

excel 自动导入数据库步骤

利用 Postgres 数据库, MySQL数据库, Navicat 16 工具, VScode工具。

1、进入文件夹, 打开对应项目。

启动 `vscode`, 文件 -> 打开文件夹 -> 选择 `D:/public/vue-excel-process-system-master` 文件夹, 打开项目。然后点击 `import-excel` -> `excelTodatabase.py`。

2、修改 `excel` 路径, 以及对应数据库名称。

点击 `excelTodatabase.py`, 修改以下内容:

```
// 其中path为需要导入数据库的excel的路径。路径名称不能含有-
// database 为导入excel的数据库名称。
def createtable():
    path='C:/Users/320200255/project/importExcel/test.xlsx'
    ...
    conn = psycopg2.connect(database='postgres-uploadsearch', user='postgres',
password='postgres', host='localhost')
    ...
```

3、运行项目。

修改好, 点击保存。右键 `Run code`, 运行。若报语法错误, 大都是 `excel` 表格属性名不规范导致, 参照以下, 对 `excel` 内容进行修改即可。

注意:

(1) 如需上传的`excel`的 `sheet` 表名与数据库已有表重名, 请到 步骤4, 删除旧的重复的表。再进行一下操作。(2) 导入的`excel`表的属性行需符合数据库表的规范。`excel` 表的属性行(第一行)符合:

- 采用26个英文字母和0-9的自然数加上下划线 `_` 组成, 多个单词用下划线 `_` 分隔
- 删除/重命名 重复的列名
- 禁止使用数据库关键字, 如: `name`, `time`, `datetime` `password` 等
- 在命名表的列时, 不要重复表的名称

(3) 导入的`excel`表内容需符合数据库表的规范。`excel` 表内容符合:

- 不能含有 `'`, `;`, `--`, `==`, `<`, `>` 等非法字符。(可将这些非法字符替换为 `_`)
- 如 `can't` -> `can not` / `can_t`
- 如 `what's` -> `what is` / `what_s`
- 如 `It's` -> `It is` / `It_s`
- 如 `We're` -> `We are` / `We_re`

4、打开 `postgreSQL` 数据库，刷新表。

可直接通过 `Navicat` 工具，查看 `postgres-uploadsearch` 数据库中的表，点击刷新。

或者打开电脑应用程序 `postgreSQL 11` 中的 `pgAdmin 4` 应用，账户密码都为 `postgres`。依次点击 `postgres-uploadsearch -> postgres-uploadsearch -> public -> 表`。右键 `表`，进行刷新，即可看到刚刚上传的 `excel` 中的每个 `sheet` 表。

5、将 `postgreSQL` 中的表复制到 `MySQL`。

通过 `Navicat` 工具，将 `postgres-uploadsearch` 中的 `postgres-uploadsearch` 数据库中的表复制到 `localhost_3306` 下边的 `uploadsearch` 数据库。完成本地 `excel` 导入 `postgreSQL` 数据库，并通过 `Navicat` 工具复制到 `mysql` 数据库。

还可通过 `DBeaver` 工具，点击 `postgres-uploadsearch` 中的 `postgres-uploadsearch` 数据库中的表。右击这个表 -> 导出数据 -> 复制到 `localhost_3306` 下边的 `uploadsearch` 数据库 -> 一直点击 `next` 即可。完成本地 `excel` 导入 `postgreSQL` 数据库，并通过 `DBeaver` 工具复制到 `mysql` 数据库。

vue 实现多表查找搜索

1、进入文件夹，打开对应项目。

启动 `vscode`，文件 -> 打开文件夹 -> 选择 `vue-excel-process-system-master` 文件夹。

2、修改 `server` 文件夹下的：`API`。`index.js`。`router.js`。`HelloWorld.vue` 样式。

(2.1) 修改 `API`：在 `API` 文件夹下，新建 以需要导入的表名为名字的 `js` 文件，如 `sheet1.js`。

```
// sheet1.js 示例
let db = require('../db/index')

exports.get = (req, res) => {
  // 修改下边的语句，sheet1为待替换的表的名字
  var sql = 'select * from sheet1'
  db.query(sql, (err, data) => {
    if(err) {
      return res.send('错误: ' + err.message)
    }
    res.send(data)
  })
}
```

(2.2) 若需要修改数据库名称时，进入 `server/db/index.js`，可修改相关配置，保存。

```
let mysql = require('mysql')

let db = mysql.createPool({
  host: '127.0.0.1', //数据库IP地址
```

```
    user: 'root',           //数据库登录账号
    password: 'root',       //数据库登录密码
    database: 'uploadsearch' //要操作的数据库
  })
  module.exports = db
```

(2.3) 修改 `router.js`。

```
let express = require('express')
let router = express.Router()

// 以sheet1为例，新增下边两行代码
let sheet1 = require('./API/sheet1')
router.get('/sheet1', sheet1.get)

module.exports = router
```

(2.4) 修改 `src/components/HelloWorld.vue` 前端样式，保存。

```
<!-- 从数据库获取表 -->
<div class="drop-select">
  <el-dropdown @command="handleCommand">
    <span class="el-dropdown-link">
      下拉菜单，选择表
      <i class="el-icon-arrow-down el-icon--right"></i>
    </span>
    <el-dropdown-menu slot="dropdown" placement="top-end">

      // 在此处添加 sheet1为示例新增表名 开始添加-----
      <el-dropdown-item command="sheet1">sheet1</el-dropdown-item>
      // 添加结束-----

      <el-dropdown-item command="refresh2">refresh2</el-dropdown-item>

    </el-dropdown-menu>
  </el-dropdown>
</div>
```

3、启动服务器。

终端输入 `cd server`。终端输入 `node app.js`。若显示 服务器启动成功，则 ok。

4、启动项目。

终端界面，点击 `powershell` 右边的 + 后，终端再输入 `npm run dev` 启动项目。网页输入生成的地址，即可。

5、打包项目。

终端输入 `ctrl+c` 停止项目，终端输入 `npm run build` 打包项目。此时项目目录会自动更新 `dist` 文件夹。将 `dist` 文件夹复制到内网，替换原先的 `dist` 文件夹。

6、内网访问项目。

依次点击 `public` 文件夹 -> `dist` 文件夹 -> `index.html`。即可打开网页，进行搜索。