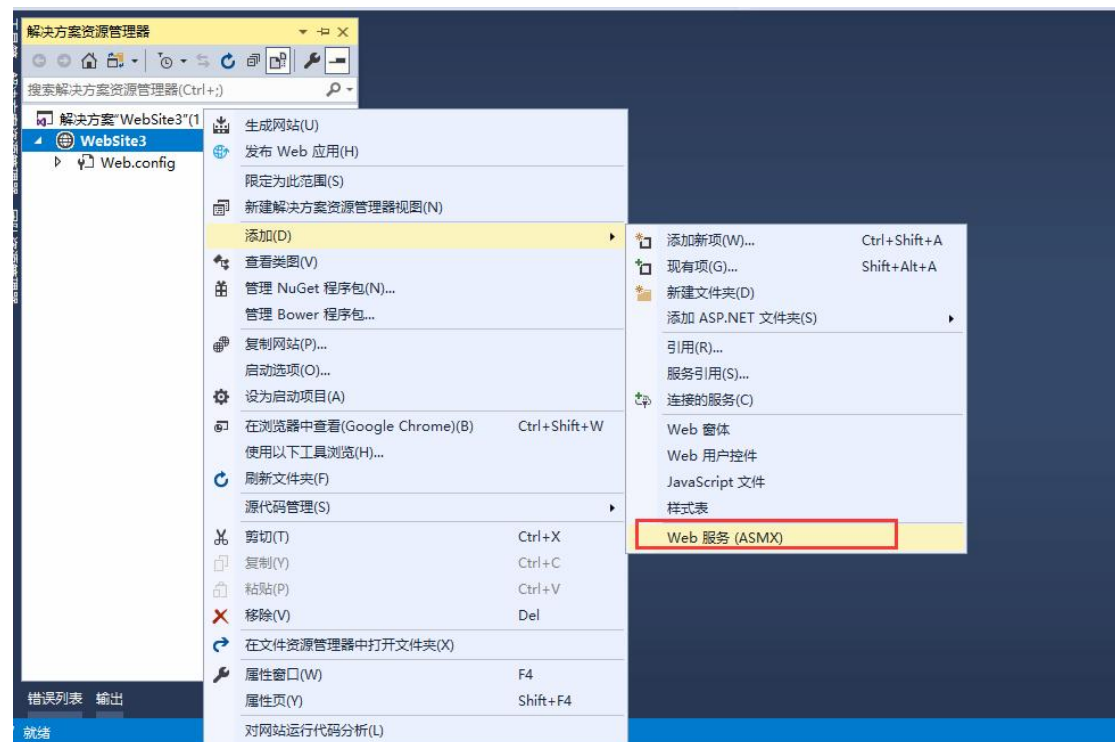
2.1.1 新建项目工程



2.1.2 创建解决方案



2.1.3 添加 Web 服务



服务名为 IF_Service

2.1.4 编写 IF_Service 代码

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Services;
6 using System.Web.Services.Protocols;
7
8 /// <summary>
9 /// IF_Service 的简要说明
10 /// </summary>
11 [WebService(Namespace = "http://www.pgxxgc.com/IF_Service")]
12 [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
13 [SoapDocumentService(RoutingStyle = SoapServiceRoutingStyle.RequestElement)]
14 [System.ComponentModel.ToolboxItem(false)]
15 public class IF_Service : System.Web.Services.WebService
16 {
17
18     public IF_Service()
19     {
20
21         //如果使用设计的组件，请取消注释以下行
22         //InitializeComponent();
23     }
24
25     [WebMethod]
26     [return: System.Xml.Serialization.XmlElementAttribute("IFServiceReturn")]
27     public string IFService(string sContent)
28     {
29
30         // 在此实现 服务内容
31         string str = "World";
32         return string.Format("Hello {0}", str);
33     }
34 }
```

2.1.5 发布 Webservice 服务

按 F5 键发布服务，浏览器会自动弹出并访问。

3. 客户端测试工具

3.1 Soap UI 测试

利用 soap ui 工具对发布的 webservice 地址进行测试，如生成如下代码，则表示发布的 webservice 服务符合需要。

