**项目名称：**智能化文本识别

**文档标题：**概要设计说明书

**作者：**任智超

**日期：**2024年7月2号

**版本号：**1.0

目录

[一、 概要说明 2](#_Toc8226)

[背景目的 2](#_Toc24701)

[二、总体设计 3](#_Toc20420)

[2.1功能需求 3](#_Toc1950)

[1） 人工观察 3](#_Toc19189)

[2）数据识别 3](#_Toc3183)

[3）模型结果规则处理 3](#_Toc240)

[4） 模型识别结果 4](#_Toc9806)

[2.2流程结构 4](#_Toc25729)

[1） 数据标注阶段 5](#_Toc14898)

[2） 数据模型识别阶段 5](#_Toc30387)

[三、模块描述 5](#_Toc32033)

[3.1 模块基本信息表 5](#_Toc6134)

[3.2 模块算法处理逻辑 6](#_Toc22341)

[3.3接口描述 11](#_Toc23392)

[3.3.1 上下文规则接口 11](#_Toc29865)

[3.3.2 标注识别接口 12](#_Toc13552)

[3.3.3 存储规则信息 14](#_Toc6378)

[3.3.4 删除规则信息 16](#_Toc25345)

[3.3.5导出模型文件 17](#_Toc13661)

[3.3.6导入模型文件 18](#_Toc13396)

# 概要说明

## 背景目的

智能化文本识别希望通过人工标识的情况下，针对重要数据进行分析提取和规则的形成；减少后期开发人员的人工干预，提高效率；

该设计书目的是为了描述智能化文本识别系统的设计，提供设计思想，收集新的意见，并做以改进及优化，确保系统开发和实现能够符合后期想要达到的预期和效果；

# 二、总体设计

智能化文本识别旨在利用人工智能技术，实现高效、准确的文本识别和处理。系统通过人工标识和自动识别相结合，根据形成上下文规则，数据标签，数据标注三要素结合，对后续文本进行自动提取和识别文本内容；

## 2.1功能需求

### 人工观察

· 通过人工观察分析，选择需要标注的具体数据；

### 2）数据识别

· 上下文规则：将人工选择需要标注的数据，进行分析，并返回上下文规则信息；

· 数据标签：根据预定的上下文规则，对识别的文本进行处理；

· 数据标注：结合已有的标注数据，选择TEXT文本识别、JSON文本识别、JSON多选文本识别，自动识别和提取文本数据；

### 3）模型结果规则处理

· 数据验证：通过自己标识的信息，人工观察数据是否出现误报和错误信息；

· 数据新增：根据标识信息，形成当前子模型规则信息，持久化存储到PKL文件中；

· 模型搜索：已经存储的子模型形成列表存放到界面，可以搜索子模型名称；

· 模型修改：修改已经存在的子模型型数据，包括子模型名称、数据标签、标注信息等等；

· 模型删除：删除已经存在的子模型型数据；

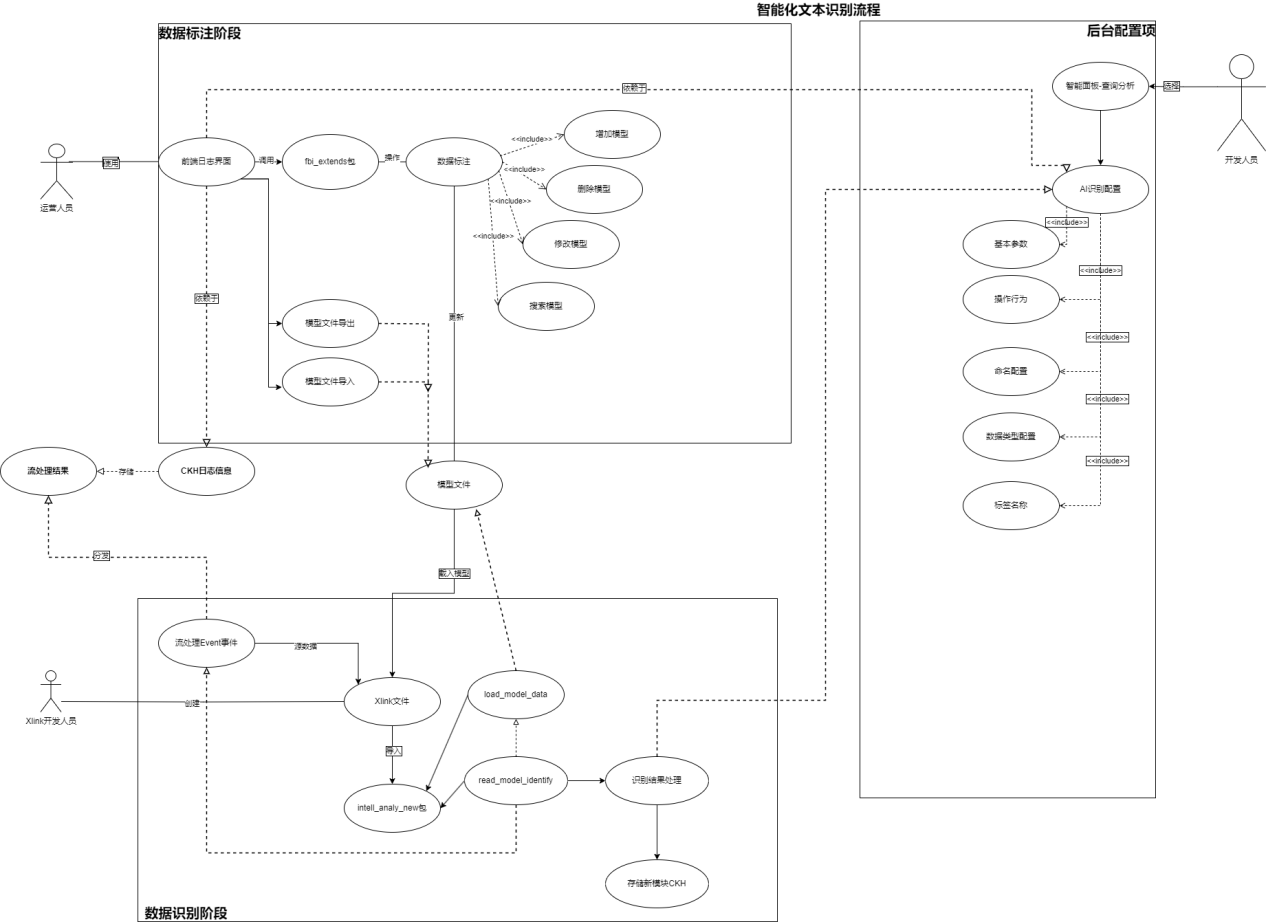
· 数据导入及导出：存储PKL模型文件可以用于导入和导出，生成zip压缩文件，可以用于平台的不同环境当中；

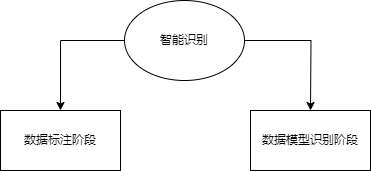
### 模型识别结果

· 模型识别：流处理获取模型数据，对实时流量进行自动化识别；

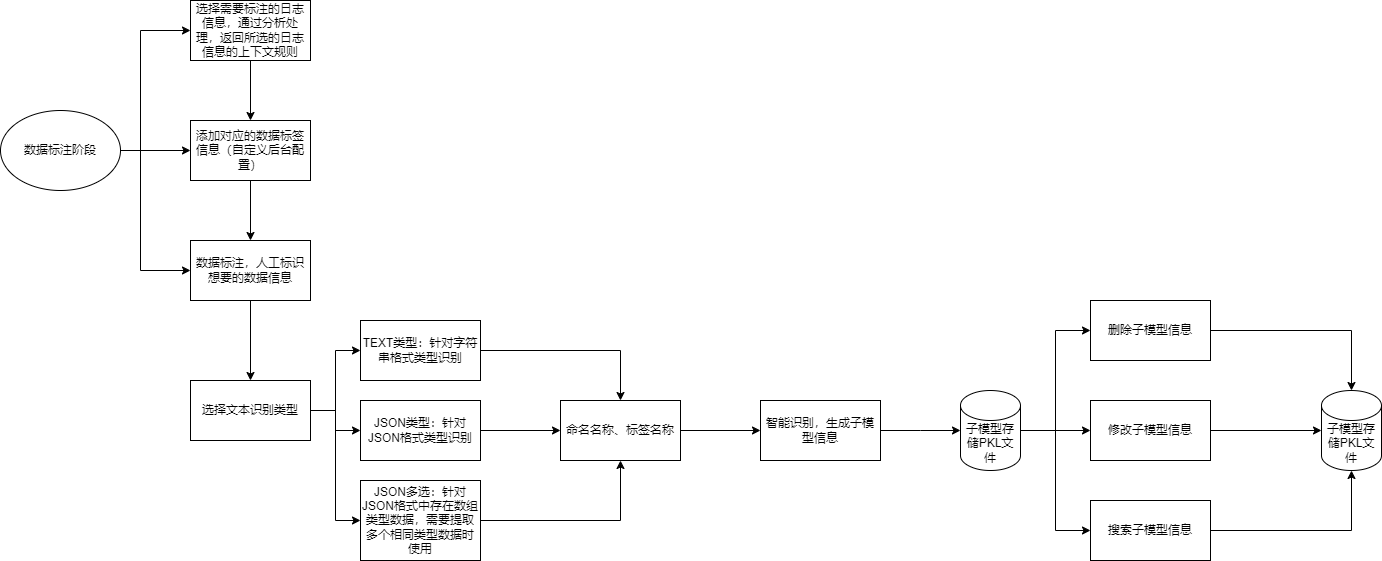
· 结果输出：识别结果以Python对象（Dict）形式输出；

## 2.2流程结构





### 数据标注阶段



### 数据模型识别阶段



# 三、模块描述

## 3.1 模块基本信息表

|  |  |  |
| --- | --- | --- |
| 名称 | 设计者 | 所在文件 |
| 数据标注模块 | 任智超 | /opt/openfbi/fbi-bin/fbi\_extends.py |
| 数据模型识别 | 任智超 | /opt/openfbi/pylibs/intell\_analy\_new.py |
| 配置项 | 任智超 | /opt/openfbi/pylibs/config.ini |

## 3.2 模块算法处理逻辑

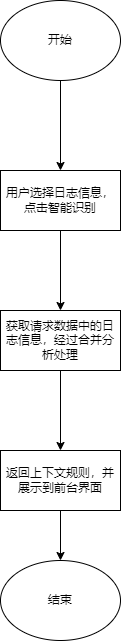


图3-1 上下文规则处理逻辑

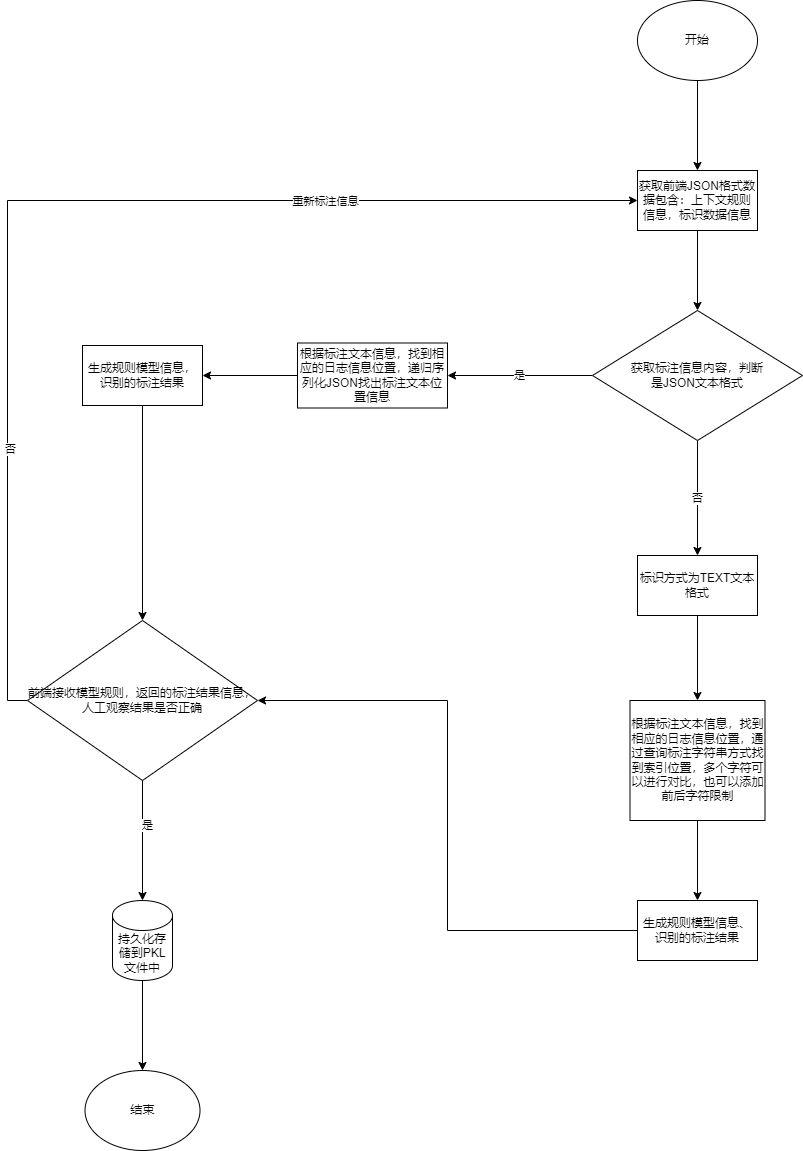


图3-2 文本标注识别逻辑

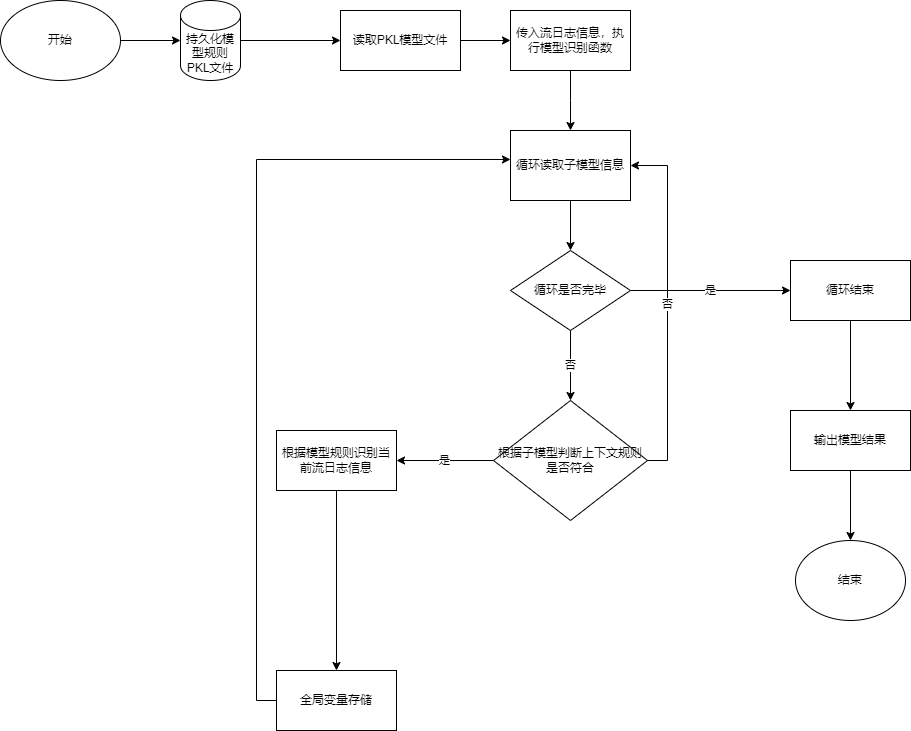


图3-3 模型识别逻辑流程

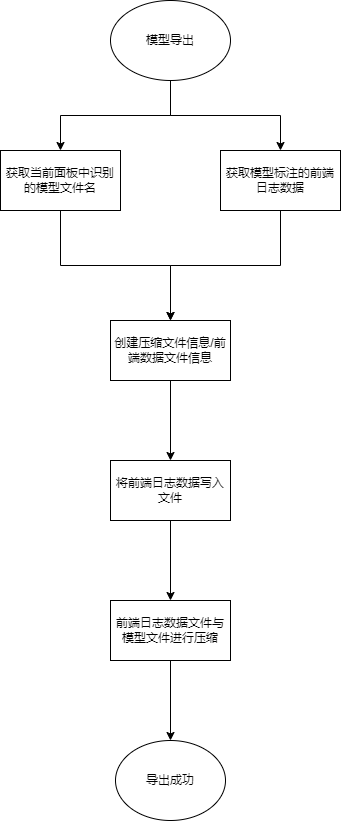


图3-4模型导出流程

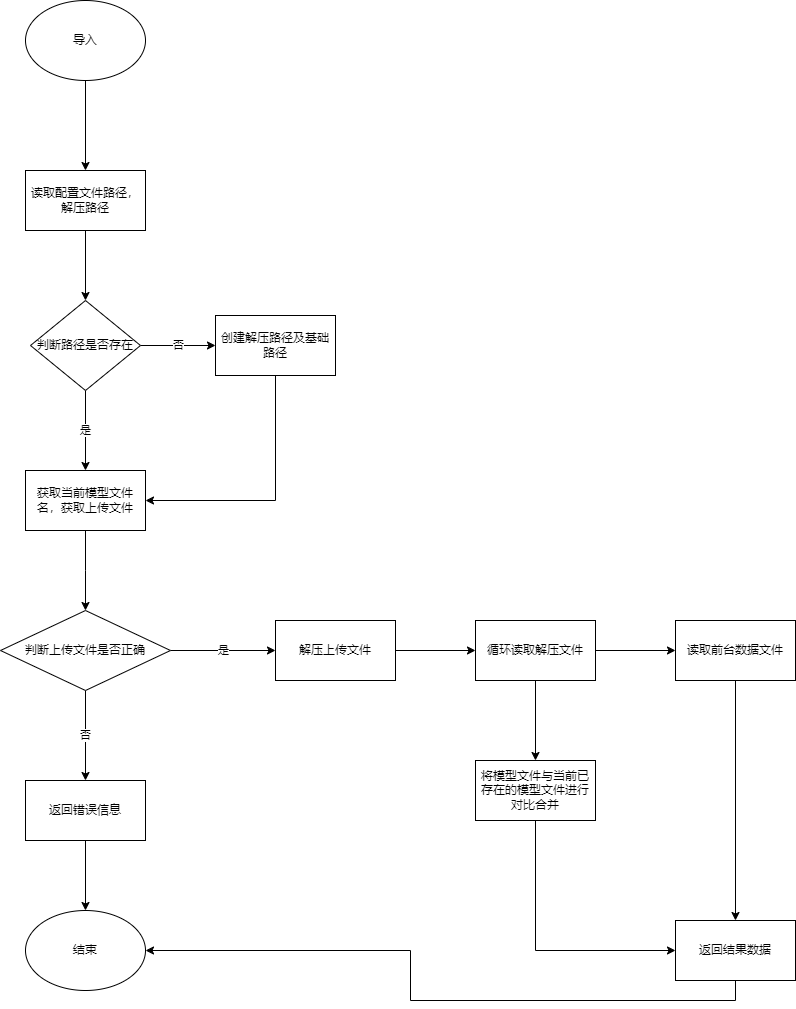


图3-5模型导入流程

## 3.3接口描述

### 3.3.1 上下文规则接口

#### 1.接口描述

这里的IP可以为任意公司环境/或者现场环境

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | cls\_d | | |
| URL Path | https://192.168.124.254/db/cls\_d | | |
| 请求类型 | POST | | |
| 请求报文 |  | | |
| 处理逻辑 | 验证是否存在请求报文  验证上下文规则信息关联是否出错  根据以上判断、验证返回 JSON 数据 | | |
| 返回值 | 若未传入请求报文  {      "status": "Error",      "msg": "'NoneType' object has no attribute 'get'"  }  若传入请求报文为空：{“datas”:[]}  {      "status": "Success",      "cls\_res": {}  }  正常传入： | | |
| 返回参数 | 参数名 | 中文说明 | 描述 |
| status | 执行状态 |  |
| msg | 程序运行出错报错信息 |  |
| cls\_res | 存储上下文规则数据 | 通过处理过的上下文信息返回给前端展示 |

#### 2.请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **必填** | **类型** | **说明** |
| datas | 是 | LIST | 用来传递用户选择的日志信息 |

### 3.3.2 标注识别接口

#### 1.接口描述

这里的IP可以为任意公司环境/或者现场环境

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | asl\_d | | |
| URL Path | https://192.168.124.254/db/als\_d | | |
| 请求类型 | POST | | |
| 请求报文 |  | | |
| 处理逻辑 | 判断传入请求报文中是否缺少键  判断处理规则是否处理失败  获取规则之后判断识别信息是否出错 | | |
| 返回值 | 请求报文中缺少“con”键  {      "status": "Error",      "res": "'con'"  }  请求报文中缺少“datas”键  {      "status": "Error",      "res": "'datas'"  }  处理规则数据失败时  {      "status": "Error",       "msg": "规则提取失败:"  }  识别信息出错  {      "status": "Error",       "msg": "数据识别失败:"  }  识别成功时 | | |
| 返回参数 | 参数名 | 中文说明 | 描述 |
| status | 执行状态 |  |
| msg | 程序运行出错报错信息 |  |
| res | 成功结果 | 存放识别结果、子模型名称、上下文规则、模型规则信息 |

#### 2.请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **必填** | **类型** | **说明** |
| datas | 是 | LIST | 用来传递用户选择的日志信息 |
| cons | 是 | DICT | 用来传递前台识别好的上下文规则 |
| data | 是 | DICT | datas的子键，是存放日志信息 |
| Imps | 是 | LIST | 标注的文本信息 |
| Idx | 是 | INT | 当前日志信息的索引 |

### 3.3.3 存储规则信息

#### 1.接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | s\_rules\_con | | |
| URL Path | https://192.168.124.254/db/s\_rules\_con | | |
| 请求类型 | POST | | |
| 请求报文 |  | | |
| 判断逻辑 | 判断传入请求报文是否为空  判断file\_str是否为空  判断新增模型是否成功  判断修改模型是否成功 | | |
| 返回值 | 请求报文为空  {   "status": "Error",   "msg": "获取前台数据失败:{e.\_\_str\_\_()}"  }  参数file\_str为空  {      "status": "Error",   "msg": "模型文件不能为空"  }  新增模型错误  {   "status": "Error",   "msg": "子模型新增错误"  }  修改模型错误  {   "status": "Error",   "msg": "子模型修改错误"  }  新增成功时  {      "status": "Success",      "message": "模型存储成功！",      "add\_msg": "新增模型成功！",      "alter\_msg": ""  }  修改模型成功时  {      "status": "Success",      "message": "模型存储成功！",      "add\_msg": "",      "alter\_msg": "修改模型成功！"  } | | |
| 返回值参数 | 参数名 | 中文说明 | 描述 |
| status | 执行状态 |  |
| message | 执行成功时 |  |
| add\_msg | 新增 | 新增时返回值否则为空 |
| alter\_msg | 修改 | 修改时返回值否则为空 |
| msg | 执行失败时 |  |

#### 2.请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **必填** | **类型** | **说明** |
| add | 否 | DICT | 新增时的值信息 |
| alter | 否 | DICT | 修改时的值信息 |
| file\_str | 是 | String | 模型文件名 |
| con | 是 | DICT | add或alter的子键，上下文信息 |
| rules | 是 | DICT | add或alter的子键，规则信息 |
| linfo | 是 | DICT | add或alter的子键，数据标签信息 |
| model\_key | 是 | String | add或alter的子键,子模型名称 |
| orl\_key | 是 | String | alter的子键，修改时旧的子模型名称 |

### 3.3.4 删除规则信息

#### 1.接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | delete\_rules | | |
| URL Path | https://192.168.124.254/db/delete\_rules | | |
| 请求类型 | POST | | |
| 请求报文 | {      "model\_key":"模型2222",      "file\_str":"大模型"  } | | |
| 判断逻辑 | 判断模型数据读取是否成功  判断删除模型数据是否成功  判断子模型数据是否存在 | | |
| 返回值 | 子模型删除成功  {      "status": "Success",      "msg": "子模型-模型2222-删除成功！"   }  模型数据读取失败  {      "status": "Error",      "msg": "模型数据读取错误：{e.\_\_str\_\_()}"   }  子模型不存在  {      "status": "Error",      "msg": "子模型-模型2222-模型数据不存在！"  } | | |
| 返回参数 | 参数名 | 中文说明 | 描述 |
| status | 执行状态 |  |
| msg | 执行结果详细信息 |  |

#### 2.请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **必填** | **类型** | **说明** |
| model\_key | 是 | String | 子模型名称 |
| file\_str | 是 | Srting | 模型文件名称 |

### 3.3.5导出模型文件

#### 1.接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | model\_download | | |
| URL Path | https://192.168.124.254/db/model\_download | | |
| 请求类型 | POST | | |
| 请求报文 |  | | |
| 处理逻辑 | 判断参数是否为空  判断前台数据导出是否成功  判断导出压缩文件是否成功 | | |
| 返回值 | filename为空  {      "status": "Error",      "msg": "未提供文件名"   }  前台数据导出失败  {      "status": "Error",      "msg": "子模型列表导出失败"   }  导出成功  {      "status": "Success",      "download\_link": "导出路径"  } | | |
| 返回参数 | 参数名 | 中文说明 | 描述 |
| status | 执行状态 |  |
| msg | 错误信息 |  |
| download\_link | 路径信息 |  |

#### 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **必填** | **类型** | **说明** |
| filename | 是 | String | 模型文件名 |
| current\_data | 是 | String | 模型列表前台数据 |

### 3.3.6导入模型文件

#### 1.接口描述

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名 | model\_upload | | |
| URL Path | https://192.168.124.254/db/model\_upload | | |
| 请求类型 | POST | | |
| 请求参数 | 参数名 | 中文说明 | 描述 |
| file\_str | 模型文件名 | 当前面板识别下的模型文件名 |
| File | 上传的文件 |  |
| 处理逻辑 | 判断上传文件是否存在  解压上传文件是否出错  解压路径问题  上传错误 | | |
| 返回值 | 上传文件不存在  {      "status": "Error",      "msg": "未提供文件或文件名"  }  解压模型文件出错  {      "status": "Error",      "msg": "解压模型文件出错"  }  解压路径文件出错  {      "status": "Error",      "msg": "路径文件异常，请重新上传模型文件"  }  上传成功  {      "status": "Success",      "data": [],      "model\_info": {}  } | | |
| 返回参数 | 参数名 | 中文说明 | 描述 |
| status | 执行状态 |  |
| data | 前台子模型列表数据 |  |
| model\_info | 模型文件写入 |  |
| msg | 错误信息 |  |