# SQL在线审核系统前端接口文档



# 1 之前测试时前端发送样式（用户管理供参考）

## 1. 1 用户创建 ajax 样例

```
    add: function(){
          var requestJSONObject = {
              // 发送的数据
          };

          $.ajax({
              url: latkeConfig.servePath + "/console/user",
              type: "POST",
              contentType: "application/json",
              cache: false,
              data: JSON.stringify(requestJSONObject),
              dataType: 'json',
              success: function (result, textStatus) {
                  $("#tipmessage").text(result.message); // 获取提示信息
                  if (result.code != 0) { // 根据code码判断后台是否异常

                  }
                  // 一些操作
              }
          });
```

# 2 数据格式统一标准

### code码定义

- 0 : 一切正常
- 4000 : 前台传输的数据错误
- 5000 : 后台服务器发生未知错误

> 并没有使用HTTP状态码来表示异常，所以HTTP状态码永远是200

### 后台返回统一格式

```
{
    "code":
    "message" : 提示信息
    "data": 返回数据; 若非查询类操作, 则返回null;
}
```

### 无权限判断

### 前端传输json数据

- 部分字段是不可缺少的（ 倘若缺少，则会提交失败 ）
- 未做说明的字段是可有可无的

# 3 用户管理接口

## 3.1 增加用户

#### 前端发送数据

```
url    : /console/user
method: POST
contentType : "application/json"
dataType: 'json'
data: {
    "name":  用户姓名（不可缺少）
    "email":  用户邮箱 （不可缺少）
    "account": 用户账户 （不可缺少）
}
```

#### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息(创建成功/创建失败)
    "data": null
}
```

## 3.2 删除用户

#### 前端发送数据

```
url    : /console/user/{user id}
method: DELETE
```

#### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": null
}
```

## 3.3 更新用户

#### 前端发送数据

```
url   : /console/user/{user id}
method: PUT
contentType : "application/json"
dataType: 'json'
data: {
    "name": // 用户姓名
    "email": // 用户邮箱
}
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息
    "data": null
}
```

# 3.4 查询具体用户

**前端发送数据**

```
url   : /console/user/{user id}
method: GET
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "id": 主键
        "name": 姓名
        "account": 账户名
        "email": 邮箱
    }
}
```

# 3.5 查询用户列表

**前端发送数据**

```
url   : /console/user?page={page number}
method: GET
```

#### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "total": 总行数
        "pages": 总页数
        "pageNum": 当前第几页
        "pageSize": 每页显示的行数
        "size":  当前页的行数
        "list": [ {
                "id": 主键
                "name": 用户姓名
                "account": 用户账户
                "email": 用户邮箱
            },  {
                // 同上
            }
        ]
    }
}
```

# 4 数据库管理

## 4.1 数据库创建

#### 前端发送数据

```
url   : /console/db
method: POST
contentType : "application/json"
dataType: 'json'
data: {
    "name": 数据库姓名 (不可缺少)
    "pid":  数据库关联的项目ID
    "project": 数据库关联的项目名
    "url": 数据库配置源URL
    "port": 数据库配置源Port端口号
    "account": 数据库配置源账户
    "password": 数据库配置源密码
}
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息(成功/失败原因)
    "data": null
}
```

# 4.2 数据库删除

**前端发送数据**

```
url   : /console/db/{database id}
method: DELETE
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息(成功/失败原因)
    "data": null
}
```

# 4.3 数据库更新

**前端发送数据**

```
url   : /console/db/{database id}
method: PUT
contentType : "application/json"
dataType: 'json'
data: {
    "pid":  数据库关联的项目ID
    "project": 数据库关联的项目名
    "url": 数据库配置源URL
    "port": 数据库配置源Port端口号
    "account": 数据库配置源账户
    "password": 数据库配置源密码
}
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息(成功/失败信息)
    "data": null
}
```

# 4.4 查询具体数据库

### 前端发送数据

```
url   : /console/db/{database id}
method: GET
```

### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "id": 主键
        "name": 数据库姓名
        "pid":  数据库关联的项目ID
        "project": 数据库关联的项目名
        "url": 数据库配置源URL
        "port": 数据库配置源Port端口号
        "account": 数据库配置源账户
        "password": 数据库配置源密码
        "createTime": 数据库创建时间
    }
}
```

# 4.5 查询数据库列表

### 前端发送数据

```
url   : /console/db?page={page number}
method: GET
```

### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "total": 总行数
        "pages": 总页数
        "pageNum": 当前第几页
        "pageSize": 每页显示的行数
        "size":  当前页的行数
        "list": [ {
                "id": 主键
                "name": 数据库姓名
                "pid":  数据库关联的项目ID
                "project": 数据库关联的项目名
                "url": 数据库配置源URL
                "port": 数据库配置源Port端口号
                "account": 数据库配置源账户
                "password": 数据库配置源密码
                "createTime": 数据库创建时间
            },  {
                // 同上
            }
        ]
    }
}
```

# 5 开发人员SQL管理

## 5.1 SQL创建

#### 前端发送数据

```
url   : /sql
method: POST
contentType : "application/json"
dataType: 'json'
data: {
    "dbId":  SQL语句关联的数据库ID （不可缺少，且该数据库必须存在）
    "dbName":  SQL语句关联的数据库名 (不可缺少)
    "statement": SQL语句       (不可缺少)
}
```

#### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息(成功/失败)
    "data": null
}
```

# 5.2 更新SQL语句

> 后台是默认只能更新未被提交的SQL语句，已提交的SQL语句不可被更新

**前端发送数据**

```
url   : /sql/{sql id}
method: PUT
contentType : "application/json"
dataType: 'json'
data: {
    "statement":  SQL语句 (不可缺少)
}
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息(成功/失败)
    "data": null
}
```

# 5.3 删除SQL语句

> 后台是默认只能删除未被提交的SQL语句，已提交的SQL语句不可被删除

**前端发送数据**

```
url   : /sql/{sql id}
method: DELETE
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息(成功/失败原因)
    "data": null
}
```

# 5.4 查询SQL具体信息

### 前端发送数据

```
url   : /sql/{sql id}
method: GET
```

### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "id": 主键
        "dbId": 数据库ID
        "dbName":  数据库名
        "uId": SQL的提交者ID
        "uName": SQL提交者姓名
        "mId": SQL审核人的ID
        "mName": SQL审核人姓名
        "statement": SQL语句
        "status": SQL状态显示
        "sRemark": SQL申请时的备注
        "rRemark": SQL拒绝时的备注
        "createTime": SQL创建时间
        "submitTime": SQL提交时间
        "executeTime": SQL执行时间
    }
}
```

# 5.5 查询SQL列表

### 前端发送数据

```
url   : /sql?page={page number}&status=0
(status含义: 0 - 用户查询自己的全部SQL列表
             1 - 用户查询未提交的SQL列表
             2 - 用户查询已提交的SQL列表
method: GET
```

### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "total": 总行数
        "pages": 总页数
        "pageNum": 当前第几页
        "pageSize": 每页显示的行数
        "size":  当前页的行数
        "list": [ {
            "id": 主键
            "dbId": 数据库ID
            "dbName":  数据库名
            "uId": SQL的提交者ID
            "uName": SQL提交者姓名
            "mId": SQL审核人的ID
            "mName": SQL审核人姓名
            "statement": SQL语句
            "status": SQL状态显示
            "sRemark": SQL申请时的备注
            "rRemark": SQL拒绝时的备注
            "createTime": SQL创建时间
            "submitTime": SQL提交时间
            "executeTime": SQL执行时间
            },  {
                // 同上
            }
        ]
    }
}
```

## 5.6 提交SQL申请

### 前端发送数据

```
url   : /sql/{sql id}/submit
method: PUT
contentType: application/json
dataType: json
data: {
    remark: 提交申请（可填写，也可不填写）
}
```

#### 后台返回数据

```
{
    code: code码
    message : 提示信息
    data：null
}
```

# 6 管理员SQL管理

## 6.1 SQL列表查看

#### 前端传输数据

```
url: /console/sql?page={page number}&status={查询条件}
(status: 0 - 管理员查询全部SQL列表
         1 - 管理员查询未审核SQL列表
         2 - 管理员查询已审核SQL列表)
method: GET
```

#### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "total": 总行数
        "pages": 总页数
        "pageNum": 当前第几页
        "pageSize": 每页显示的行数
        "size":  当前页的行数
        "list": [ {
            "id": 主键
            "dbId": 数据库ID
            "dbName":  数据库名
            "uId": SQL的提交者ID
            "uName": SQL提交者姓名
            "mId": SQL审核人的ID
            "mName": SQL审核人姓名
            "statement": SQL语句
            "status": SQL状态显示
            "sRemark": SQL申请时的备注
            "rRemark": SQL拒绝时的备注
            "createTime": SQL创建时间
            "submitTime": SQL提交时间
            "executeTime": SQL执行时间
            },  {
                // 同上
            }
        ]
    }
}
```

# 6.2 SQL具体信息查看

### 前端发送数据

```
url    : /console/sql/{sql id}
method: GET
```

### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息
    "data": {
        "id": 主键
        "dbId": 数据库ID
        "dbName":  数据库名
        "uId": SQL的提交者ID
        "uName": SQL提交者姓名
        "mId": SQL审核人的ID
        "mName": SQL审核人姓名
        "statement": SQL语句
        "status": SQL状态显示
        "sRemark": SQL申请时的备注
        "rRemark": SQL拒绝时的备注
        "createTime": SQL创建时间
        "submitTime": SQL提交时间
        "executeTime": SQL执行时间
    }
}
```

# 6.3 SQL通过

### 前端发送数据

```
url    : /console/sql/{sql id}/approve
method: PUT
```

### 后台返回数据

```
{
    "code": code码
    "message" : 提示信息(成功/失败原因)
    "data": null
}
```

# 6.4 SQL驳回

### 前端发送数据

```
url   : /console/sql/{sql id}/against
method: PUT
contentType: application/json
dataType: json
data: {
    remark: 拒绝理由(可选)
}
```

**后台返回数据**

```
{
    "code": code码
    "message" : 提示信息(成功/失败原因)
    "data": null
}
```

# 7 日志查询

## 7.1 获取SQL语句执行日志

**前端发送数据**

```
url: /log?sqlid={sql id}
method: GET
```

**后台返回数据**

```
{
    code:
    message :
    data: [ {
        "id": 主键
        "sId": 日志关联的SQL ID
        "log": 日志内容
        "createTime": 日志创建时间
    }, {
        // 同上
    }]
}
```