

ICS 35.240.50  
CCS X 07

# 团 体 标 准

T/FDSA 020—2021

## 重大活动食品安全情报 研判系统接口规范

Specification for interface of food safety information research and  
judgment system for major event

2021-11-30 发布

2021-12-30 实施

中国食品药品企业质量安全促进会 发布

中国食品药品企业质量安全促进会

全国团体标准信息平台

中国食品药品企业质量安全促进会

## 目 次

前言.....	II
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 缩略语.....	1
5 接口技术要求.....	1
6 接口调用方式.....	2
6.1 请求.....	2
6.2 返回.....	2
7 接口定义格式.....	2
8 接口原则.....	2
8.1 基本要求.....	2
8.2 公共原则.....	2
9 接口整体架构.....	2
10 数据接口.....	3
10.1 数据访问接口定义.....	3
10.2 数据新增、修改、删除、查询接口定义.....	4
11 业务接口.....	6
11.1 食品合格预测接口定义.....	6
11.2 食品中毒害危险物最大限量值预警接口定义.....	8
11.3 食品安全风险图神经网络预警接口定义.....	9
11.4 食品安全知识图谱问答系统接口定义.....	10
11.5 基于 LSTM 的毒害危险物风险预测接口定义.....	11
附录 A （规范性） 接口返回代码值.....	14
附录 B （资料性） 接口调用示例.....	15

## 前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由北方工业大学提出。

本文件由中国食品药品企业质量安全促进会归口。

本文件起草单位：北方工业大学、中国人民公安大学、北京工业大学、江苏华安博远检测技术有限公司。

本文件主要起草人：史运涛、李书钦、董哲、王力、李志军、田华伟、李晓理、王康、蒋维嘉、李春雷、殷翔、周萌、李杰、丁瑜。

# 重大活动食品安全情报研判系统接口规范

## 1 范围

本规范规定了重大活动食品安全情报研判系统的接口技术要求、接口调用方式、接口定义格式、接口原则、接口整体架构、数据接口和业务接口。

本规范适用于重大活动食品安全情报研判系统的建设与管理。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 5271.9 信息技术词汇 第9部分：数据通信

## 3 术语和定义

GB/T 5271.9界定的以及下列术语和定义适用于本文件。

### 3.1

**重大活动** major event

具有一定社会影响力、需要公共卫生监督和警务保障的各项高规格的政治、经济、文化、体育等活动。

## 4 缩略语

下列缩略语适用于本文件。

HTTP：超文本传输协议（Hypertext Transfer Protocol）

JDBC：Java数据库连接（Java Database Connectivity）

JSON：JavaScript对象标记（JavaScript Object Notation）

LSTM：长短期记忆网络（Long-Short Term Memory）

SQL：结构化查询语言（Structured Query Language）

URL：统一资源定位符（Uniform Resource Locators）

XML：可扩展置标语言（eXtensible Markup Language）

## 5 接口技术要求

毒害危险物数据库、案例数据库、第三方数据库及实时监测数据应通过本文件定义的通用访问接口，为情报研判系统软件提供数据访问和管理功能，其技术要求如下：

——毒害危险物数据库应提供农药、兽药、添加剂、生物毒素、微生物、污染物、致癌物、化学危险品、非法添加物等数据的高效存取功能，支持情报实时分析研判；

- 案例数据库应提供刑事案例数据的高速访问功能，支持对案发时间、案发地点、受害人、案件信息等在线分析研判；
- 第三方数据库应提供重点人员、社交媒体、电商平台、物流、话单、通行、交易等数据的快速查询功能，支持对舆情演化、电商交易等数据的分析计算；
- 实时监测数据应提供现场快检设备检测数据、重大活动食品安全相关事件信息、辅助决策系统推送的情报信息等实时动态数据。

## 6 接口调用方式

### 6.1 请求

应通过HTTP请求格式调用数据库服务接口，并将请求参数包含在请求的URL中。

### 6.2 返回

调用成功后应以XML或JSON格式返回结果，调用失败时返回相应的错误信息。

## 7 接口定义格式

情报研判系统的接口定义如下：

- 接口名：接口的名称；
- 描述：对接口内容的描述；
- 参数：请求调用的参数；
- 返回值：请求返回的内容；
- 错误信息：调用错误时返回的信息。

## 8 接口原则

### 8.1 基本要求

系统间接口应满足下列基本要求：

- 在系统的高并发和大容量的基础上提供安全可靠的接入；
- 提供接入有效性校验机制，如验证参数个数、类型、值域范围等；
- 接口设计遵循可扩展原则；
- 接口设计提供调用安全校验机制，以确保接口的安全性。

### 8.2 公共规则

所有接口应满足下列规则：

- 接口统一采用基于 HTTP 1.1 协议的消息交互；
- HTTP 方法兼容 HTTP 1.1 协议，统一采用 POST 方法；
- 请求和应答统一采用 UTF-8 编码方式；
- 接口返回代码采用统一代码表示，见附录 A。

## 9 接口整体架构

情报研判系统接口架构见表1。

表1 情报研判系统接口架构

接口类型	接口名
数据接口	数据访问
	数据新增
	数据修改
	数据删除
	数据查询
业务接口	食品合格预测
	食品中毒害危险物最大限量值预警
	食品安全风险图神经网络预警
	食品安全知识图谱问答系统
	基于LSTM的毒害危险物风险预测

## 10 数据接口

### 10.1 数据访问接口定义

#### 10.1.1 接口描述

情报研判系统采用JDBC访问数据库，使用JDBC访问数据库的基本步骤如下：

- 加载 JDBC 驱动程序；
- 建立数据库连接；
- 创建操作数据库 SQL 的对象；
- 执行语句并分析执行结果；
- 关闭连接。

#### 10.1.2 接口参数描述

情报研判系统通过 JDBC 访问 MySQL 数据库的 Url 参数描述见表 2。

表2 数据访问接口参数描述

参数名称	参数说明
user	数据库用户名，用于连接数据库
password	用户密码，用于连接数据库
useUnicode	是否使用Unicode字符集，如果参数characterEncoding设置为gb2312、gbk或utf8，本参数的值必须设置为true
characterEncoding	useUnicode为true时，指定字符编码，比如可以设置为gb2312、gbk或utf8
autoReconnect	当数据库连接异常中断时，是否重新连接
autoReconnectForPools	是否使用针对数据库连接池的重连策略
failOverReadOnly	自动重连成功后，连接是否设置为只读
maxReconnects	autoReconnect设置为true时，重试连接的次数
initialTimeout	autoReconnect设置为true时，两次重连之间的时间间隔，单位：秒

表2 (续)

参数名称	参数说明
connectTimeout	和数据库服务器建立socket连接时的超时，单位：毫秒。0表示永不超时，适用于JDK 1.4及更高版本
socketTimeout	网络套接字连接的超时（默认值0表示无超时）

示例：

情报研判系统通过JDBC访问MySQL数据库：

```
jdbc:mysql://localhost:3306/foodDb?user=root&password=&useUnicode=true&characterEncoding=gbk&autoReconnect=true&failOverReadOnly=false
```

### 10.1.3 接口调用规则

调用时序如图1所示。

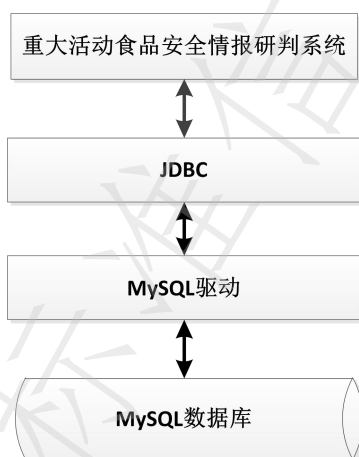


图1 数据访问接口时序图

### 10.1.4 接口调用

使用Java语言来进行接口调用，调用示例见附录B（示例1）。

## 10.2 数据新增、修改、删除、查询接口定义

### 10.2.1 接口描述

#### 10.2.1.1 新增

数据新增接口描述见表3。

表3 数据新增接口描述

接口描述	说明
函数名称	save
功能描述	在毒害危险物数据库、案例数据库中新增数据
前置条件	为对应的数据对象实体赋值
输入参数	数据对象实体
输出参数	1) 处理结果
	2) 异常描述

### 10.2.1.2 修改

数据修改接口描述见表4。

表4 数据修改接口描述

接口描述	说明
函数名称	update
功能描述	在毒害危险物数据库、案例数据库中修改数据
前置条件	为对应的数据对象实体赋值
输入参数	数据对象实体
输出参数	1) 处理结果
	2) 异常描述

### 10.2.1.3 删除

数据删除接口描述见表5。

表5 数据删除接口描述

接口描述	说明
函数名称	delete
功能描述	在毒害危险物数据库、案例数据库中删除数据
前置条件	取得要删除数据对象的id
输入参数	数据对象实体的id
输出参数	1) 处理结果
	2) 异常描述

### 10.2.1.4 查询

数据查询接口描述见表6。

表6 数据查询接口描述

接口描述	说明
函数名称	show
功能描述	在毒害危险物数据库、案例数据库、第三方数据库、实时监测数据中查询数据
前置条件	取得要查询数据对象的id
输入参数	数据对象实体的id
输出参数	1) 处理结果
	2) 异常描述

## 10.2.2 接口调用规则

调用时序如图2所示。

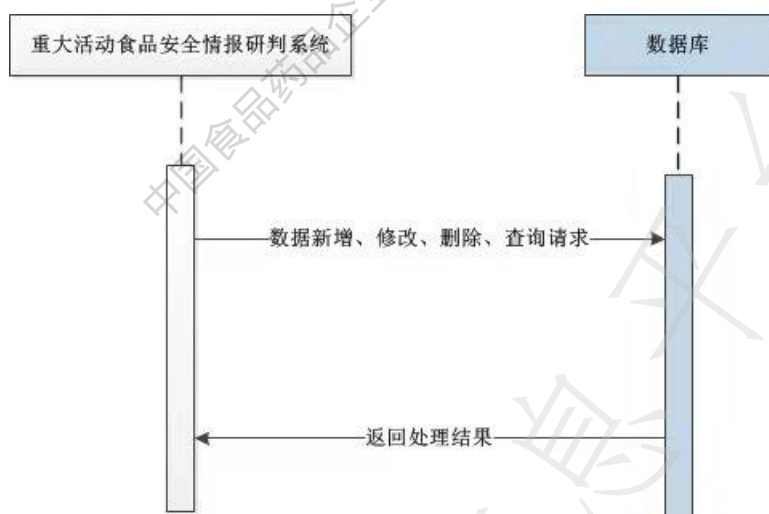


图 2 数据接口时序图

### 10.2.3 接口调用

使用Java语言来进行接口调用，接口调用示例见附录B（示例2、示例3、示例4、示例5）。

## 11 业务接口

### 11.1 食品合格预测接口定义

#### 11.1.1 接口描述

根据某段时间内某一食品的抽检数据，输入任一批次的食品特征数据，预测其是否合格。食品合格预测接口描述见表7。

表 7 食品合格预测接口描述

接口描述	说明
函数名称	forecast
接口方式	HttpPost
接口协议	HttpPost
功能描述	输入任一批次的食品特征数据，预测其是否合格
前置条件	食品合格预测模型已部署
输入参数	1) 食品亚类
	2) 规格
	3) 生产时间
	4) 企业类型
	5) 生产省份
	6) 异地运输
输出参数	1) 预测结果
	2) 模型信息

## 11.1.2 接口参数描述

食品合格预测接口参数描述见表8。

表8 食品合格预测接口参数描述

参数名称	参数说明	输入/输出	参数类型	取值范围	是否必须	参数示例
foodCategory	食品亚类	输入	string	0~50个字符	是	食品亚类/分类
specification	规格	输入	string	0~50个字符	是	散装称重/预包装
productionTime	生产时间	输入	string	0~50个字符	是	第一季度/第二季度/第三季度/第四季度
enterpriseType	企业类型	输入	string	0~50个字符	是	公司/店铺
produceProvince	生产省份	输入	string	0~50个字符	是	所有省份
longDistanceTransportation	异地运输	输入	string	0~50个字符	是	是/否
result	预测结果	输出	string	0~50个字符	是	合格/不合格
modelInfo	模型信息	输出	string	1000个字符	是	模型名称, 训练集精度, 测试集精度, 正确率, 精准率, 召回率, 调和平均值, 权重优化器, 激活函数, 隐层数量及神经元个数

**示例:**

根据表8形成的输出参数JSON:

```
{ "result": "合格", "modelInfo": { "模型名称": "BP神经网络", "训练集精度": 92.64, "测试集精度": 86.90, "正确率": 86.90, "精准率": 90.31, "召回率": 89.25, "调和平均值": 89.78, "权重优化器": "adam", "激活函数": "tanh", "隐层数量及神经元个数": 80 } }
```

## 11.1.3 接口调用规则

调用时序如图3所示。

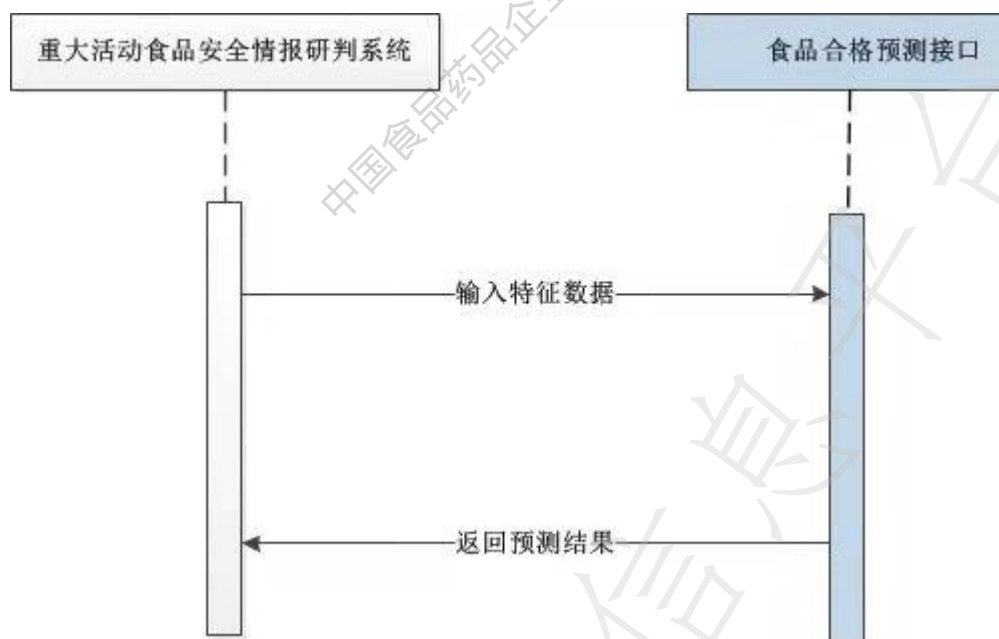


图3 食品合格预测接口时序图

#### 11.1.4 接口调用

使用Java语言来进行接口调用。调用示例见附录B（示例6）。

### 11.2 食品中毒害危险物最大限量值预警接口定义

#### 11.2.1 接口描述

根据某段时间内食品中毒害危险物含量数据，得到线性回归方程，输入任一批次的食品中毒害危险物监测含量数据，预测毒害危险物达到其最大限量值的时间。

食品中毒害危险物最大限量值预警接口描述见表9。

表9 食品中毒害危险物最大限量值预警接口描述

接口描述	说明
函数名称	findFoodC4DataByQueryData
接口方式	HttpPost
接口协议	HttpPost
功能描述	输入任一批次的食品中毒害危险物监测含量数据，预测毒害危险物达到其最大限量值的时间
前置条件	食品安全风险毒害危险物最大限量值预警模型已部署
输入参数	1) 食品名称 2) 毒害物名称 3) 检测含量
输出参数	预测结果

### 11.2.2 接口参数描述

食品中毒害危险物最大限量值预警接口参数描述见表10。

表 10 食品中毒害危险物最大限量值预警接口参数描述

参数名称	参数说明	输入/输出	参数类型	取值范围	是否必须	参数示例
foodName	食品名称	输入	string	0~50个字符	是	酱油
poisonName	毒害物名称	输入	string	0~50个字符	是	3-氯丙醇
productionTime	检测含量	输入	int	0~50000	是	0.4
result	预测结果	输出	date	10位	是	2021-05-16

示例:

根据表10形成的输出参数JSON:

```
{"result":2021-05-16}
```

### 11.2.3 接口调用

使用Java语言来进行接口调用。调用示例见附录B（示例7）。

## 11.3 食品安全风险图神经网络预警接口定义

### 11.3.1 接口描述

根据某段时间内的食品抽检数据，输入任一种食品的特征数据，预测其风险等级。食品安全风险图神经网络预警接口描述见表11。

表 11 食品安全风险图神经网络预警接口描述

接口描述	说明
函数名称	Han
接口方式	HttpPost
接口协议	HttpPost
功能描述	输入任一种食品的特征数据，预测其风险等级
前置条件	食品安全风险图神经网络预警模型已部署
输入参数	1) 食品名称
	2) 包装类型
	3) 生产时间
	4) 生产企业类型
	5) 同异地
	6) 抽检级别
	7) 生产省份
	8) 检测毒害物
输出参数	预测结果

## 11.3.2 接口参数描述

食品安全风险图神经网络预警接口参数描述见表12。

表 12 食品安全风险图神经网络预警接口参数描述

参数名称	参数说明	输入/输出	参数类型	取值范围	是否必须	参数示例
name	食品名称	输入	string	0~50个字符	是	食品名称
specifications	包装类型	输入	string	0~50个字符	是	散装称重/预包装
productionTime	生产时间	输入	string	0~50个字符	是	第一季度/第二季度/第三季度/第四季度
enterpriseType	企业类型	输入	string	0~50个字符	是	公司/工厂/超市/店铺
offSite	同/异地	输入	string	0~50个字符	是	同地/异地
samplingLevel	抽检等级	输入	string	0~50个字符	是	国抽/省抽
province	生产省份	输入	string	0~50个字符	是	各省份名称
hazard	检测毒害危险物	输入	string	0~50个字符	是	各食品毒害危险物
result	预测结果	输出	string	0~50个字符	是	低风险/较低风险/中等风险/较高风险/高风险

## 11.3.3 接口调用

使用Java语言来进行接口调用。调用示例见附录B（示例8）。

## 11.4 食品安全知识图谱问答系统接口定义

## 11.4.1 接口描述

输入与食品安全事件相关的问题，返回答案。食品安全知识图谱问答系统接口描述见表 13。

表 13 食品安全知识图谱问答系统接口描述

接口描述	说明
函数名称	KGQA_answer
接口方式	HttpPost
接口协议	HttpPost
功能描述	输入与食品安全事件相关的问题，返回答案
前置条件	食品安全知识图谱问答系统已部署
输入参数	食品安全事件相关的问题
输出参数	返回cypher语句

### 11.4.2 接口参数描述

食品安全知识图谱问答系统接口参数描述见表14。

表 14 食品安全知识图谱问答系统接口参数描述

参数名称	参数说明	输入/输出	参数类型	取值范围	是否必须	参数示例
name	问题	输入	string	0~50个字符	是	与食品安全相关的问题
result	问题答案	输出	string	0~100个字符	是	返回答案的cypher语句

### 11.4.3 接口调用

使用Java语言来进行接口调用。调用示例见附录B（示例9）。

## 11.5 基于 LSTM 的毒害危险物风险预测接口定义

### 11.5.1 接口描述

以周为单位，根据前十周任一种食品的毒害危险物含量来预测未来一周该食品中毒害危险物的风险。基于LSTM的毒害危险物风险预测接口描述见表15。

表 15 基于 LSTM 的毒害危险物风险预测接口描述

接口描述	说明
函数名称	LstmForecast
接口方式	HttpPost
接口协议	HttpPost
功能描述	输入前十周任一种食品的毒害危险物含量来预测未来一周该食品中毒害危险物的风险。
前置条件	食品中毒害危险物含量风险预测预警模型已部署
输入参数	1) 第一周某食品的检测毒害危险物含量
	2) 第二周某食品的检测毒害危险物含量
	3) 第三周某食品的检测毒害危险物含量
	4) 第四周某食品的检测毒害危险物含量
	5) 第五周某食品的检测毒害危险物含量
	6) 第六周某食品的检测毒害危险物含量
	7) 第七周某食品的检测毒害危险物含量
	8) 第八周某食品的检测毒害危险物含量
	9) 第九周某食品的检测毒害危险物含量
	10) 第十周某食品的检测毒害危险物含量
输出参数	1) 预测结果
	2) 模型信息

## 11.5.2 接口参数描述

基于 LSTM 的毒害危险物风险预测接口参数描述见表 16。

表 16 基于 LSTM 的毒害危险物风险预测接口参数描述

参数名称	参数说明	输入/输出	参数类型	取值范围	是否必须	参数示例
firstweek	第一周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第一周某食品的检测毒害危险物含量
secondweek	第二周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第二周某食品的检测毒害危险物含量
thirdweek	第三周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第三周某食品的检测毒害危险物含量
forthweek	第四周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第四周某食品的检测毒害危险物含量
fifthweek	第五周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第五周某食品的检测毒害危险物含量
sixthweek	第六周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第六周某食品的检测毒害危险物含量
seventhweek	第七周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第七周某食品的检测毒害危险物含量
eighthweek	第八周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第八周某食品的检测毒害危险物含量
ninthweek	第九周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第九周某食品的检测毒害危险物含量
tenthweek	第十周某食品的检测毒害危险物含量	输入	float	国标以内或适量超标	是	第十周某食品的检测毒害危险物含量
result	预测结果	输出	string	0到5以内的风险值	是	预测结果
modelinfo	模型参数	输出	string	1000个字符	是	模型名称，输入步长，输出步长，网络层数，训练轮次

示例：

根据表24形成的输出参数JSON：

```
{
  "result": "1.86",
  "model_info": {
    "模型名称": "LSTM",
    "输入步长": "10",
    "输出步长": "1",
    "网络层数": "4",
    "训练轮次": "100"
  }
}
```

### 11.5.3 接口调用

使用 Java 语言来进行接口调用。调用示例见附录 B（示例 10）。

附录 A  
(规范性)  
接口返回代码值

接口返回代码如表A.1所示。

表 A.1 接口返回异常代码说明

错误码 (Code)	异常描述 (ErrorMessage)	说明
0	正常	Code=0, 接口调用成功
2	签名过期	TimeStamp过期, API服务端允许客户端请求时间误差为5分钟
8	无效时间戳	TimeStamp格式错误, TimeStamp格式: 北京时间YYYY-MM-DD hh:mm:ss
12	无效签名	Sign错误
19	无效Token	Token错误
21	无效appKey	appKey错误
31	业务参数错误	一般是由于调用方输入的参数有误引起, 应根据响应内容Data报错信息调整参数内容
110	其他未知错误	其他未知错误

## 附录 B

### (资料性)

### 接口调用示例

本附录提供了接口调用的示例。以下示例包括：

- 示例 1：请求接口；
- 示例 2：新增接口；
- 示例 3：修改接口；
- 示例 4：删除接口；
- 示例 5：查询接口；
- 示例 6：食品合格预测接口（以肉及肉制品为例）；
- 示例 7：食品中毒害危险物最大限量值预警接口；
- 示例 8：食品安全风险图神经网络预警接口；
- 示例 9：食品安全知识图谱问答系统接口；
- 示例 10：基于 LSTM 的毒害危险物风险预测接口。

#### 示例1：请求接口

```
datasource:
  dynamic:
    hikari:
      connection-timeout: 30000
      max-lifetime: 1800000
      max-pool-size: 15
      min-idle: 5
      pool-name: FfHikariCP
    primary: base
  datasource:
    base:
      driver-class-name: com.mysql.cj.jdbc.Driver
      url: jdbc:mysql://127.0.0.1:3306/spaqqbdb?useUnicode=true&characterEncoding=UTF-8&serverTimezone=Asia/Shanghai&useSSL=false
      username: root
      password: 123456
      connection-test-query: select 1
    p6spy: true
```

#### 示例2：新增接口

```
public AjaxJson save(DhwFftjwEntity dhwFftjwEntity, String roleIds) {
    AjaxJson j = new AjaxJson();
    try{
        //后端校验
        if(StringUtils.isEmpty(dhwFftjwEntity.getSsdl())) {
```

```

        j.setSuccess(false);
        j.setMsg("非法添加物大类不能为空");
        return j;
    }
    if(StringUtils.isEmpty(dhwFftJwEntity.getSsxl())) {
        j.setSuccess(false);
        j.setMsg("非法添加物小类不能为空");
        return j;
    }
    if(StringUtils.isEmpty(dhwFftJwEntity.getCname())) {
        j.setSuccess(false);
        j.setMsg("非法添加物名称不能为空");
        return j;
    }
    dhwFftJwService.save(dhwFftJwEntity);
} catch (Exception e) {
    log.error("保存用户信息报错, 错误信息:" + e.getMessage());
    j.setSuccess(false);
    e.printStackTrace();
}
return j;
}

```

**示例3: 修改接口**

```

public AjaxJson update(DhwFftJwEntity dhwFftJwEntity, String roleIds) {
    AjaxJson j = new AjaxJson();
    try{
        //后端校验
        if(StringUtils.isEmpty(dhwFftJwEntity.getSsdl())) {
            j.setSuccess(false);
            j.setMsg("非法添加物大类不能为空");
            return j;
        }
        if(StringUtils.isEmpty(dhwFftJwEntity.getSsxl())) {
            j.setSuccess(false);
            j.setMsg("非法添加物小类不能为空");
            return j;
        }
        if(StringUtils.isEmpty(dhwFftJwEntity.getCname())) {
            j.setSuccess(false);
            j.setMsg("非法添加物名称不能为空");
            return j;
        }
    }
}

```

```

        DhwFftjwEntity tmp = dhwFftjwService.getById(dhwFftjwEntity.getId());
        MyBeanUtils.copyBeanNotNull2Bean(dhwFftjwEntity, tmp);
        dhwFftjwService.update(tmp);
    } catch (Exception e) {
        log.error("保存用户信息报错, 错误信息:" + e.getMessage());
        j.setSuccess(false);
        e.printStackTrace();
    }
    return j;
}

```

**示例4: 删除接口**

```

public AjaxJson delete(DhwFftjwEntity dhwFftjwEntity) {
    AjaxJson j = new AjaxJson();
    try {
        if (null != dhwFftjwEntity.getId()) {
            dhwFftjwService.removeById(dhwFftjwEntity.getId());
        }
    } catch (Exception e) {
        log.error("删除非法添加物信息出错, 错误信息:" + e.getMessage());
        j.setSuccess(false);
        e.printStackTrace();
    }
    return j;
}

```

**示例5: 查询接口**

```

public String show(DhwFftjwEntity dhwFftjwEntity, Model model) {
    DhwFftjwEntity fftjwEntity = dhwFftjwService.getById(dhwFftjwEntity.getId());
    model.addAttribute("fftjwEntity", fftjwEntity);
    return "dhw/fftjw/fftjwshow.html";
}

```

**示例6: 食品合格预测接口 (以肉及肉制品为例)**

```

public MeatResult forecastMeat(Meat meat) throws IOException {
    HttpEntity responseEntity = HttpUtils.post(meatUrl, JSON.toJSONString(meat));
    String response = EntityUtils.toString(responseEntity);
    System.out.println(response);
    MeatResult result = JSON.parseObject(response, MeatResult.class);
    return result;
}

```

**示例7: 食品中毒害危险物最大限量值预警接口**

```

public List<FoodC4Data> findFoodC4DataByQueryData(FoodC4Query foodC4Query) throws IOException {
    HttpEntity responseEntity = HttpUtils.post(foodC4Url, JSON.toJSONString(foodC4Query));
    String response = EntityUtils.toString(responseEntity);
}

```

```

FoodC4Result foodC4Result= JSON.parseObject(response, FoodC4Result.class);
List<FoodC4Data> foodC4DataList=new ArrayList<FoodC4Data>();
if(foodC4Result.getId()>0){
    QueryWrapper<FoodC4Data> queryWrapper=new QueryWrapper<FoodC4Data>();
    queryWrapper.orderByAsc("jcri");
    foodC4DataList=this.list(queryWrapper);
}
return foodC4DataList;
}

```

示例8：食品安全风险图神经网络预警接口

```

public HANResult forecastHAN(HAN han) throws IOException {
    HttpEntity responseEntity = HttpUtils.post(hanUrl, JSON.toJSONString(han));
    String response = EntityUtils.toString(responseEntity);
    System.out.println(response);
    HANResult result= JSON.parseObject(response, HANResult.class);
    return result;
}

```

示例9：食品安全知识图谱问答系统接口

```

public Iterable<Map<String, Object>> queryAnswer(String question) throws IOException {
    HashMap<String, String>params=new HashMap<>();
    params.put("name", question);
    HttpEntity responseEntity = HttpUtils.post(url, JSON.toJSONString(params));
    String response = EntityUtils.toString(responseEntity);
    System.out.println(response);
    HashMap<String, String> result= JSON.parseObject(response, HashMap.class);
    String sql=result.get("sql");
    if(sql!=null){
        Session session = TransactionUtils.getSession(sessionFactory);
        return session.query(sql, new HashMap<>());
    }
    return new ArrayList<>();
}

```

示例10：基于LSTM的毒害危险物风险预测接口

```

public LSTMResult forecastLSTM (LSTM lstm) throws IOException {
    HttpEntity responseEntity = HttpUtils.post(lstmUrl, JSON.toJSONString(han));
    String response = EntityUtils.toString(responseEntity);
    System.out.println(response);
    LSTMResult result= JSON.parseObject(response, LSTMResult.class);
    return result;
}

```