pandas && 强大的数据操作处理程序

安装

```
pip install pandas -i https://pypi.tuna.tsinghua.edu.cn/simple

# excel支持
pip install openpyxl -i https://pypi.tuna.tsinghua.edu.cn/simple
pip install xlsxwriter  -i https://pypi.tuna.tsinghua.edu.cn/simple
```

连接到数据

CSV,JSON,Excel,mysql等。

https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html

api设计的非常简洁:

- 读取: read_csv , read_json , read_excel,read_sql
- 写入: to_csv , to_json, to_excel , to_sql

连接到mysql

```
pip install pandas -i https://pypi.tuna.tsinghua.edu.cn/simple
pip install sqlalchemy -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
# 增加pandas库的导入
import pandas as pd

# 添加SQLAlchemy的导入
from sqlalchemy import create_engine
```

```
# 示例用法
if True:
    engine = create_engine(
        "mysql+pymysql://root:password@192.168.0.10:33318/renren_cloud_basic"
    )
    query = "SELECT * FROM park_user"
    # 使用chunksize参数来实现流式处理
    df = pd.read_sql(query, engine)
    # 显示数据的结构和数据类型
    print(df.info())
    # 显示列名
    print(df.columns)
```

连接到mysql的额外操作

- read_sql_table(table_name,connection)
- read_sql_query(sql,connection)
- read_sql(sql,connection)

操作数据

- 获取数据形状

```

row_size, col_size = grouped.shape
log.info(f"row_size:{row_size},col_size:{col_size}")


# row_size:13,col_size:7
```

- 数据清洗：提取数据到新的列

```

# 将月份列提取为年
dataFrame["年份'] = pd.to_datetime(dataFrame["月份']).dt.year
```

- 数据清洗：清洗空值
- 数据清洗：清洗格式错误数据
- 数据清洗：清洗错误数据

- 数据清洗：清洗重复数据
- 分组聚合

```
grouped = (
    dataFrame.groupby(["客户名称", "合同编号", "年份"])
    .agg(
        {
            "减免总金额": "sum",
            "政策减免金额": "sum",
            "运营减免金额": "sum",
            "其他减免金额": "sum",
        }
    )
    .reset_index()  # 使分组变成一个平面
)

# 设置索引列的名称为“序号”
grouped.index.name = '序号'
```

- 筛选数据

```
# 提取整列数据
print(grouped["客户名称"])



# 筛选客户名称列数据
print(grouped[grouped["客户名称"] == "山东智拓大数据有限公司"])
```

写入Eexcel时设置样式

pandas是做数据处理的，默认导出并不能设置样式、合并单元格等，需要记住其他工具。

```
# 自定义写入程序，使用xlsxwriter写入
with pd.ExcelWriter("./files/output.xlsx", engine="xlsxwriter") as writer:


    # 直接写入数据，没有样式
```

```
# grouped.to_excel(writer, sheet_name="Sheet1", startrow=1, startcol=1, index=True)

# 获取工作表
workbook = writer.book
worksheet = None
# 如果worksheet不存在，则创建一个工作表
if "Sheet1" in writer.sheets:
    worksheet = writer.sheets["Sheet1"]
else:
    worksheet = workbook.add_worksheet("Sheet1")

# 创建样式
default_format = {
    "bold": False,
    # 边框
    "border": 1,
    # 垂直居中
    "valign": "vcenter",
    # 自动换行
    "text_wrap": True,
    # 水平居中
    # "align": "center",
    # 字体大小
    "font_size": 9,
}

# 创建样式
cell_format_default = workbook.add_format(default_format)
cell_format_head = workbook.add_format(
    {
        **default_format,
        **{
            "align": "center",
            "bold": True,
        },
    }
)

# 设置列的宽度，单位是多少个字符，一个中文占两个字符
worksheet.set_column("A:A", 10)
worksheet.set_column("B:B", 30)
```

```
worksheet.set_column("C:C", 20)
worksheet.set_column("E:H", 12)

# 插入logo图片
worksheet.insert_image(
    "A1",
    "./files/logo.png",
    {
        "x_scale": first_row_height / logo_image_height,
        "y_scale": first_row_height / logo_image_height,
        "x_offset": 5,
        "y_offset": 5,
    },
)
# 合并单元格
worksheet.merge_range("A1:B1", "", cell_format_head)
worksheet.merge_range("C1:H1", '租金减免表', cell_format_head)
# 设置行高，单位是像素
worksheet.set_row(0, first_row_height)

# 增加序号列
worksheet.write("A2", '序号', cell_format_head)
worksheet.write_column("A3", range(1, len(grouped) + 1), cell_format_head)

# 计算行和列的偏移量
row_offset = 2
col_offset = 1

# 写入数据并设置样式，使用xlsxwriter作为驱动引擎时，需要将数据与样式一起写入，并不能直接修改样式
for row_num, row_data in grouped.iterrows():
    for col_num, col_data in enumerate(row_data):
        worksheet.write(
            row_num + row_offset,
            col_num + col_offset,
            col_data,
            cell_format_default,
        )

# 合并合同编号列，如果连续的行数据相同，则合并
start_row = 0
for row_num, row_data in grouped.iterrows():
```

```
        current_value = row_data["合同编号']
        next_row = row_num + 1
        next_value = (
            grouped.iloc[next_row]["合同编号'] if next_row < len(grouped) else None
        )
        if current_value != next_value:
            log.info(f"{start_row} - {row_num} - {current_value}")
            if row_num - start_row > 0:
                worksheet.merge_range(
                    start_row + row_offset,
                    grouped.columns.get_loc("合同编号') + col_offset,
                    row_num + row_offset,
                    grouped.columns.get_loc("合同编号') + col_offset,
                    current_value,
                    cell_format_default,
                )
            start_row = next_row
```

版本号 #22
由 董列涛 创建于 11 三月 2025 19:11:35
由 董列涛 更新于 18 九月 2025 03:45:38