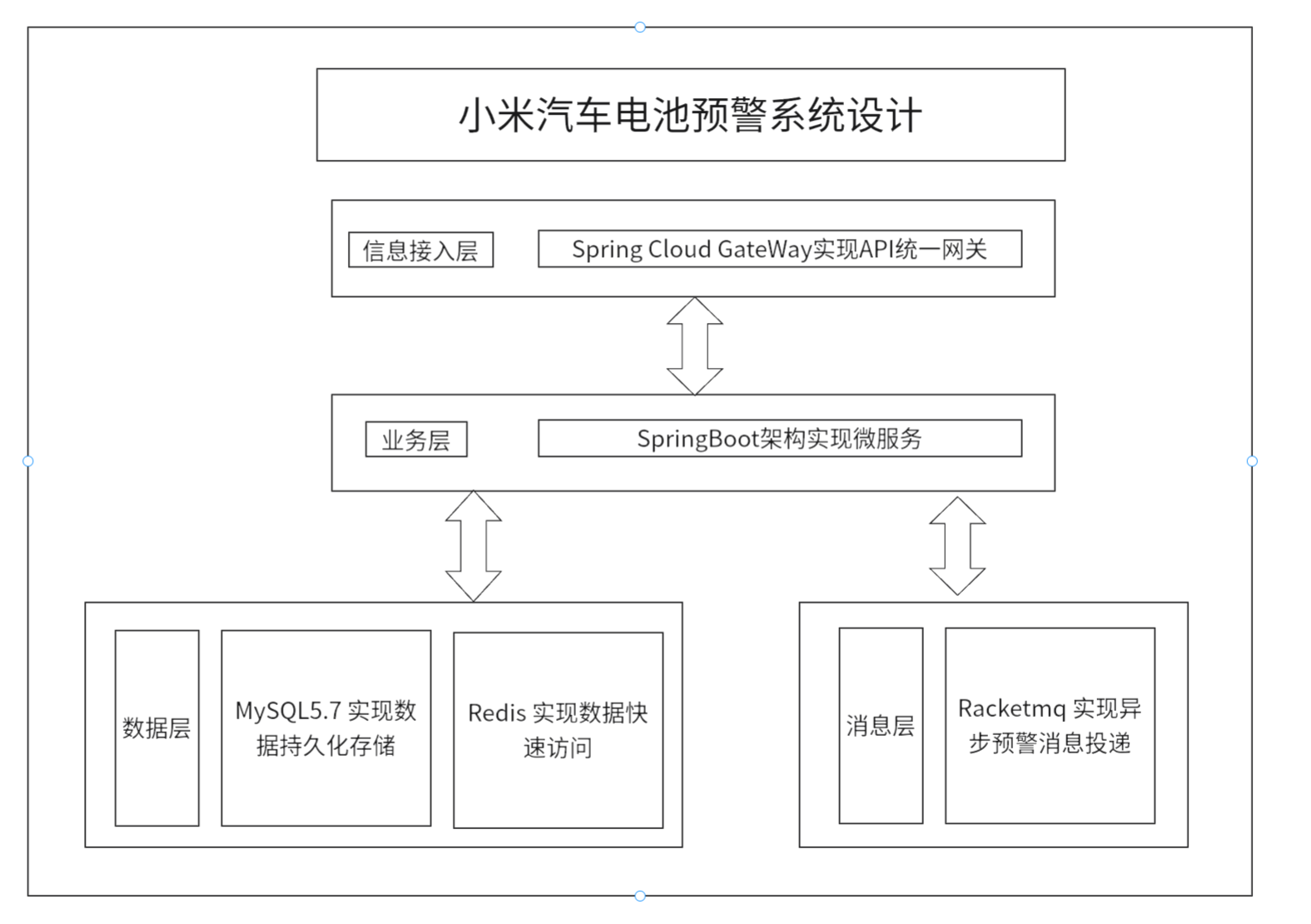
# 小米汽车电池预警技术方案

## 1 系统设计



在系统设计上我选择了四层结构 包括信息接入层 业务层 数据层 以及消息层。

在信息接入层，使用Spring Cloud GateWay实现API统一网关 端口统一为9000。

在业务层 我使用SpringBoot设计各种微服务包括车辆管理、规则管理、预警信息管理、预警信号管理等实体管理包括基础增删改查接口以及定时任务接口（向消费者定时输送未处理的信号）等业务逻辑。业务层既需要同信息接入层进行交互，即向外开放服务访问，同时也要同数据层进行交互，增删改查各类数据后对数据的持久化以及快速访问，同时也要与消息层进行交互。

在数据层，我使用MySQL5.7实现数据持久化存储同时使用redis存储一次查询后的数据从而实现二次及多次访问的快速查询，数据层即可以与业务层进行交互即基础增删改查等功能。

在消息层，我使用Rockermq5.3.0实现异步预警消息的投递，即在业务层中提供定时服务将未处理的信号信息投送至消息队列，同时在消费者端进行接收并将数据依次进行消费处理，对需要报警的信息进行处理后加入数据库进行持久化存储。

## 2 数据库表设计

### 2.1表vehicle （汽车表）

表描述：表中存放支持的数据名称和其字段描述如下表所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| 1 | vid | varchar | 16 | 0 | Y | N | 汽车vid(自生成) |
| 2 | Id | int | 11 | 0 | N | Y | 车架编号 |
| 3 | battery\_type | varchar | 20 | 0 | Y | N | 电池类型 |
| 4 | total\_mil | int | 11 | 0 | Y | N | 总里程 |
| 5 | bh\_state | int | 11 | 0 | Y | N | 健康状态 |

### 2.2表rule（规则表）

表描述：表中存放支持的数据名称和其字段描述如下表所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| 1 | id | int | 11 | 0 | N | Y | 序号 |
| 2 | warn\_id | int | 11 | 0 | Y | N | 规则编号 |
| 3 | battery\_type | varchar | 20 | 0 | Y | N | 电池类型 |
| 4 | name | varchar | 20 | 0 | Y | N | 名称 |
| 5 | warn\_rule | varchar | 20 | 0 | Y | N | 预警规则数 |

### 2.3表warn\_signal（信号表）

表描述：表中存放支持的数据名称和其字段描述如下表所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| 1 | id | int | 16 | 0 | N | Y | 信号id |
| 2 | warn\_id | int | 11 | 0 | Y | N | 规则编号 |
| 3 | car\_id | int | 20 | 0 | N | N | 车架编号 |
| 4 | cwsignal | varchar | 20 | 0 | N | N | 信号 |
| 5 | signal\_state | int | 20 | 0 | Y | N | 处理状态 （1代表处理 0代表未处理） |

### 2.4表warn\_message（警告信息表）

表描述：表中存放支持的数据名称和其字段描述如下表所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 数据类型 | 长度 | 小数位 | 允许空值 | 主键 | 说明 |
| 1 | id | int | 16 | 0 | N | Y | 信号id |
| 2 | signal\_id | int | 11 | 0 | Y | N | 信号id |
| 3 | car\_id | int | 20 | 0 | Y | N | 电池类型 |
| 4 | battery\_type | varchar | 20 | 0 | Y | N | 电池类型 |
| 5 | warn\_name | varchar | 20 | 0 | Y | N | 警告名 |
| 6 | warn\_level | int | 20 | 0 | Y | N | 警告等级 |

## 接口设计

### 添加电池信号

接口地址：

*[http://localhost:9000/signal/add](http://localhost:80/id-report-manage/manage/hpInfectiousDisease/countTrendByQuarter)*

请求类型：*application/json*

请求方式：POST

接口备注：添加预警信号信息

调试工具：*ApiPost*

请求体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| carId | 3 | int | 车架id |
| warnId | 1 | int | 规则编号 |
| signal | "{\"Mx\":12.0,\"Mi\":0.6}" | Varchar | 信号数据 |

请求示例：

|  |
| --- |
| {  "carId": 3,  "warnId": 1,  "signal": "{\"Mx\":12.0,\"Mi\":0.6}"  } |

返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| success | true | bool | 调用接口成功 |
| message | 操作成功！ | Varchar | 返回接口信息 |
| code | 200 | int | 响应码 |
| data | 1 | int | 添加成功与否 |

### 修改电池信号

接口地址：

*[http://localhost:9000/signal/update](http://localhost:80/id-report-manage/manage/hpInfectiousDisease/countTrendByQuarter)*

请求类型：*application/json*

请求方式：POST

接口备注：修改预警信号信息

调试工具：*ApiPost*

请求体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| id | 35 | int | 信号id |
| car\_id | 3 | int | 车架id |
| warn\_id | 1 | int | 规则编号 |
| cwsignal | "{\"Mx\":12.0,\"Mi\":0.6}" | varchar | 信号数据 |
| signal\_state | 1 | int | 信号处理状态 |

请求示例：

|  |
| --- |
| {  "id": 35,  "cwsignal": "{\"Mx\":12.0,\"Mi\":0.6}",  "car\_id": 3,  "warn\_id": 1,  "signal\_state": 1  } |

返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| success | true | bool | 调用接口成功 |
| message | 成功将数据库中的数据进行更改并进行redis的删除同步 | varchar | 返回接口信息 |
| code | 200 | int | 响应码 |
| data | 1 | int | 添加成功与否 |

### 查询电池信号

接口地址：

*[http://localhost:9000/signal/get](http://localhost:80/id-report-manage/manage/hpInfectiousDisease/countTrendByQuarter)*

请求类型：*application/json*

请求方式：POST

接口备注：查询预警信号信息

调试工具：*ApiPost*

请求体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| id | 35 | int | 信号id |

请求示例：

|  |
| --- |
| 参数名称：id 参数值：32 |

返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| success | true | bool | 调用接口成功 |
| message | 成功将数据库中的数据进行更改并进行redis的删除同步 | varchar | 返回接口信息 |
| code | 200 | int | 响应码 |
| data | {"id":32,"cwsignal":"{\"Mx\":12.0,\"Mi\":0.6}","car\_id":1,"warn\_id":1,"signal\_state": 1} | 预警对象 | 成功就返回对象 失败返回null |

### 删除电池信号

接口地址：

*[http://localhost:9000/signal/delete](http://localhost:80/id-report-manage/manage/hpInfectiousDisease/countTrendByQuarter)*

请求类型：*application/json*

请求方式：POST

接口备注：删除预警信号信息

调试工具：*ApiPost*

请求体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| id | 35 | int | 信号id |

请求示例：

|  |
| --- |
| 参数名称：id 参数值：32 |

返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| success | true | bool | 调用接口成功 |
| message | 成功将数据库中的数据进行删除并进行redis的删除同步 | varchar | 返回接口信息 |
| code | 200 | int | 响应码 |
| data | 1 | int | 删除成功与否 |

### 根据车架id查询电池信号

接口地址：

*[http://localhost:9000/signal/getByCarId](http://localhost:80/id-report-manage/manage/hpInfectiousDisease/countTrendByQuarter)*

请求类型：*application/json*

请求方式：POST

接口备注：根据车架id拆线呢电池信号

调试工具：*ApiPost*

请求体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| carId | 3 | int | 车架id |

请求示例：

|  |
| --- |
| 参数名称：carId参数值：3 |

返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| success | true | bool | 调用接口成功 |
| message | 已成功获取数据 | varchar | 返回接口信息 |
| code | 200 | int | 响应码 |
| data | [{"id":34,"cwsignal":"{\"Mx\":11.0,\"Mi\":9.6,\"Ix\":12.0,\"Ii\":11.7}","car\_id":3,"warn\_id":null,"signal\_state": 1}] | 预警信号list | 查询成功返回一个预警信号list 失败则为null |

### 根据车架id查询警告信号

接口地址：

*[http://localhost:9000/api/getMessageByCarId](http://localhost:80/id-report-manage/manage/hpInfectiousDisease/countTrendByQuarter)*

请求类型：*application/json*

请求方式：POST

接口备注：根据车架id查询警告信息

调试工具：*ApiPost*

请求体参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| carId | 3 | int | 车架id |

请求示例：

|  |
| --- |
| 参数名称：carId参数值：3 |

返回参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **示例值** | **类型** | **参数说明** |
| success | true | bool | 调用接口成功 |
| message | ok | varchar | 返回接口信息 |
| code | 200 | int | 响应码 |
| data | [{"id": 9,"car\_id": 3,"battery\_type": "三元电池","warn\_name": "电压差报警","warn\_level": 2,"signal\_id": 34}] | 警告信息list | 查询成功返回一个警告信息list 失败则为null |

## 4.单元测试

### 4.1 添加电池信号接口测试

Code: @Test

public void addtest(){

WarnSignalDto warnSignalDto=new WarnSignalDto();

warnSignalDto.setCarId(3);

warnSignalDto.setWarnId(1);

warnSignalDto.setSignal("{\"Mx\":12.0,\"Mi\":0.6}");

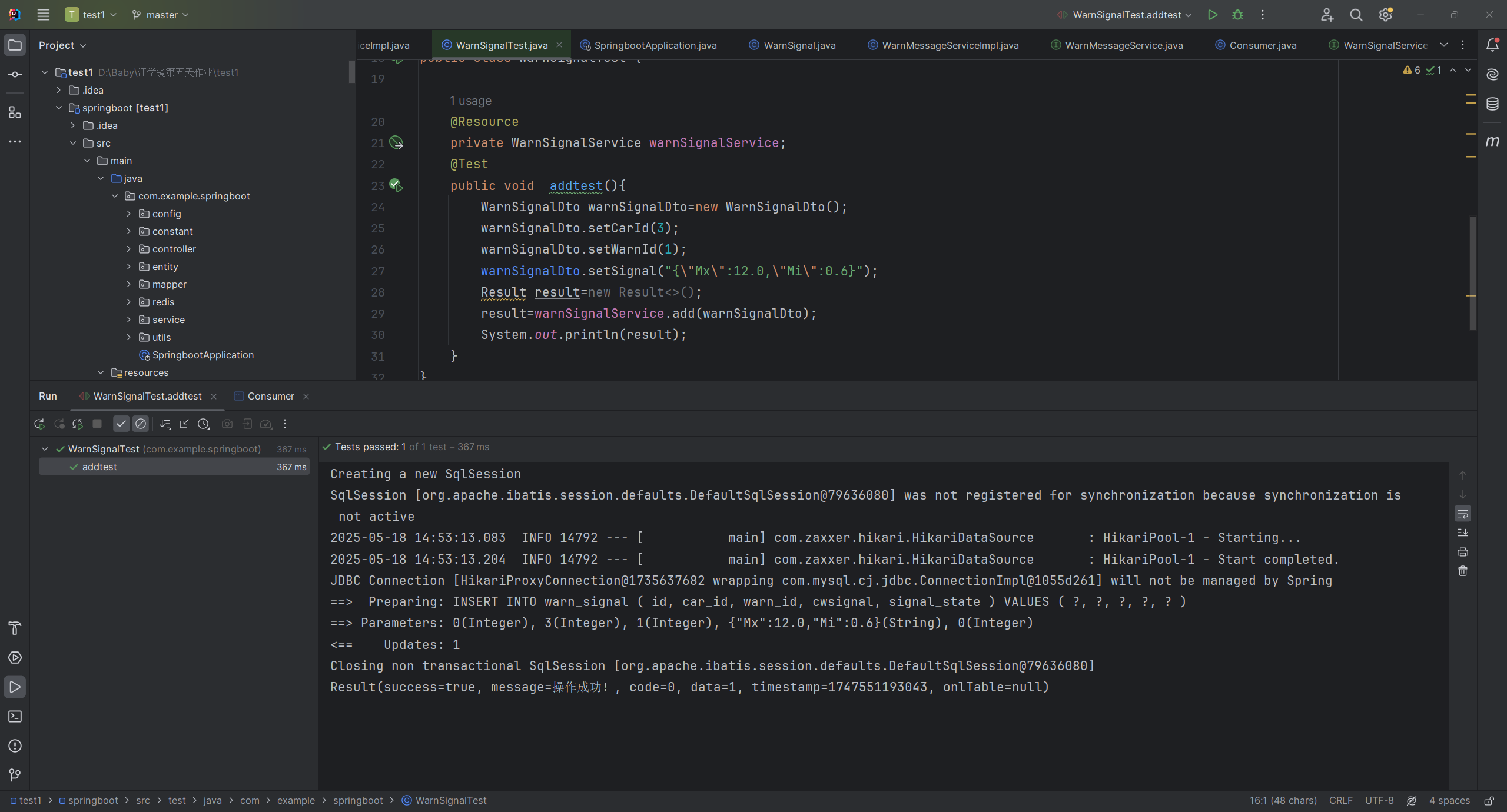
Result result=new Result<>();

result=warnSignalService.add(warnSignalDto);

System.out.println(result);

}

结果图片:



### 4.2 修改电池信号接口测试

Code: //测试修改接口

@Test

public void updatetest(){

WarnSignal warnSignal=new WarnSignal();

warnSignal.setCwsignal("{\"Mx\":12.0,\"Mi\":0.6}");

warnSignal.setWarnId(2);

warnSignal.setCarId(1);

warnSignal.setId(40);

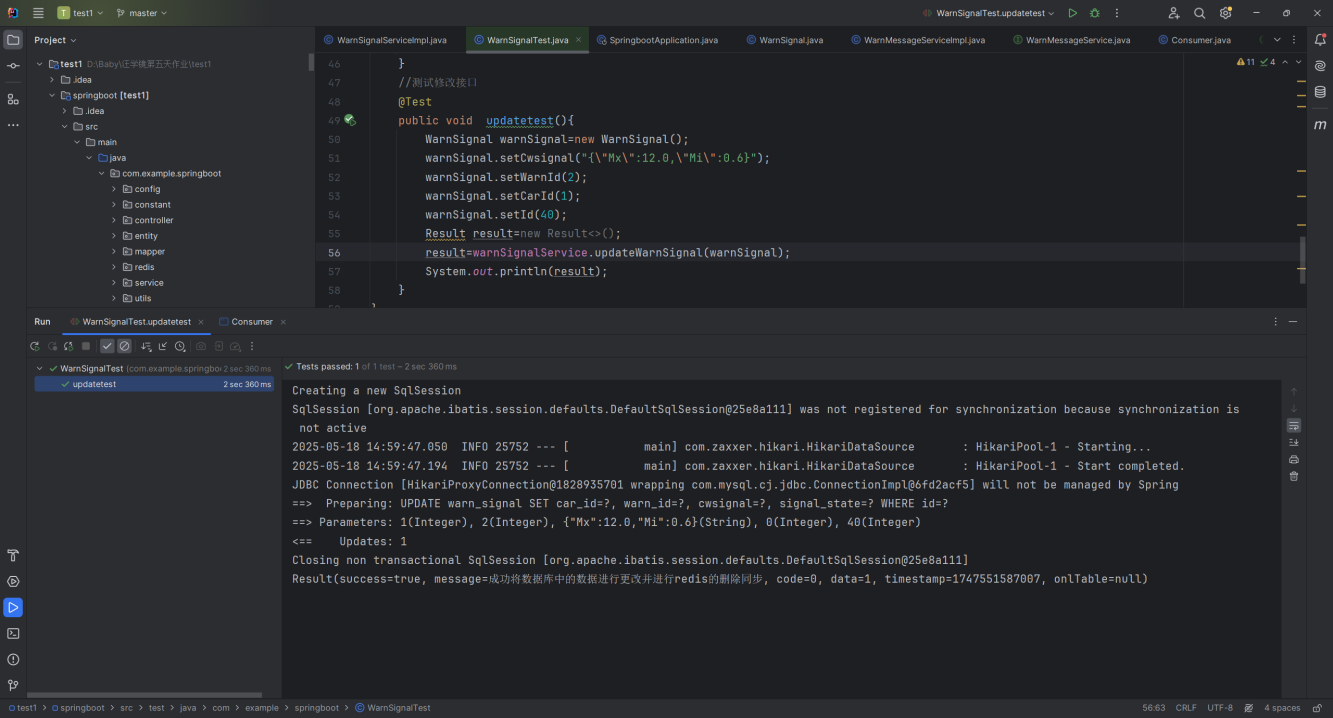
Result result=new Result<>();

result=warnSignalService.updateWarnSignal(warnSignal);

System.out.println(result);

}

结果图片：



### 4.3 查询电池信号接口测试

Code: @Test

public void gettest(){

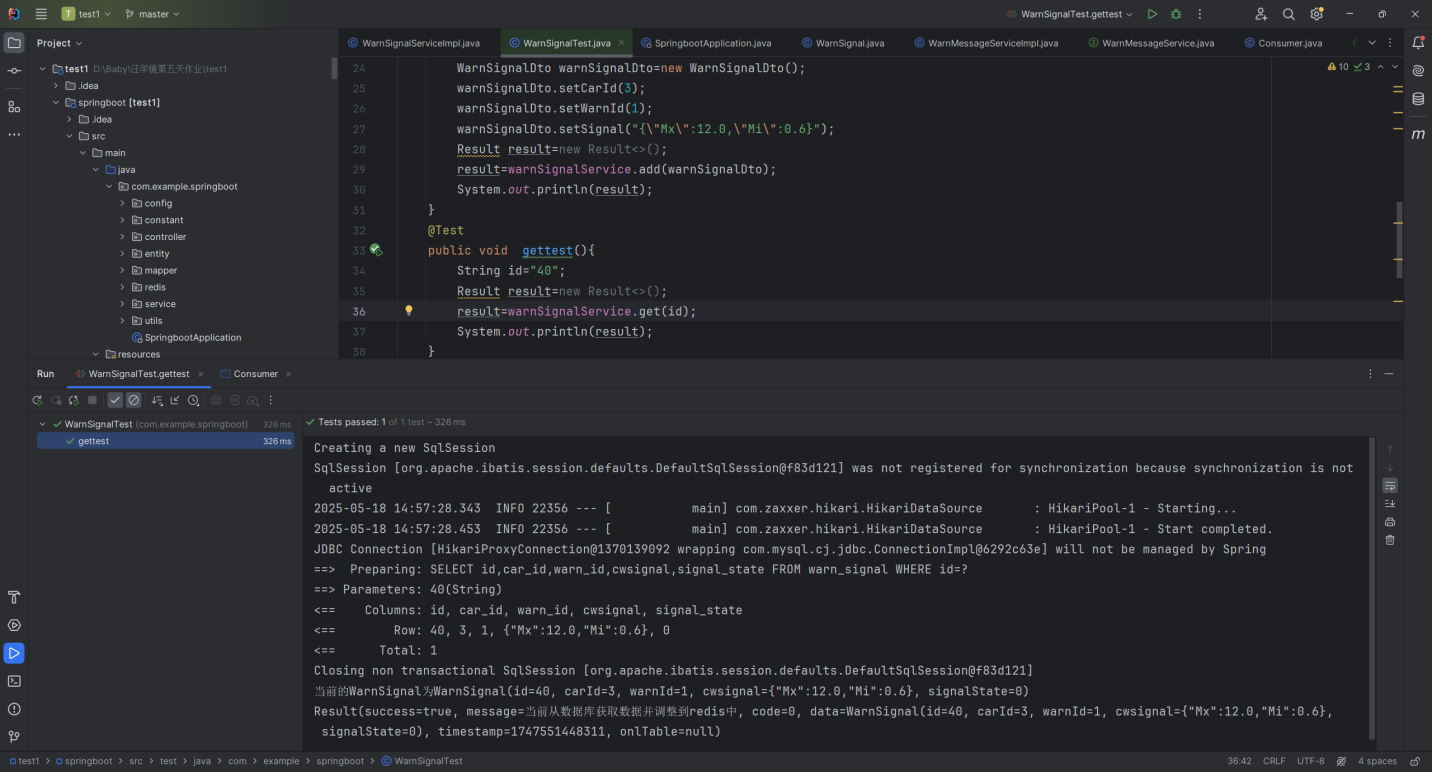
String id="40";

Result result=new Result<>();

result=warnSignalService.get(id);

System.out.println(result);

}

结果图片：

### 4.4 删除电池信号接口测试

Code: //测试删除接口

@Test

public void deletetest(){

String id="40";

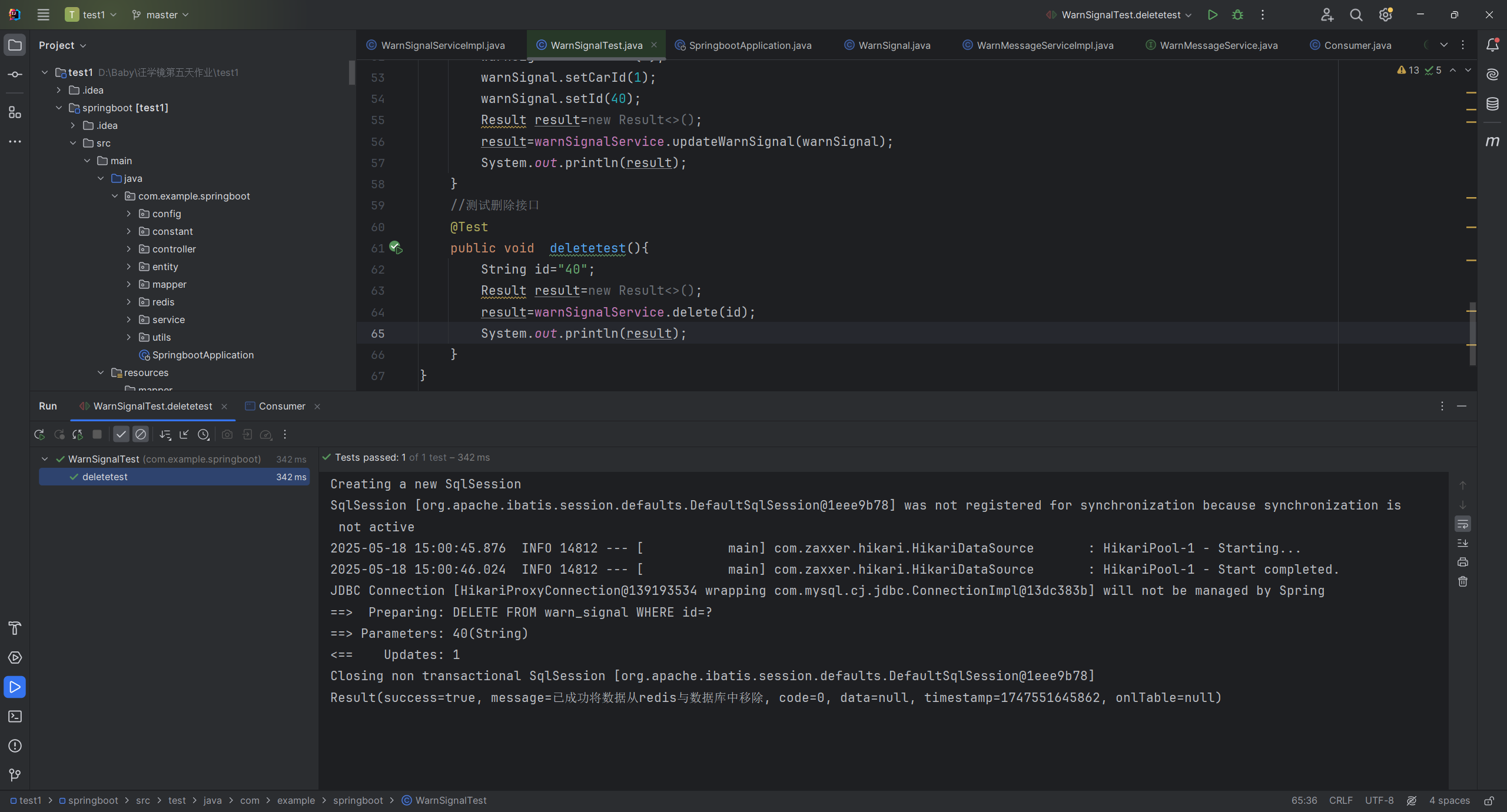
Result result=new Result<>();

result=warnSignalService.delete(id);

System.out.println(result);

}

结果图片：



### 4.5 根据车架id查询电池信号接口测试

Code: //测试根据车架id获取对应车架id的数据

@Test

public void getByCarIdtest(){

String carId="3";

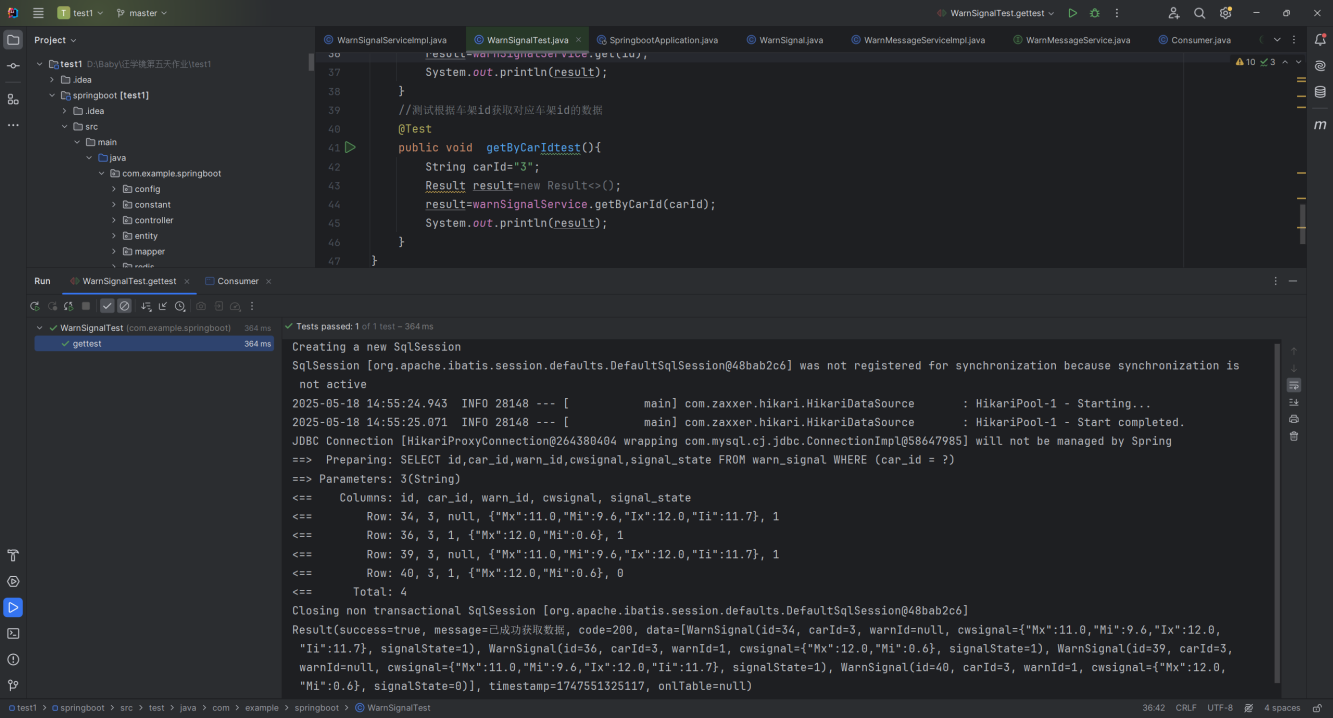
Result result=new Result<>();

result=warnSignalService.getByCarId(carId);

System.out.println(result);

}

结果图片：



### 4.6 根据车架id查询警告信息接口测试

Code: //测试删除接口

@Test

public void getMessageByCidtest(){

String id="3";

Result result=new Result<>();

result=warnMessageService.getMessageByCarId(id);

System.out.println(result);

}

结果图片：

