



链滴

SpringBoot2 集成 cxf 实现 Webservice 接口生成和客户端调用

作者: [k36666](#)

原文链接: <https://ld246.com/article/1577784756271>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

WebService简介

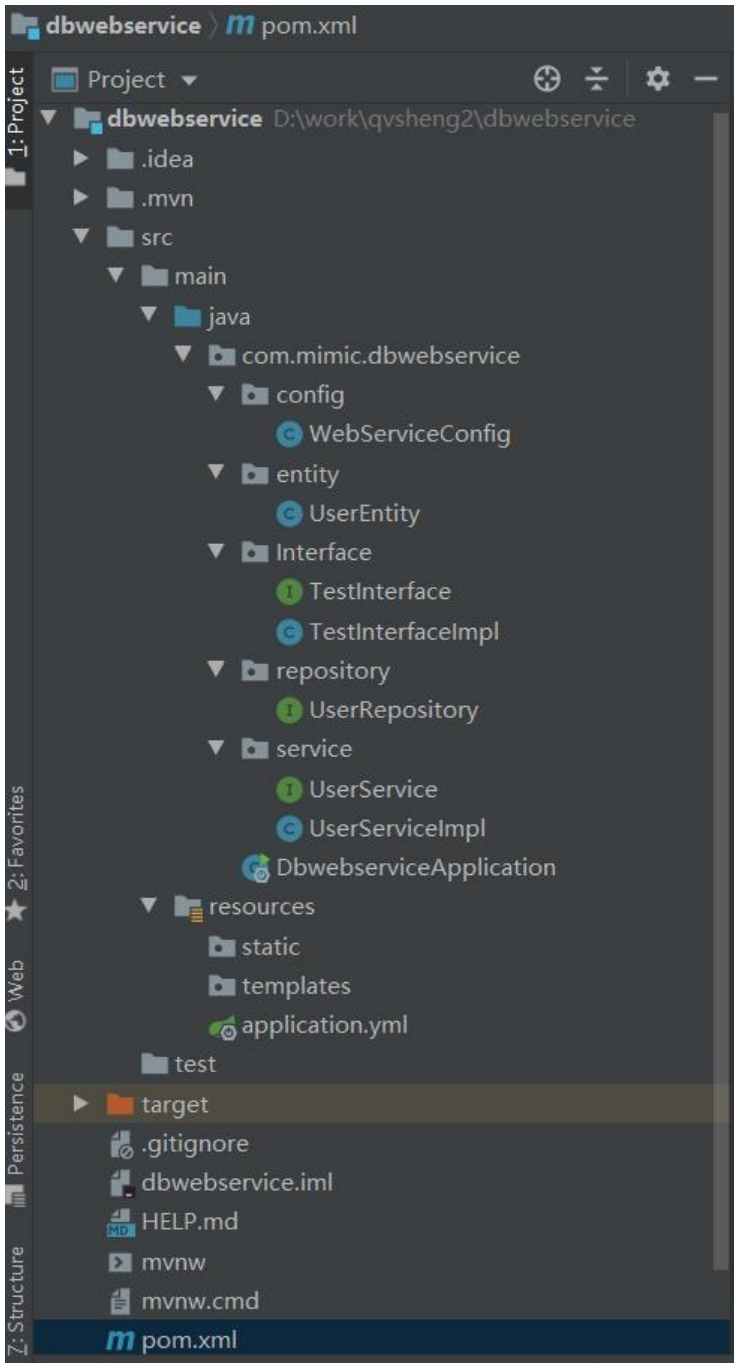
最近接了个新活，要给一个老系统写一些传数据的接口，用到了WebService，网上使用SpringBoot结合cxf实现WebService接口的教程中SpringBoot版本都较低，摸索了一下SpringBoot2集成cxf的方法，记录一下。

WebService是一种跨编程语言、跨操作系统平台的远程调用技术。

- 远程调用技术：远程调用是指一台设备上的程序A可以调用另一台设备上的方法B。比如：银联提供商场的pos刷卡系统，商场的pos机转账调用的转账方法的代码其实是跑在银行服务器上的。再比如，mazon，天气预报系统，淘宝网，校内网，百度等把自己的系统服务以WebService服务的形式暴露来，让第三方网站和程序可以调用这些服务功能，这样扩展了自己系统的市场占有率。
- 跨编程语言：是指服务端、客户端程序的编程语言可以不同
- 跨操作系统平台：是指服务端、客户端可在不同的操作系统上运行

服务器端代码

项目的目录结构如下，基于SpringBoot2.2.1和cxf3.2.9构建



pom.xml

```
/*引入cxf依赖*/  
<dependency>  
  <groupId>org.apache.cxf</groupId>  
  <artifactId>cxf-spring-boot-starter-jaxws</artifactId>  
  <version>3.2.9</version>  
</dependency>  
/*引入jpa依赖*/  
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-data-jpa</artifactId>  
</dependency>  
/*引入mysql驱动依赖*/
```

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

实体类

数据库表user结构如下

	username	password	uuid	name
▶	admin	admin	qwe464w6q6	admin
*	NULL	NULL	NULL	NULL

实体类代码

UserEntity.java

```
package com.mimic.dbwebservice.entity;
```

```
import javax.persistence.*;
import java.util.Objects;
```

```
@Entity
@Table(name = "user", schema = "login", catalog = "")
public class UserEntity {
    private String username;
    private String password;
    private String uuid;
    private String name;

    @Id
    @Column(name = "username")
    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    @Basic
    @Column(name = "password")
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    @Basic
    @Column(name = "uuid")
```

```

public String getUuid() {
    return uuid;
}

public void setUuid(String uuid) {
    this.uuid = uuid;
}

@Basic
@Column(name = "name")
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    UserEntity that = (UserEntity) o;
    return Objects.equals(username, that.username) &&
        Objects.equals(password, that.password) &&
        Objects.equals(uuid, that.uuid) &&
        Objects.equals(name, that.name);
}

@Override
public int hashCode() {
    return Objects.hash(username, password, uuid, name);
}

@Override
public String toString()
{
    return this.name+" "+this.password+" "+this.uuid;
}
}

```

repository类代码

repository.java

```

package com.mimic.dbwebservice.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import com.mimic.dbwebservice.entity.UserEntity;

public interface UserRepository extends JpaRepository<UserEntity,String> {
}

```

service接口及其实现类

UserService.java

```
package com.mimic.dbwebservice.service;
import com.mimic.dbwebservice.entity.UserEntity;

public interface UserService {
    public UserEntity getUser(String id);
}

```

UserServiceImpl.java

```
package com.mimic.dbwebservice.service;

import com.mimic.dbwebservice.entity.UserEntity;
import com.mimic.dbwebservice.service.UserService;
import com.mimic.dbwebservice.repository.UserRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class UserServiceImpl implements UserService{

    @Autowired
    UserRepository userRepository;
    @Override
    public UserEntity getUser(String id) {
        return userRepository.findById(id).orElse(null);
    }
}

```

服务发布接口及实现类

TestInterface.java

```
package com.mimic.dbwebservice.Interface;

import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService(targetNamespace = "http://interface.dbwebservice.mimic.com")
public interface TestInterface {
    @WebMethod
    public String getUser(String id);
}

```

TestInterfaceImpl.java

```
package com.mimic.dbwebservice.Interface;

import com.mimic.dbwebservice.entity.UserEntity;
import com.mimic.dbwebservice.service.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import org.springframework.stereotype.Component;

import javax.jws.WebMethod;
import javax.jws.WebService;
import java.util.ArrayList;
@WebService(serviceName = "UserService",
            targetNamespace = "http://interface.dbwebservice.mimic.com",
            endpointInterface = "com.mimic.dbwebservice.Interface.TestInterface")
@Component
public class TestInterfaceImpl implements TestInterface{
    @Autowired
    UserService userService;
    @Override
    public String getUser(String id)
    {
        String username="admin";
        UserEntity user = userService.getUser(username);
        return user.toString();
    }
}
```

WebService配置类

WebServiceConfig.java

```
package com.mimic.dbwebservice.config;

import com.mimic.dbwebservice.entity.UserEntity;
import com.mimic.dbwebservice.service.UserService;
import com.mimic.dbwebservice.Interface.TestInterface;
import org.apache.cxf.Bus;
import org.apache.cxf.jaxws.EndpointImpl;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import javax.xml.ws.Endpoint;
@Configuration
public class WebServiceConfig {
    @Autowired
    private Bus bus;
    @Autowired
    private TestInterface testInterface;
```

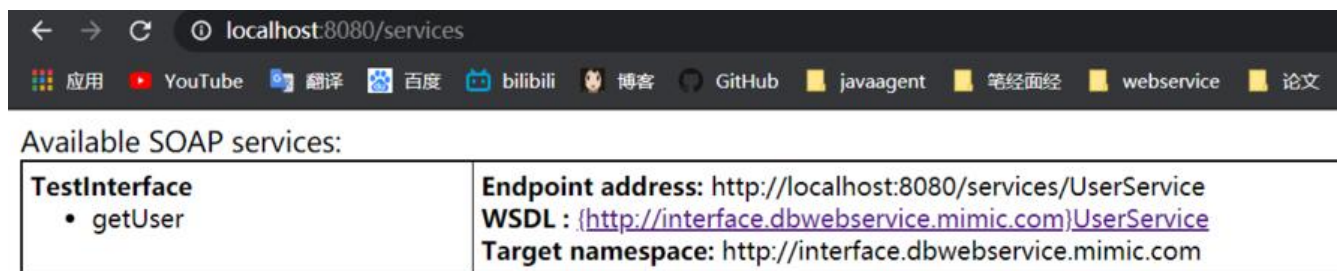
```
@Bean
public Endpoint endpointStudentService() {
    EndpointImpl endpoint = new EndpointImpl(bus,testInterface);
    endpoint.publish("/UserService");
    return endpoint;
}
}
```

application.yml配置文件

```
spring:
  datasource:
    url: jdbc:mysql://localhost:3306/login?serverTimezone=UTC
    username: 你的用户名
    password: 你的密码
    driver-class-name: com.mysql.cj.jdbc.Driver
  jpa:
    hibernate:
      ddl-auto: update
    show-sql: true
```

服务端完成

访问 <http://localhost:8080/services>, 出现可供选择的服务。



客户端代码

客户端引入cxf依赖

```
<dependency>
  <groupId>org.apache.cxf</groupId>
  <artifactId>cxf-spring-boot-starter-jaxws</artifactId>
  <version>3.2.9</version>
</dependency>
```


测试类代码

Test.java

```
package com.mimic.dbclient;

import org.apache.cxf.endpoint.Client;
import org.apache.cxf.jaxws.endpoint.dynamic.JaxWsDynamicClientFactory;
public class Test {
    public static void main(String[] args) {
        JaxWsDynamicClientFactory dcf = JaxWsDynamicClientFactory.newInstance();
        printInformation(dcf);
    }

    private static void printInformation(JaxWsDynamicClientFactory dcf) {
        Client client = dcf.createClient("http://localhost:8080/services/UserService?wsdl");
        Object[] object = new Object[0];
        try {
            object = client.invoke("getUser", "");
            System.out.println(object[0]);
        } catch (java.lang.Exception e) {
            e.printStackTrace();
        }
    }
}
```

运行测试类，结果如下，调用服务端接口成功

```
15:42:14.935 [main] DEBUG org.apache.cxf.ws.policy.PolicyVerificationInIr
admin admin qwe464w6q6

Process finished with exit code 0
```