**食品监管平台**

**仪器检测数据上传接口说明**

|  |  |
| --- | --- |
| **版本编号：** | V1.0 |
| **发布时间：** | 2019-02-25 |
|  |  |

**【** 历次版本修订记录，版本修订描述重要说明**】**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 日期 | 版本号 | 版本修订描述 | 拟制/修订人 | 审核人 | 批准人 |
| 2019-4-05 | V1.0 | 起草，确定平台、仪器的对接流程 | 赵智 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 前言

## 编写目的

监管平台为了实现与终端检测仪器软件进行数据传输，实现被检对象信息和检测结果数据的跨平台对接，特编写接口设计说明，以便为终端数据对接平台作为指导。

## 名词定义

**HTTP**：是互联网上应用最为广泛的一种网络协议，是一个客户端和服务器端请求和应答的标准（TCP），用于从WWW服务器传输超文本到本地浏览器的传输协议，它可以使浏览器更加高效，使网络传输减少。

**WebService**：是一个平台独立的，低耦合的，自包含的、基于可编程的web的应用程序，用于开发分布式的互操作的应用程序，很方便的实现数据跨平台使用。

**JSon**：JavaScript Object Notation, 是一种轻量级的数据交换格式。它基于 ECMAScript (w3c制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

# 接口设计

## 设计要求

1. 数据传输方式：http；
2. 接口采用：WebService；
3. 返回的数据格式：JSon；
4. 用户通过验证后，生成一个唯一身份标识token；
5. 调用接口必传唯一身份标识token；
6. 接口参数和返回结果，都应该要保留一定的扩展能力，建议每个保留3-5个参数可扩展；
7. 返回码，代表指定错误信息；

## 接口地址

真实WebService的WSDL地址:

<http://39.96.163.128:8089/food/services/DatRockService?wsdl>

# 接口

## 用户验证接口

**接口名称：getUser**

请求实例：getUser(String username,String password)

**请求参数说明**:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **备注** |
| username | String | 是 | 用户名 |
| password | String | 是 | 密码（明码） |

返回值json：

{

"id": "1",

"code": "1000",

"msg": "操作成功",

" token": "3245434533"

}

返回json参数说明：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| Id | String | 用户ID（唯一标识） |
| code | String | 结果代码，代码定义详见：附-返回代码表 |
| msg | String | 结果提示信息：通信成功！或 通信失败：密码错误！或用户不存在等 |
| token | String | 用户验证唯一标识 |

## 市场信息（项目部）接口

**接口名称：getDepartment**

请求实例：**getDepartment**(String token)

**请求参数说明**:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **备注** |
| token | String | 是 | 用户登录返回的唯一码 |

注：返回的市场信息为所有市场信息。

返回值json：

{

"code":"1000",

"count":"2",

"msg":"操作成功",

"jigougl":[

{

"id":1,

"jgmc":"江苏农贸市场"

},

{

"id":2,

"jgmc":"江苏果菜市场"

}

]

}

返回json参数说明：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **备注** |
| code | String | 结果代码，代码定义详见：附-返回代码表 |
| msg | String | 结果提示信息：操作成功！ |
| count | String | 数量 |
| jigougl | String | 项目部 |
| id | String | 项目部ID（唯一标识） |
| jgmc | String | 项目部名称 |

## 市场信息（被检单位）接口

**接口名称：getCompany**

请求实例：**getCompany** (String id,String token)

**请求参数说明**:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **备注** |
| id | String | 是 | 项目部ID（唯一标识） |
| token | String | 是 | 用户登录返回的唯一码 |

注：返回所选项目部下的所有被检单位。

返回值json：

{

"code":"1000",

"count":"40",

"msg":"操作成功",

"company":[

{

"bjcdwmc":"xxx单位",

"id":1

},

{

"bjcdwmc":"xxx单位",

"id":2

}

]

}

返回json参数说明：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **备注** |
| code | String | 结果代码，代码定义详见：附-返回代码表 |
| msg | String | 结果提示信息：操作成功！ |
| count | String | 数量 |
| bjcdwmc | String | 被检单位名称 |
| id | String | 被检单位id（唯一标识） |

## 市场信息（摊位号）接口

**接口名称：getBooth**

请求实例：**getBooth**(String id,String token)

**请求参数说明**:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **数据类型** | **必填** | **备注** |
| id | String | 是 | 被检单位ID（唯一标识） |
| token | String | 是 | 用户登录返回的唯一码 |

返回值json：

{

"code":"1000",

"count":"2",

"msg":"操作成功",

"booth":

[

{

"id":3,

"twhmc":"XXX摊位号"

},

{

"id":4,

"twhmc":"XXX摊位号"

}

]

}

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **备注** |
| code | String | 结果代码，代码定义详见：附-返回代码表 |
| msg | String | 结果提示信息：操作成功！ |
| count | String | 数量 |
| id | String | 摊位号ID（摊位号唯一标识） |
| twhmc | String | 摊位号名称 |

## 上传数据接口

**接口名称：**uploadDataDock

请求实例：String uploadDataDock(string data，string token)

注：token为通信测试返回的验证唯一码， data是上传数据拼成的json字符串，如下

[

{

“CheckID”:”201902170907221001”,

“department”:”1”,

“detComp”:”1”,

”boothNumber”:”1”,

”sampNumber”:”样品编号”,

”category”:”样品分类”,

”sampleName”:” 样品名称”,

”testItem”:”检测项目”,

”standardVal”:”标准值”,

”detection”:”检测数值”,

“detectionCom”:”检测值单位”,

“detResult”:”检测结果”,

“detGist”:”检测依据”,

“detDate”:”检测时间”,

“jcshi”:”检测室”,

“inspector”:”1”

},

{多条的话重复}

]

**参数说明：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名称 | 参数含义 | 参数类型 | 是否必填 | 备注 |
| CheckID | 记录唯一标识 | String | 是 | 例：201902170907221001 |
| department | 项目部 | String | 是 | 上传对应的ID |
| detComp | 被检单位名称 | String | 是 | 上传对应的ID |
| boothNumber | 摊位号 | String | 是 | 上传对应的ID |
| sampNumber | 样品编号 | String | 是 |  |
| category | 样品分类 | String | 是 |  |
| sampleName | 样品名称 | String | 是 |  |
| testItem | 检测项目 | String | 是 |  |
| standardVal | 标准值 | String | 是 |  |
| detection | 检测数值 | String | 是 |  |
| detectionCom | 检测值单位 | String | 是 |  |
| detResult | 检测结果 | String | 是 |  |
| detGist | 检测依据 | String | 是 |  |
| detDate | 检测时间 | String | 是 |  |
| jcshi | 检测室 | String | 是 |  |
| inspector | 检测员 | String | 是 | 上传登录人员的ID |

返回值json：

{

{

“code”:”1000”,

“count”:”1”

"msg": "操作成功",

"CheckID":"201902170907221001"

},

.

.

.

{

“多条重复”

}

}

返回json参数说明：

|  |  |  |
| --- | --- | --- |
| **参数名称** | **数据类型** | **说明** |
| code | String | 结果代码，代码定义详见：附-返回代码表 |
| msg | String | 结果提示信息：操作成功！或 操作失败：未登录！或参数不能为空等 |
| count | String | 数量 |
| CheckID | String | 返回上传记录的唯一码，方便失败提示哪条记录错误 |

**附.返回代码表**

所有的成功结果返回1000和相应的数据集合；

|  |  |  |
| --- | --- | --- |
| 返回代码 | 代码描叙 | 备注 |
| 1000 | 操作成功 |  |
| 1001 | 参数XX不能为空 |  |
| 1002 | 参数不正确 |  |
| 1003 | 参数格式不正确 |  |
| 1004 | 数据类型未定义 |  |
| 1005 | 记录不存在或已删除 |  |
| 1006 | 记录已存在 |  |
| 1007 | 用户TOKEN失效 |  |
| 1008 | 账户不存在 |  |
| 1009 | 密码错误 |  |
| 1010 | 账户被停用 |  |
| 1011 | 未知异常 |  |
| 1012 | 该数据已经上传 |  |

备注信息：

项目部是一级菜单，被检单位是二级菜单，摊位号是三级菜单

测试账号 admin 密码 a123456