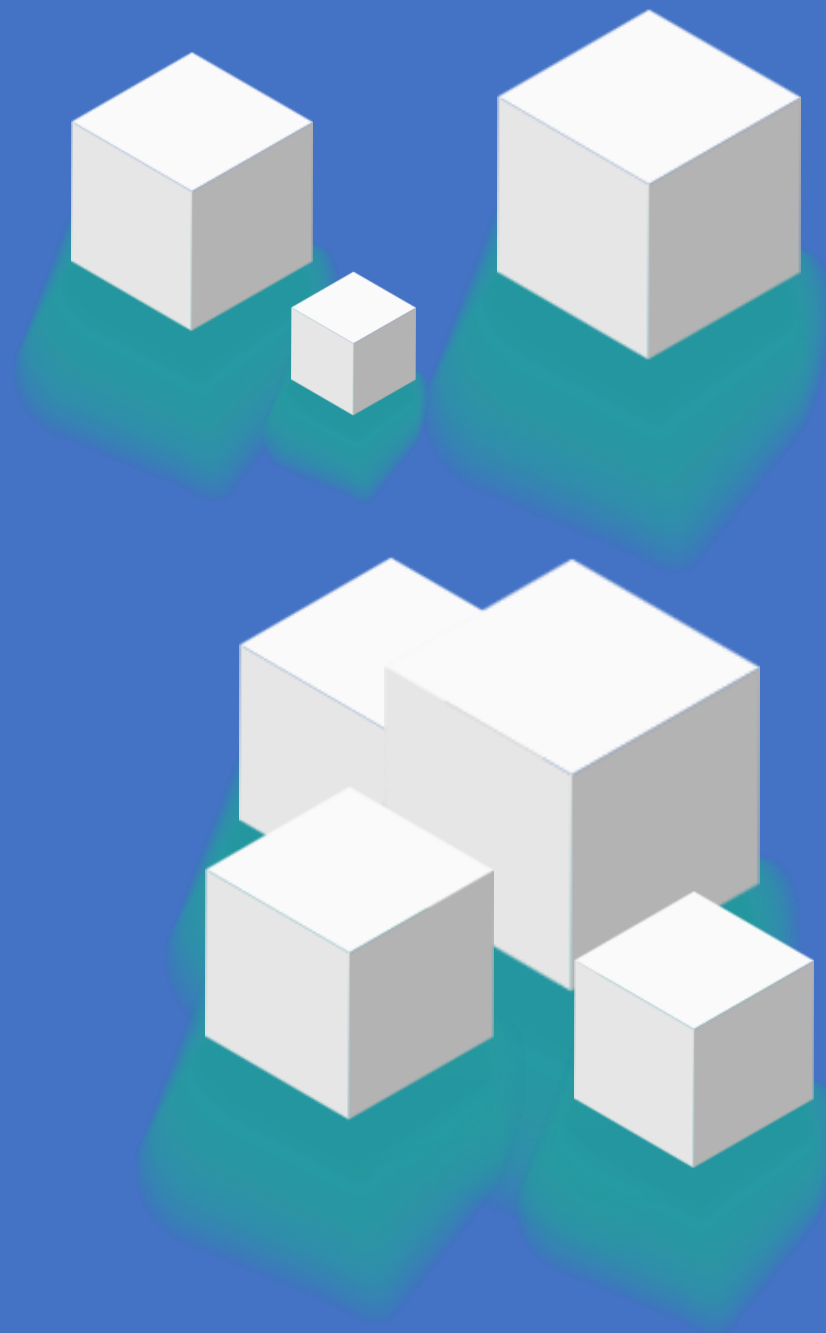
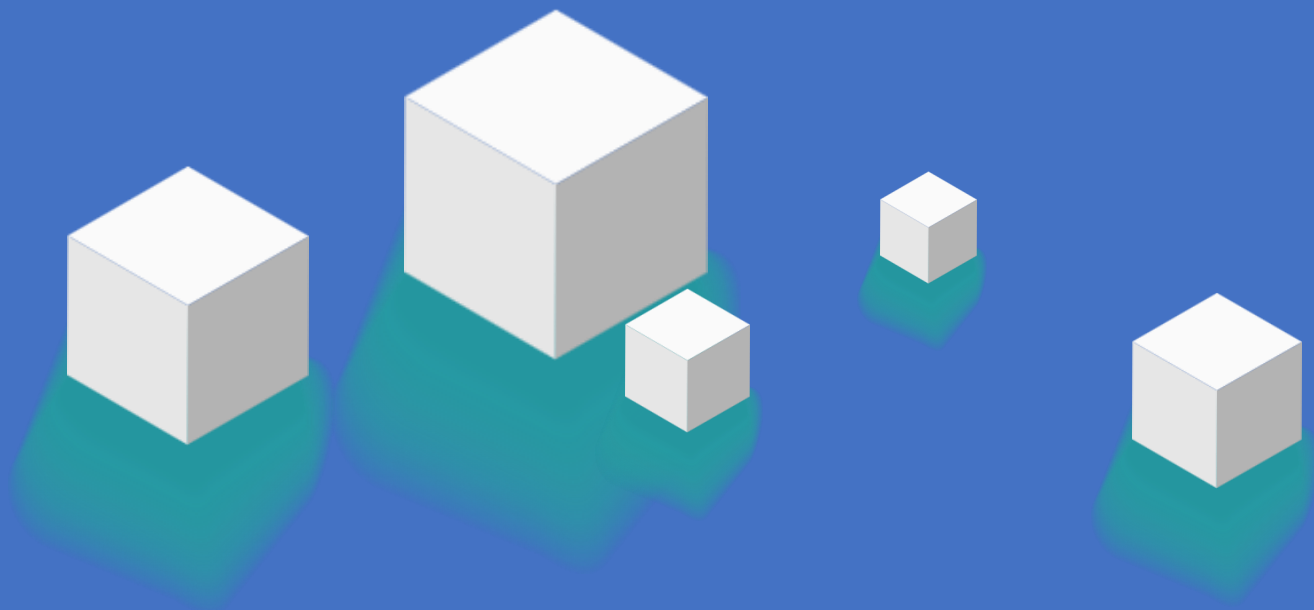


# Cursor编程



# >> 今天的学习目标

---

## Cursor编程

- 什么是Cursor Rules
- Cursor的主要功能
- CASE：多张Excel报表处理
- CASE：疫情实时监控
- Trae使用

# Cursor编程

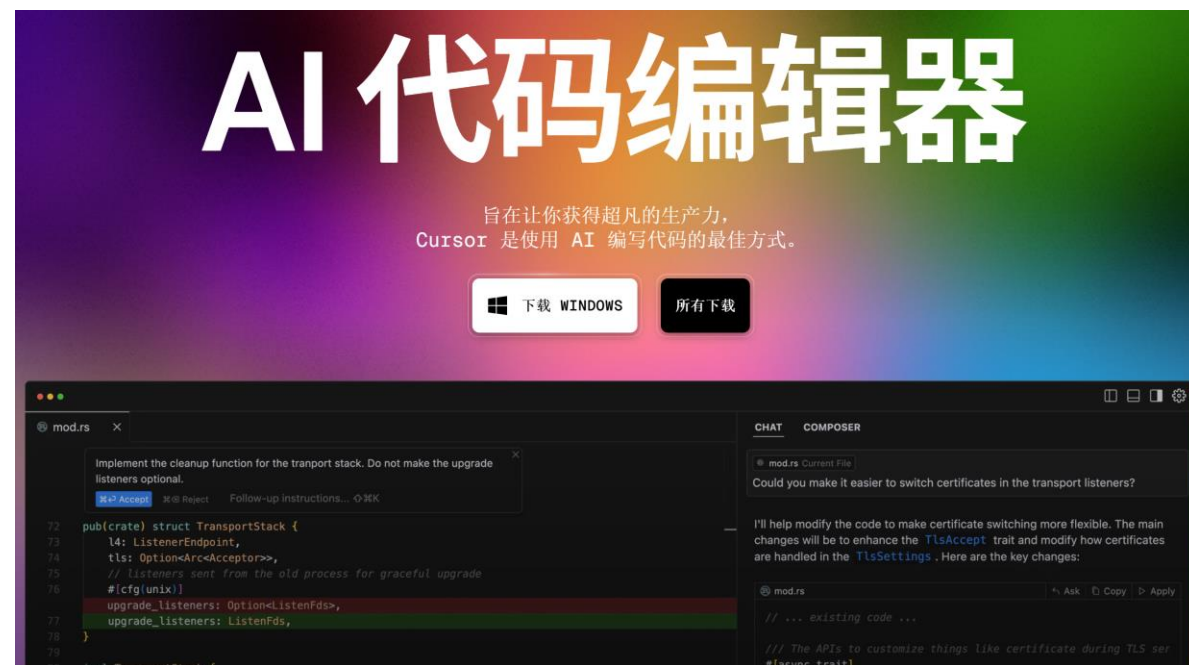
Cursor编辑器：

<https://cursor.com/>

Cursor 是基于 VSCode 开发的AI代码编辑器，提供智能代码补全、代码生成、代码修改、代码搜索和代码解释等。

与其他工具不同，Cursor 将 AI 辅助编码直接融入到编辑器的核心功能中，比如Cursor可以理解整个工程的代码，同时修改多个文件。

可以通过 .cursorrules 文件定制 AI 的行为。



# Cursor Rules

## Thinking: 什么是Cursor Rules?

Cursor Rules帮助你定制 AI 行为，让它符合你的编码风格和项目需求。

Cursor 提供两种类型的规则：

- **全局规则**

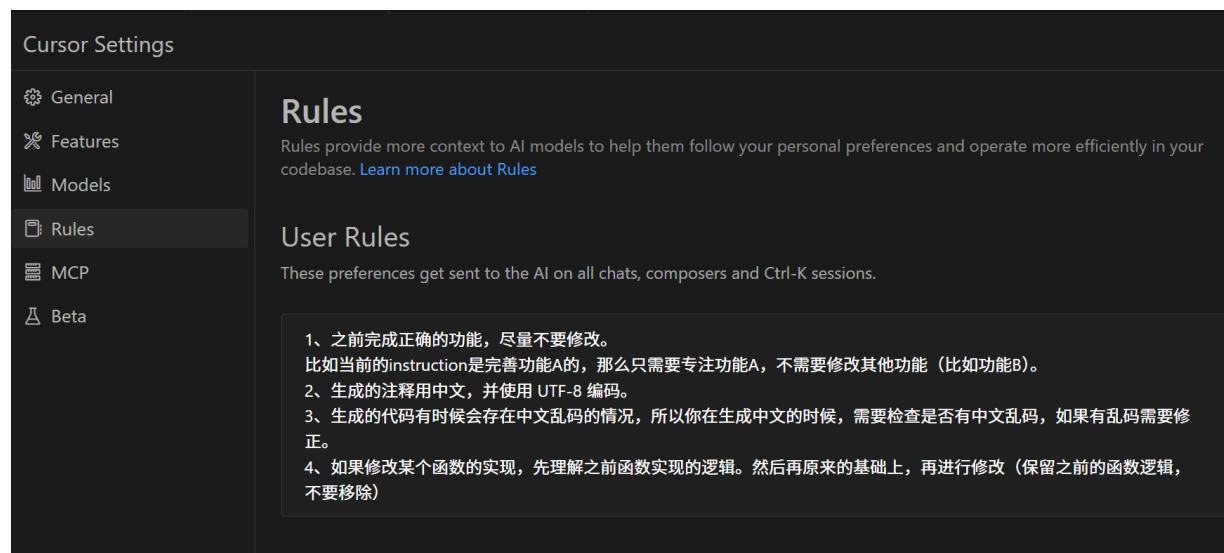
在 Cursor 设置的"通用" > "AI 规则"中配置。规则适用于你的所有项目，适合设置个人编码偏好和保持 AI 行为的一致性。

# 在 Cursor 设置 > AI 规则中

- 所有新代码使用 TypeScript
- 遵循代码整洁之道
- 优先使用 async/await 而不是回调
- 编写完善的错误处理

## 项目规则

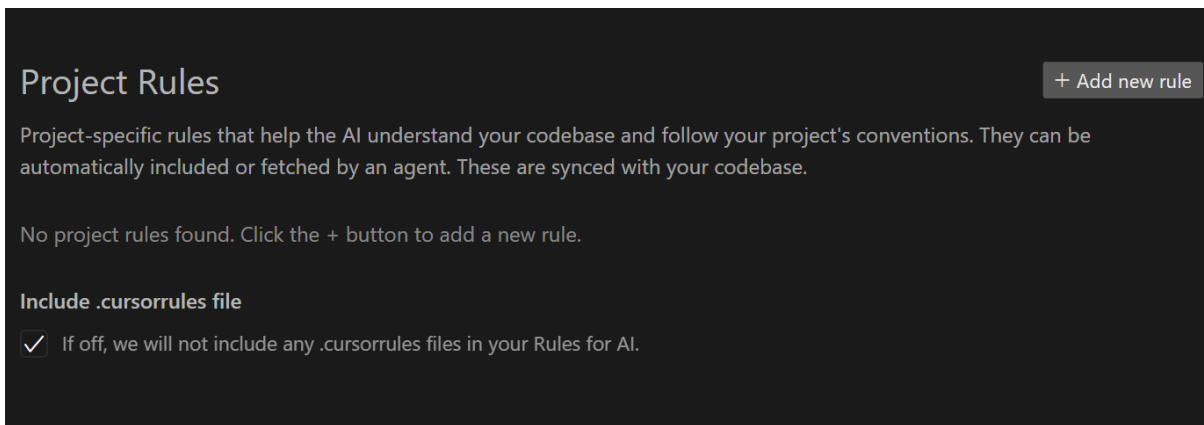
以 .mdc 格式存储在 .cursor/rules 目录中的项目特定规则。这些规则帮助 AI 理解你的代码库并遵循项目规范。通过 Cursor 设置 > 通用 > 项目规则来添加。它们会与你的代码库



# Cursor Rules

## 项目规则

仅对特定项目生效，通常存储在项目根目录的 `.cursorrules` 文件中。这些规则帮助 AI 理解你的代码库并遵循项目规范。通过 Cursor 设置 > 通用 > 项目规则来添加。



```
# 文件模式: *.tsx, *.ts
```

```
## React 规范
```

- 使用函数组件
- 实现完整的 prop 类型
- 遵循 React 最佳实践

```
## 样式
```

- 使用 Tailwind CSS
- 遵循团队样式指南

```
@file ../tsconfig.json
```

```
@file ../tailwind.config.js
```

项目规则优先级高于全局规则：如果项目中存在 `.cursorrules` 文件，则项目规则会覆盖全局规则。

全局规则：适合设置通用的编码规范、安全策略或个人偏好，确保所有项目保持一致。

项目规则：适合针对特定项目的个性化需求，例如特定技术栈、文件结构或安全要求。

# Cursor编程

---

Cursor的主要功能：

- Composer/Agent (Cmd + I) => 现已升级为Agent模式

允许同时编辑多个文件，并根据高级指令生成完整的应用程序，突破了单行和单文件编辑的限制。它能够理解项目结构上下文，并进行交互式代码优化。Composer 字面意思是作曲家，在 Cursor 中，它可以帮助你快速生成代码，如同自动驾驶，只需要告诉它你要做什么，它就可以帮你完成。

- 聊天功能 (Cmd + L)

用于针对更宽泛的代码问题进行对话，支持多轮对话，解答更广泛的编程问题

- 提示框功能(Cmd + K)

用于生成或修改局部的代码

# Cursor编程

---

Tab功能:

相比GitHub Copilot等辅助开发工具，Cursor的优势在于它的代码编辑能力，不仅可以插入代码，而且可以对现有的代码进行修改，这也是Cursor在官方文档中多次强调的

GitHub Copilot can insert text at your cursor position. It cannot edit the code around your cursor or remove text.

Cursor can insert text at your cursor, and much more:

— Multi-character edits

```
const updates: BlockUpdate[] = [];  
const [vx, ] = rigitContrrls.voxel;           const [vx, vy, vz] = rigidControls.voxel;
```

# CASE：多张Excel报表处理



# CASE：多张Excel报表处理

TO DO：两张Excel报表合并

比如 将员工基本信息表与员工绩效表合并，比如在主表基础上，添加员工2024年第4季度的绩效评分

员工ID	姓名	性别	部门	入职日期
1001	赵静	男	人力资源部	2020-08-29
1002	杨娜	女	人力资源部	2020-08-18
1003	黄芳	女	财务部	2020-06-16
1004	王强	女	财务部	2021-03-03
1005	张美丽	男	市场部	2021-04-21
1006	陈伟	女	市场部	2021-12-19
1007	李静	男	人力资源部	2022-04-02
1008	王芳	男	人力资源部	2020-02-10
1009	陈伟	男	财务部	2020-09-17
.....	.....	.....	.....	.....

员工基本信息表.xlsx

员工ID	年度	季度	绩效评分
1001	2024	1	3.53
1001	2024	2	3.57
1001	2024	3	4.67
1001	2024	4	4.41
1002	2024	1	4.9
1002	2024	2	3.97
1002	2024	3	4.08
1002	2024	4	4.81
1003	2024	1	3.56

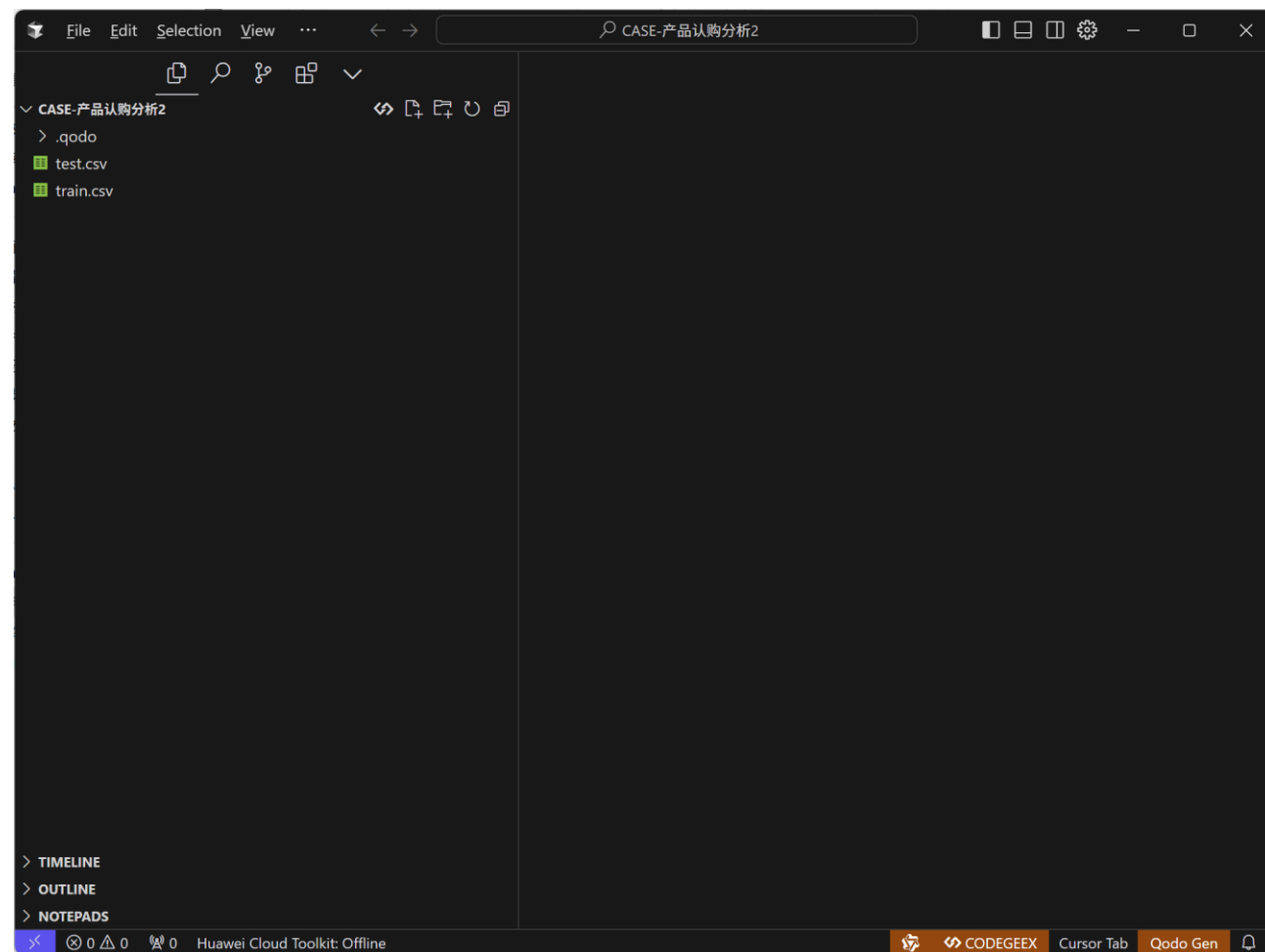
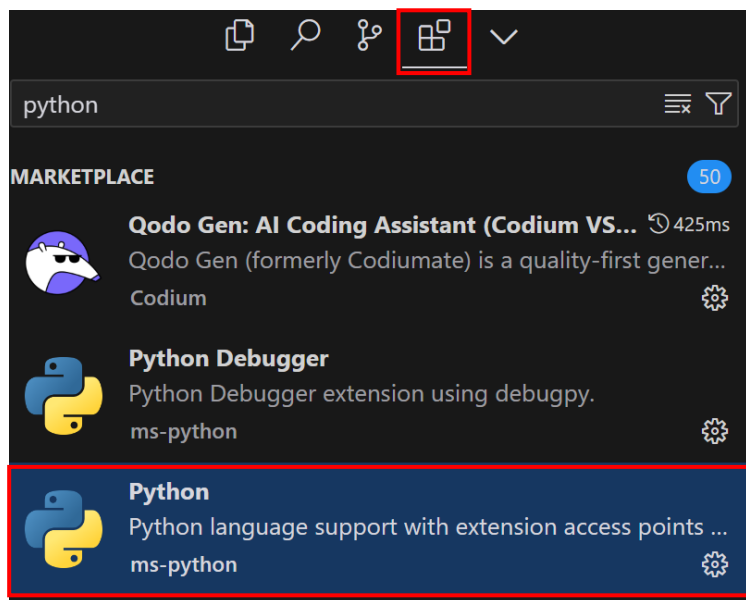
员工绩效表.xlsx

# Cursor使用

Step1, 使用Cursor, 打开对应的文件夹

<https://www.cursor.com/cn>

- 1) 下载对应的版本, 然后打开本地的项目文件夹
- 2) 配置python编译环境



# Cursor使用

---

Cursor 辅助AI编程提供了三个主要功能:

- 跨文件编辑 (Cmd/Ctrl + I)

AI帮你创建代码、修改代码

- 聊天功能 (Cmd/Ctrl + L)

与AI交流方案, 解释逻辑

- 提示框功能(Cmd/Ctrl + K)

向AI提问, 让它执行特定功能

Cursor具有文件管理的更高权限, 可以看到文件夹下所有的文件, 并进行编辑/创建

可以对项目做背景说明, 比如放到文件夹, 或者 Notepads中

跨文件编辑功能 (Composer) 可实现多文件协同修改, 适合复杂项目开发。

# Cursor中的DeepSeek配置

点击 File => Preferences => Cursor Settings中的Models

添加模型 Add Models, 创建deepseek-r1和deepseek-v3

因为这个需要调用第三方的API, 所以打开下方的OpenAI API Key, 并进行相应设置:

Override OpenAI Base URL:

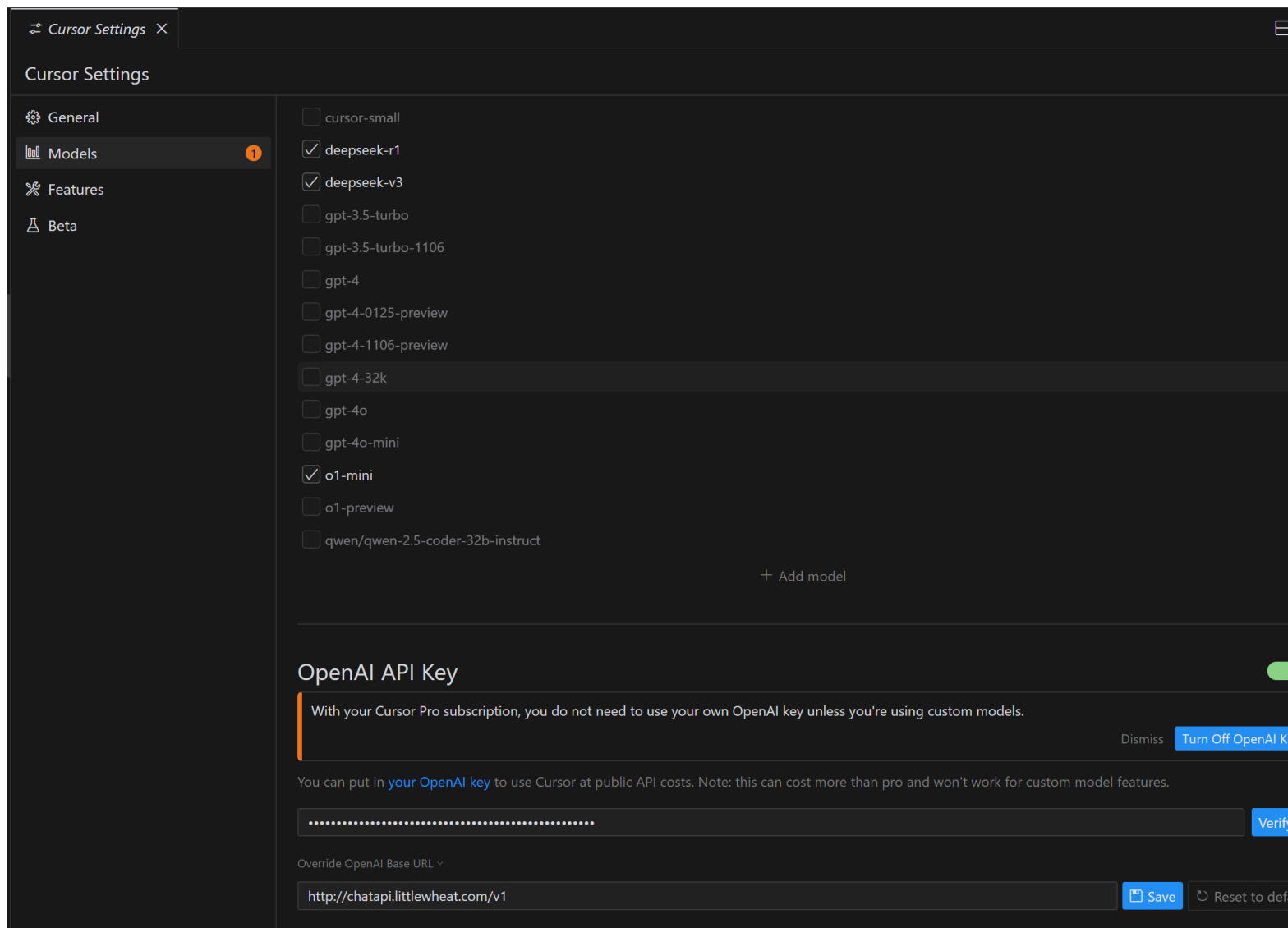
<http://chatapi.littlewheat.com/v1>

your OpenAI Key:

sk-

Q2gN9CgZOz9jrzjCCHkijalkUyaXpHS6xssmmkl

327kkib0G




# CASE：多张Excel报表处理

Step2，读取两张数据表的字段含义

编写Python，打开 员工基本信息表.xlsx 和 员工绩效表.xlsx，查看对应的字段

```
# -*- coding: utf-8 -*-
import pandas as pd

# 安装依赖库 (如果尚未安装)
# !pip install openpyxl pandas

Qodo Gen: Options | Test this function | 

def read_excel_columns(file_path):
    """读取Excel文件并返回列名"""
    try:
        df = pd.read_excel(file_path, engine='openpyxl')
        return df.columns.tolist()
    except Exception as e:
        print(f"读取文件{file_path}时出错: {e}")
        return []

# 读取员工基本信息表
basic_info_columns = read_excel_columns("员工基本信息表.xlsx")
print("员工基本信息表字段: ")
print('\n'.join(basic_info_columns))

# 读取员工绩效表
performance_columns = read_excel_columns("员工绩效表.xlsx")
print("\n员工绩效表字段:")
print('\n'.join(performance_columns))
```

<think>

好的，我现在需要帮用户编写Python代码来打开两个Excel文件，查看它们的字段。首先，我得确定用户的需求是什么。他们可能想合并这两个表，或者进行数据分析，但当前的任务只是查看字段，所以不需要做进一步的处理。

接下来，我需要考虑使用哪个库来处理Excel文件。常用的有pandas，因为它可以方便地读取Excel并展示数据结构。用户可能没有安装所需的库，所以代码中应该包含安装openpyxl和pandas的指令，因为这两个是处理xlsx文件必需的.....

点击Accept File，并运行Python

# CASE：多张Excel报表处理

运行后可以看到数据表的字段

员工基本信息表字段：

员工ID

姓名

性别

部门

入职日期

员工绩效表字段：

员工ID

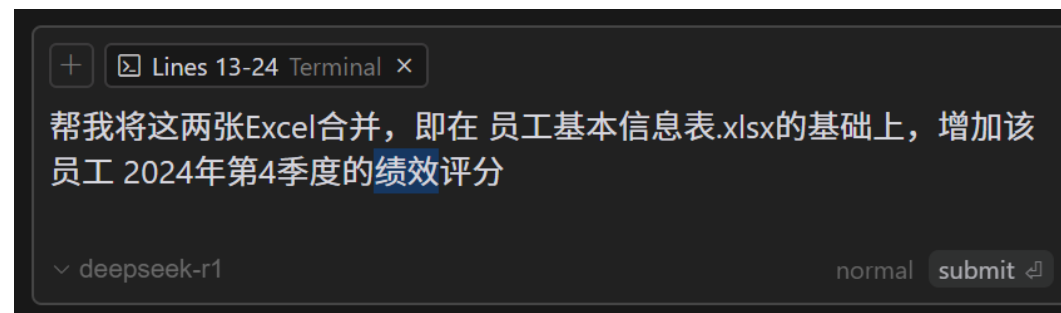
年度

季度

绩效评分

Step3，合并两张数据表

帮我将这两张Excel合并，即在 员工基本信息表.xlsx的基础上，增加该员工 2024年第4季度的绩效评分



需要将Terminal输出的字段选中，AI既可以看到这些字段含义，然后下指令将数据表进行合并

# CASE：多张Excel报表处理

```
merge_employee_data.py > merge_employee_data
1  #-*- coding: utf-8 -*-
2  import pandas as pd
3  from datetime import datetime
4
5  Qodo Gen: Options | Test this function |
6  def merge_employee_data():
7      try:
8          # 读取原始数据
9          basic_df = pd.read_excel("员工基本信息表.xlsx", engine='openpyxl')
10         performance_df = pd.read_excel("员工绩效表.xlsx", engine='openpyxl')
11
12         # 过滤2024年第4季度绩效数据
13         q4_2024 = performance_df[(performance_df['年度'] == 2024) &
14                                   (performance_df['季度'] == 4)]
15
16         # 合并数据 (左连接以保留所有员工)
17         merged_df = pd.merge(
18             basic_df,
19             q4_2024[['员工ID', '绩效评分']], # 只保留需要的字段
20             on='员工ID',
21             how='left'
22         )
23
24         # 处理空值并重命名列
25         merged_df['绩效评分'] = merged_df['绩效评分'].fillna('暂无数据')
26         merged_df.rename(columns={'绩效评分': '2024Q4绩效评分'}, inplace=True)
27
28         # 生成带时间戳的文件名
29         timestamp = datetime.now().strftime("%Y%m%d%H%M")
30         output_file = f"员工信息综合表_{timestamp}.xlsx"
```



	A	B	C	D	E	F
1	员工ID	姓名	性别	部门	入职日期	2024Q4绩效评分
2	1001	赵静	男	人力资源部	2020-08-29	4.41
3	1002	杨娜	女	人力资源部	2020-08-18	4.81
4	1003	黄芳	女	财务部	2020-06-16	3.75
5	1004	王强	女	财务部	2021-03-03	3.41
6	1005	张美丽	男	市场部	2021-04-21	3.62
7	1006	陈伟	女	市场部	2021-12-19	4.24
8	1007	李静	男	人力资源部	2022-04-02	4.83
9	1008	王芳	男	人力资源部	2020-02-10	4.09
10	1009	陈伟	男	财务部	2020-09-17	3.5
11	1010	周静	女	销售部	2020-06-01	3.65
12	1011	王建国	男	人力资源部	2020-10-29	4.55
13	1012	刘亮	女	市场部	2020-01-21	4.86
14	1013	陈芳	男	销售部	2021-10-08	4.39
15	1014	王敏	女	市场部	2021-11-10	4.16
16	1015	陈静	女	市场部	2021-02-15	4.25
17	1016	王亮	男	销售部	2021-08-07	3.39
18	1017	吴美丽	女	产品部	2020-01-27	4.68
19	1018	杨建国	女	市场部	2020-11-17	3.56
20	1019	李强	女	人力资源部	2020-10-22	3.08
21	1020	杨强	女	人力资源部	2020-05-18	3.37

DeepSeek-R1 中间生成的代码，用于执行任务

运行后，得到了合并结果

Thinking：不会Python，是否可以做复杂的Excel报表处理？需要自己手写vlookup函数么？

不需要限制DeepSeek使用的工具，业务人员可以从需求和结果上进行把控，如果想要了解DeepSeek处理过程，可以和它进行交流

# CASE: 疫情实时监控



# CASE：疫情实时监控

---

CASE：疫情实时监控

数据表 data.xlsx，原始数据见 [https://github.com/cystanford/dashboard\\_epidemic](https://github.com/cystanford/dashboard_epidemic)

整合了疫情的关键指标数据，适用于疫情可视化大屏展示及数据分析。

数据表记录了香港各区每日疫情变化情况，包含确诊、康复、死亡等核心指标，以及区域风险评估结果。

TO DO：

- 1) 确诊病例数：每日新增与累计确诊数据
- 2) 地理分布图：各区域疫情分布及热点区域标识
- 3) 趋势分析图：病例增长趋势、增长率变化图表

.....

# CASE：疫情实时监控

字段名称	数据类型	中文名称	说明
report_date	日期	报告日期	疫情数据的统计日期，格式：YYYY-MM-DD
district_name	文本	地区名称	香港18个行政区的名称
new_confirmed	整数	新增确诊	当日新增确诊病例数
total_confirmed	整数	累计确诊	截至当日累计确诊病例总数
active_cases	整数	现存确诊	当前仍在治疗的确诊病例数（累计确诊减去康复和死亡）
new_recovered	整数	新增康复	当日新增康复出院病例数
total_recovered	整数	累计康复	截至当日累计康复出院病例总数
new_deaths	整数	新增死亡	当日新增死亡病例数
total_deaths	整数	累计死亡	截至当日累计死亡病例总数
incidence_rate	浮点数	发病率(每10万人)	每10万人口中的现存确诊病例数，反映疫情严重程度
population	整数	人口	该地区的常住人口数量
risk_level	文本	风险等级	地区疫情风险评估结果，分为"低风险"、"中风险"、"高风险"

# CASE：疫情实时监控

报告日期	地区名称	新增确诊	累计确诊	现存确诊	新增康复	累计康复	新增死亡	累计死亡	发病率(每10万人)	人口	风险等级
2022-01-01	中西区	4	4	4	0	0	0	0	1.646091	243000	低风险
2022-01-01	湾仔区	0	0	0	0	0	0	0	0	152000	低风险
2022-01-01	东区	0	0	0	0	0	0	0	0	550000	低风险
2022-01-01	南区	0	0	0	0	0	0	0	0	270000	低风险
2022-01-01	油尖旺区	0	0	0	0	0	0	0	0	328000	低风险
2022-01-01	深水埗区	0	0	0	0	0	0	0	0	405000	低风险
2022-01-01	九龙城区	0	0	0	0	0	0	0	0	418000	低风险
2022-01-01	黄大仙区	0	0	0	0	0	0	0	0	420000	低风险
2022-01-01	观塘区	1	1	1	0	0	0	0	0.154321	648000	低风险
2022-01-01	葵青区	0	0	0	0	0	0	0	0	511000	低风险
2022-01-01	荃湾区	0	0	0	0	0	0	0	0	307000	低风险
2022-01-01	屯门区	0	0	0	0	0	0	0	0	488000	低风险
2022-01-01	元朗区	0	0	0	0	0	0	0	0	619000	低风险
2022-01-01	北区	0	0	0	0	0	0	0	0	307000	低风险
2022-01-01	大埔区	0	0	0	0	0	0	0	0	310000	低风险
2022-01-01	沙田区	1	1	1	0	0	0	0	0.151515	660000	低风险
2022-01-01	西贡区	0	0	0	0	0	0	0	0	461000	低风险
2022-01-01	离岛区	0	0	0	0	0	0	0	0	158000	低风险
2022-01-02	中西区	2	6	6	0	0	0	0	2.469136	243000	低风险
*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****

# CASE：疫情实时监控

编写Python，查看 香港各区疫情数据\_20250322.xlsx 的字段和前20行数据

```
=====
文件名: 香港各区疫情数据_20250322.xlsx
总行数: 3240
=====
```

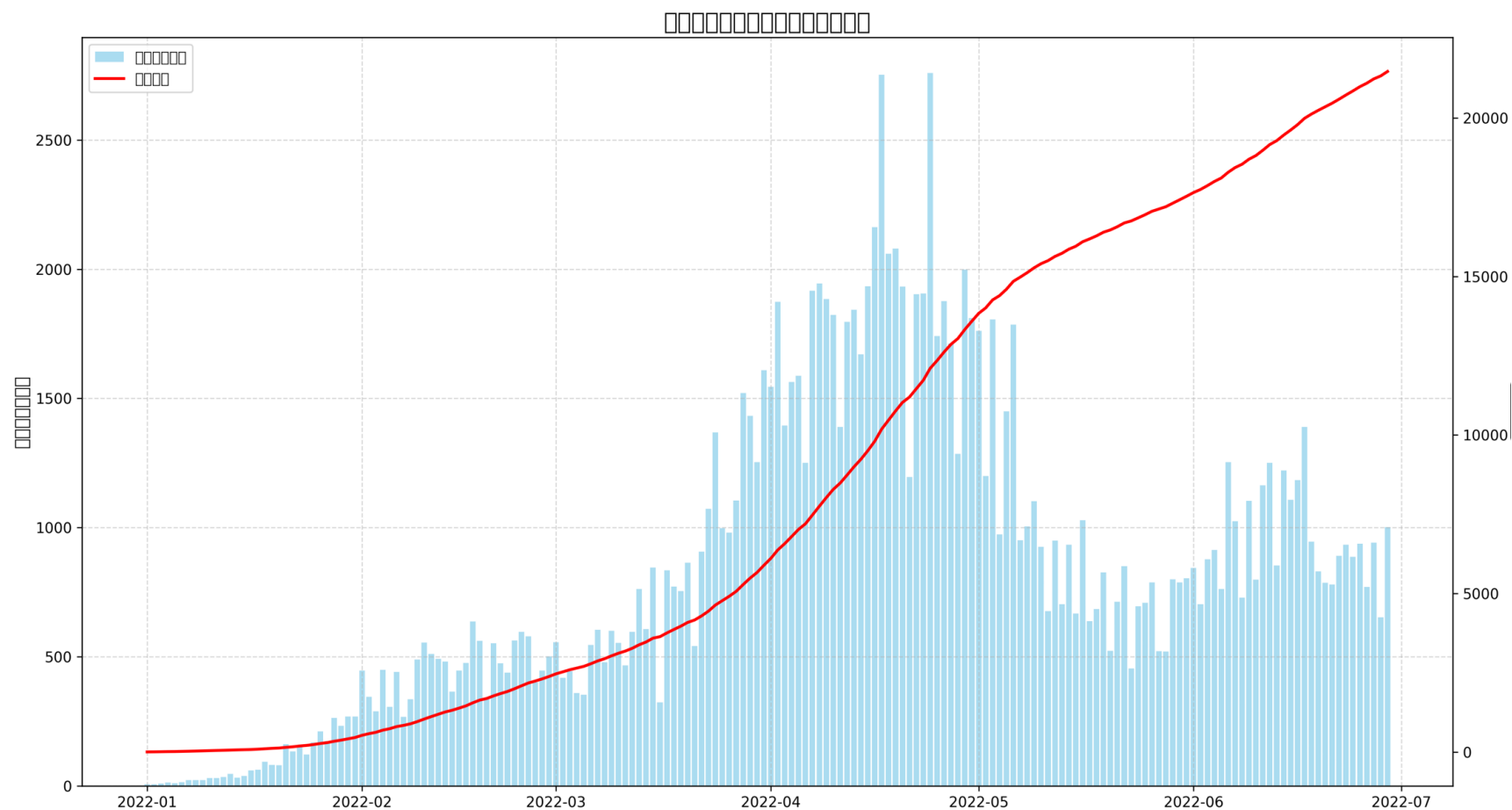
- 字段列表:
- 1. 报告日期
  - 2. 地区名称
  - 3. 新增确诊
  - 4. 累计确诊
  - 5. 现存确诊
  - 6. 新增康复
  - 7. 累计康复
  - 8. 新增死亡
  - 9. 累计死亡
  - 10. 发病率(每10万人)
  - 11. 人口
  - 12. 风险等级

前20行数据:

	报告日期	地区名称	新增确诊	累计确诊	现存确诊	新增康复	累计康复	新增死亡	累计死亡	发病率(每10万人)	人口	风险等级
0	2022-01-01	中西区	4	4	4	0	0	0	0	1.646091	243000	低风险 .....

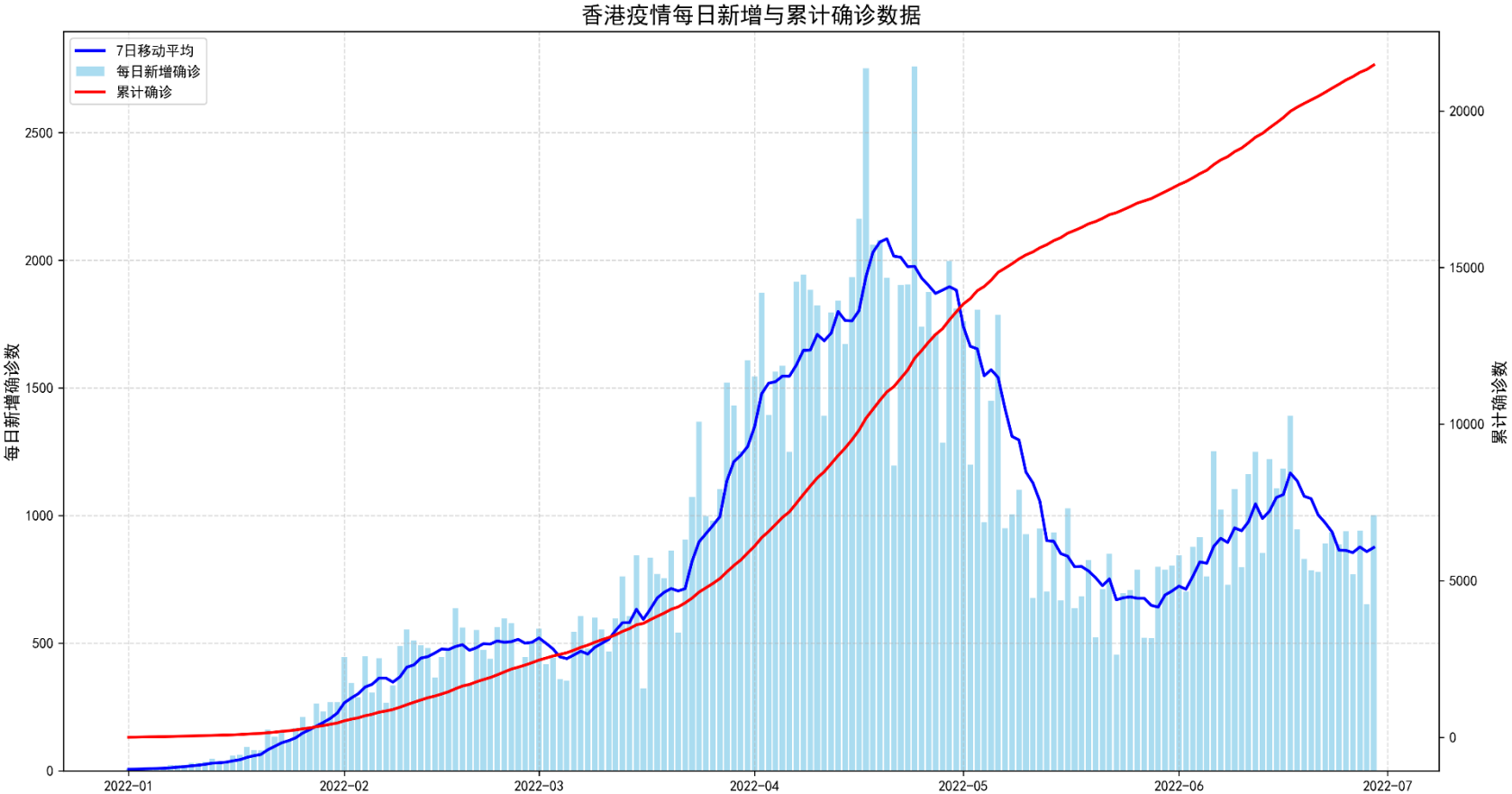
# CASE：疫情实时监控

帮我计算确诊病例数：每日新增与累计确诊数据



# CASE：疫情实时监控

图表中的中文有乱码



# CASE：疫情实时监控

我想创建可视化大屏，包括但不限于：

- 1) 确诊病例数：每日新增与累计确诊数据
- 2) 地理分布图：各区域疫情分布及热点区域标识
- 3) 趋势分析图：病例增长趋势、增长率变化图表

===

帮我安排有哪些适合的图表，如何在一个可视化大屏上进行呈现，可以使用echarts



香港疫情可视化大屏设计方案

基于您的需求，我将设计一套完整的疫情可视化大屏，使用echarts实现。这个大屏将以直观、高效的方式展示香港疫情数据，包括确诊病例数据、地理分布和趋势分析。

.....

这里使用 Ctrl + L （Chat模式）

# CASE：疫情实时监控

---

## 大屏总体布局

大屏采用分区块布局，可以分为以下几个主要部分：

顶部标题栏：显示"香港疫情实时监控大屏"及更新时间

中心区域：包含地图和主要图表

两侧辅助区域：显示关键指标和辅助图表

底部趋势区：展示长期趋势分析

## 具体图表规划

### 1. 核心指标展示（顶部卡片）

今日新增确诊：数字展示 + 与昨日对比的箭头指示

累计确诊总数：数字展示

当前活跃病例：数字展示

累计康复人数：数字展示

累计死亡人数：数字展示

重点关注区域：列出当日确诊人数最多的区域



# CASE：疫情实时监控

---

## 2. 地理分布图（中心区域左侧）

香港地区热力图：

以香港地图为底图

根据各区确诊数据显示不同深浅的热力覆盖

鼠标悬停显示区域名称、累计确诊、新增确诊等详细信息

点击区域可查看该区域的详细数据面板

## 3. 每日新增与累计确诊数据（中心区域右侧上部）

双Y轴折线+柱状图：

左Y轴：每日新增确诊（柱状图）

右Y轴：累计确诊（折线图）

X轴：日期（最近30天）

新增柱状图上添加7日移动平均线

# CASE：疫情实时监控

---

## 4. 各区域疫情对比（右侧中部）

横向柱状图：

按确诊数量从高到低排列各区域

不同风险等级用不同颜色表示

可切换显示"累计确诊"、"新增确诊"、"每10万人口确诊率"

## 5. 趋势分析图（底部）

增长率变化曲线图：

显示每日新增确诊的环比增长率

添加移动平均线平滑波动

设置0线，直观表示增长与下降

活跃病例趋势图：

展示当前活跃病例随时间变化的趋势

可与康复、死亡趋势叠加对比

# CASE：疫情实时监控

---

## 6. 风险预警看板（右下角）

风险等级分布饼图：

各区域按风险等级（高、中、低）分布

显示各风险等级区域数量及占比

关键指标预警：

高增长率区域列表

医疗资源紧张程度指示器

## 交互功能设计

时间选择器：可选择查看不同时间段的数据

区域筛选器：可选择特定区域或多区域组合查看

数据下钻：从总体视图点击进入详细信息

指标对比：可选择不同指标进行对比分析

数据导出：可导出图表或原始数据

.....

# CASE：疫情实时监控

帮我按照刚才的大屏设计，完成flask + echarts 的可视化大屏搭建

我将帮您创建一个基于Flask和ECharts的疫情可视化大屏。首先，让我了解一下项目文件结构。

📁 Listed 12 items in current directory

现在我需要查看现有的Excel数据处理代码，了解数据结构：

👁 Read read\_excel.py

现在我将创建Flask项目结构和实现可视化大屏。我们将基于现有的数据处理函数来构建Web应用。

首先，创建必要的目录结构：

```
mkdir -p static/css static/js static/images
templates
```

```
PS D:\企业培训\香港医管局\CASE-疫情实时监控-1> mkdir -p static/css static/js sta
mkdir : 找不到接受实际参数“static/js”的位置形式参数。
```

```
所在位置 行:1 字符: 1
```

```
+ mkdir -p static/css static/js static/images templates
```

```
+ ~~~~~
```

```
+ CategoryInfo          : InvalidArgument: (:) [mkdir], ParameterBindingE>
```

```
PS D:\企业培训\香港医管局\CASE-疫情实时监控-1>
```

# CASE: 疫情实时监控

现在我们创建主要的Flask应用文件:

```
app.py +161 -1 ✓
from flask import Flask, render_template, jsonify
import pandas as pd
import json
from datetime import datetime
import numpy as np

app = Flask(__name__)

# 数据文件路径
DATA_FILE = "香港各区疫情数据_20250322.xlsx"

# 加载和处理数据
def load_data():
    """加载并处理Excel数据, 返回处理后的数据集"""
```

接下来, 让我们创建HTML模板文件:

```
index.html (new) +92 -1 •
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>香港疫情实时监控大屏</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/dashboard.'
    <!-- 引入 ECharts -->
    <script src="https://cdn.bootcdn.net/ajax/libs/echarts/5.3.3/echarts.min.j
</head>
<body>
    <div class="dashboard-container">
        <!-- 顶部标题栏 -->
        <div class="header">
```

# CASE: 疫情实时监控

现在，我们创建CSS样式文件:

```
# dashboard.css +183 -1 ✓  
  
/* 全局样式 */  
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  font-family: "Microsoft YaHei", "PingFang SC", "Hiragino Sans GB", sans-serif;  
}  
  
body {  
  width: 100%;  
  height: 100vh;  
  overflow: hidden;  
  background-color: #0f1c3a;  
  color: #fff;
```

接下来，创建JavaScript文件，实现各个图表的数据加载和渲染:

```
JS dashboard.js (new) +668 -1 •  
  
// 初始化所有图表  
document.addEventListener('DOMContentLoaded', function() {  
  // 获取数据并初始化所有图表  
  Promise.all([  
    fetch('/api/daily_data').then(res => res.json()),  
    fetch('/api/region_data').then(res => res.json()),  
    fetch('/api/summary').then(res => res.json())  
  ]).then(([dailyData, regionData, summaryData]) => {  
    // 初始化各图表  
    initDailyTotalChart(dailyData);  
    initAreaComparisonChart(regionData);  
    initGrowthRateChart(dailyData);  
    initActiveCasesChart(dailyData);  
    initRiskLevelPie(summaryData, regionData);  
  });  
});
```

# CASE: 疫情实时监控

python app.py runserver

http://127.0.0.1:5000



# CASE：疫情实时监控

大屏可以更长一些，现在右侧的很多图表都重叠到一起了，需要合理化布局

我看到了大屏布局问题，右侧图表确实存在重叠。我将修改CSS样式，优化大屏布局，让所有图表能够合理显示而不重叠。

```
# dashboard.css +36 -12
.dashboard-container {
  width: 100%;
  height: 100vh;
  overflow: hidden;
  overflow: auto;
  display: grid;
  grid-template-rows: 60px 120px 1fr 280px;
  grid-template-rows: 60px 120px minmax(400px, 1fr) minmax(300px, auto);
  grid-template-columns: 1fr;
  gap: 12px;
  padding: 12px;
  gap: 15px;
  padding: 15px;
```

我还需要修改HTML模板，调整风险区域部分的布局，以确保与CSS变更保持一致：

```
<> index.html +2 -1
</div>
</div>
<!-- 右下角风险区 -->
<!-- 风险区域 - 调整为并排布局 -->
<div class="risk-container">
  <div class="chart-wrapper">
    <div class="chart-title">风险等级分布</div>
    <script src="{{ url_for('static', filename='js/dashboard.js') }}"></script>
</body>
</html>
```



# CASE：疫情实时监控

## 香港疫情实时监控大屏

最后更新时间: 2022-06-29

今日新增确诊

1002

累计确诊总数

21466

当前活跃病例

1146

累计康复人数

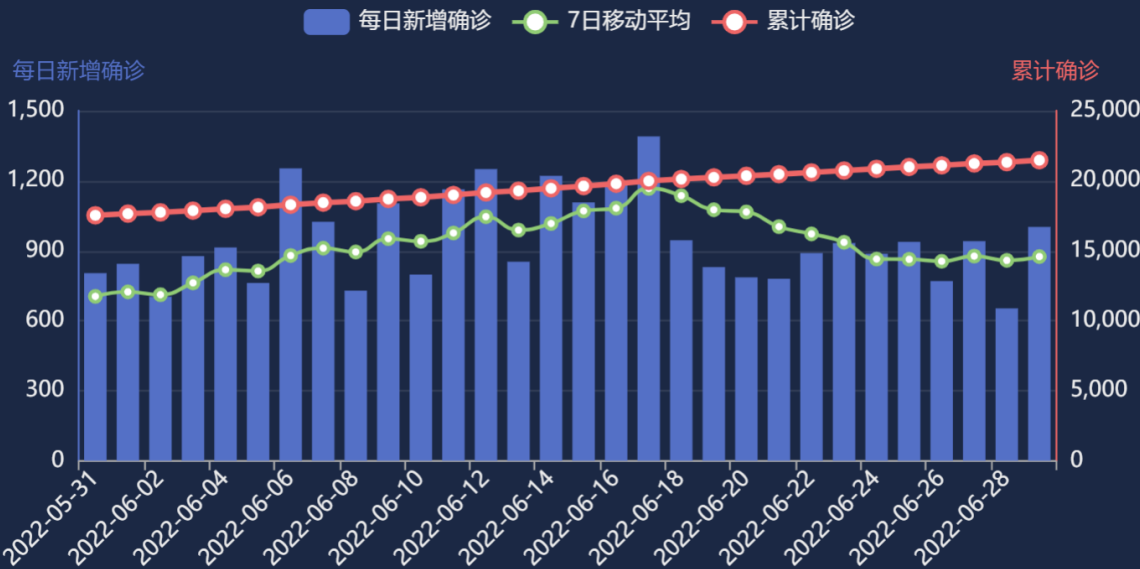
20184

累计死亡人数

136

### 香港各区疫情分布

### 每日新增与累计确诊数据



# CASE：疫情实时监控



比原来好多了，但是细节上还是需要进行调整

# CASE：疫情实时监控

香港地图没有显示出来，能否帮我找到香港的Echarts地图



我注意到目前的代码中使用了简化的点数据来模拟香港地图，而不是真正的地理数据，导致香港地图未能正确显示。我来帮你实现一个真实的香港地图可视化。

 Confirm search for "ECharts 香港地图 GeoJSON"

☐ Auto-search

Cancel

Continue

# CASE: 疫情实时监控

根据我的搜索结果，我发现可以使用阿里云DataV.GeoAtlas来获取香港地图的GeoJSON数据，这个工具非常适合ECharts使用。我将获取香港地图数据并帮你实现地图可视化。

🔍 Searched web for "阿里云 DataV.GeoAtlas 香港地图 GeoJSON" "阿里云 DataV.GeoAtlas 香港..."

根据搜索结果，我发现可以从阿里云DataV.GeoAtlas获取香港地图的GeoJSON数据。现在我会直接访问DataV.GeoAtlas网站查找香港的GeoJSON数据：

🔍 Confirm search for "https://datav.aliyun.com/tools/atlas/#&lat=22.5&lng=114.0&zoom=9 香港地图"

☐ Auto-search

Cancel

Continue

让我创建一个完整的解决方案来显示香港地图。我将使用从阿里云DataV.GeoAtlas获取的香港GeoJSON数据，这是实现香港地图可视化的最佳方式。

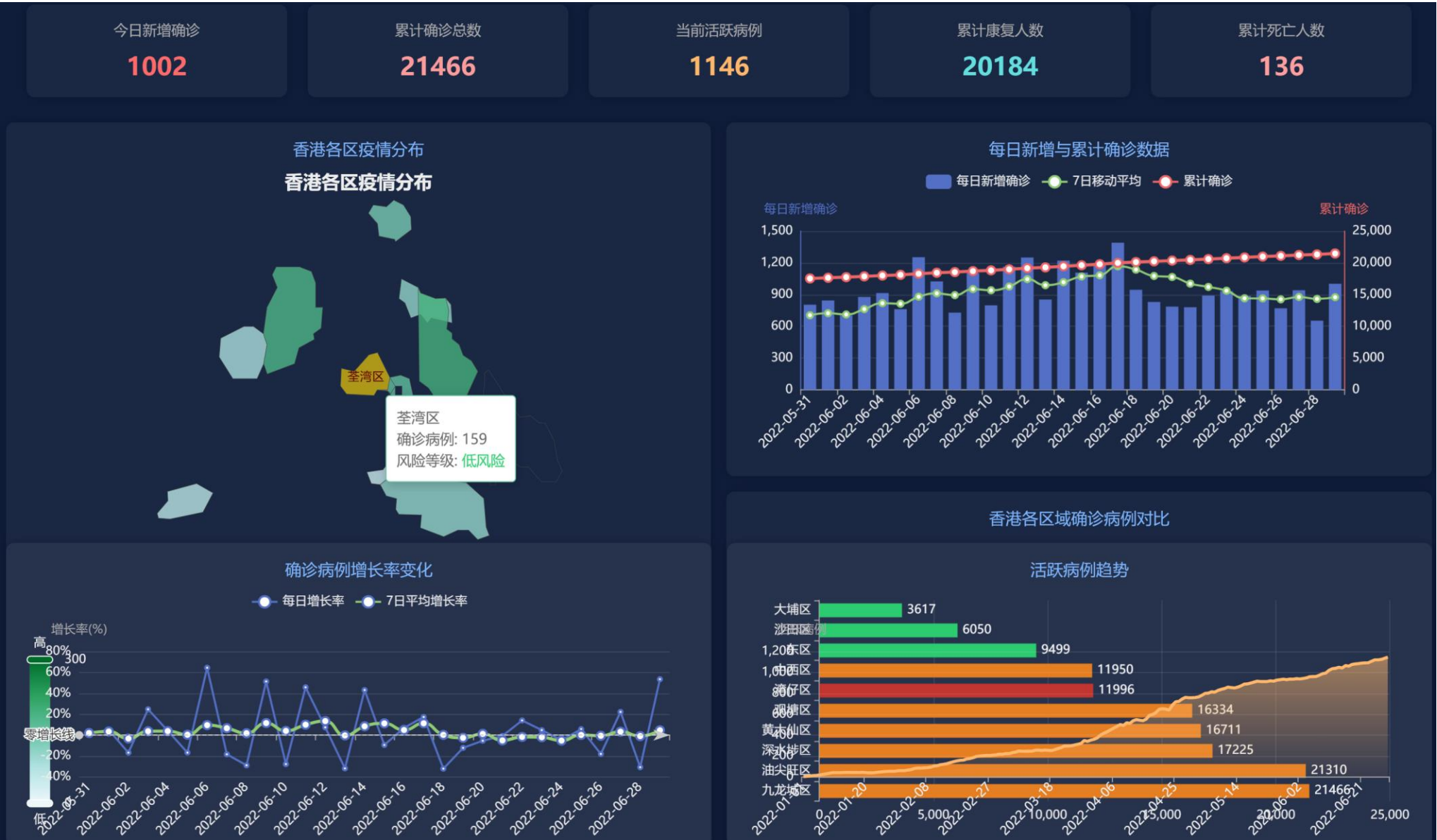
```
{ } hongkong.json (new) +311 -1 •
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "adcode": 810001,
        "name": "中西区",
        "center": [114.154334, 22.282055],
        "centroid": [114.15392, 22.28641],
        "childrenNum": 0,
        "level": "district",
        "acroutes": [100000, 810000],
        "parent": { "adcode": 810000 }
      }
    }
  ]
}
```

现在我将更新JavaScript代码，将香港地图添加到我们的应用程序中：

```
JS dashboard.js +132 -22 •
// 先加载香港地图JSON数据
fetch('/static/js/hongkong.json')
  .then(response => response.json())
  .then(hongkongJson => {
    // 注册香港地图
    echarts.registerMap('hongkong', hongkongJson);

    // 获取数据并初始化所有图表
    Promise.all([
      fetch('/api/daily_data').then(res => res.json()),
      fetch('/api/region_data').then(res => res.json()),
      fetch('/api/summary').then(res => res.json())
    ]).then(([dailyData, regionData, summaryData]) => {
      // 初始化各图表
    })
  })
}
```

# CASE：疫情实时监控



# CASE：疫情实时监控

地图形状是有了，能否放到地图上更好看一些。现在就是一些地理区域的方块

我将优化香港地图的样式，让它看起来更加自然和美观，而不是简单的方块区域。

```
JS dashboard.js +90 -1 ✓
    return `${params.name}<br/>确诊病例: ${data.value}<br/>风险:
  }
  return params.name;
},
backgroundColor: 'rgba(50,50,50,0.7)',
borderColor: '#ccc',
borderWidth: 1,
textStyle: {
  color: '#fff'
}
},
visualMap: {
  color: '#eee'
```



Cursor不一定每次都能得到满意的结果，还需要精细的调修

# CASE：疫情实时监控

帮我查看香港地图的呈现，现在出现了两个重复的地图，都可以拖动了。  
另外样式上，可以参考 百度地图，或者有中国地图的区域背景，因为现在没有后面的区域背景，看不出来完整的地图样式

修改内容说明：

1. 移除了重复显示地图的 `geo` 配置，这是导致出现两个地图的原因。
2. 增加了地图的背景色，设置为蓝色系（类似百度地图的风格），使地图看起来更有空间感。
3. 添加了 `layoutCenter` 和 `layoutSize` 属性，确保地图居中并适当放大。
4. 增加了 `aspectScale` 属性调整地图的长宽比。
5. 在默认状态下显示地区名称标签，提高可识别性。
6. 使用更自然的颜色渐变和边界样式，增强地图的美观度。

这些修改应该能使香港地图看起来更加完整和美观，并解决双重显示的问题。

使用Ctrl + L（Chat模式）更适合精细的调修，AI不会一上来就进入到修改模式



# CASE: 疫情实时监控



Thinking: 地图的问题如何进行调整?



# CASE：疫情实时监控

地区之间有很多空隙，比如元朗区和荃湾区，另外大埔区和沙田区有重叠的区域了。帮我查看地图的数据，中间很多没有相连，空隙太大了。

我注意到了香港地图显示的问题，地区之间确实存在较大空隙，且有些区域如大埔区和沙田区出现了重叠。这是因为当前使用的GeoJSON数据精度不够或坐标点可能存在问题。

让我分析一下香港地图数据并提供解决方案：

## 1. 现有GeoJSON数据问题分析：

- 空隙问题：当前数据中各区域的边界点可能不够密集，导致渲染时出现空隙
- 重叠问题：大埔区和沙田区的坐标点可能存在错误，导致两个区域在地图上重合
- 整体精度：目前使用的简化坐标可能过于简化，无法准确表示香港复杂的地理形状

## 2. 解决方案：

我建议获取更高精度的香港地图GeoJSON数据。我可以帮你修改当前的 `hongkong.json` 文件，增加更多坐标点来填补区域间的空隙。下面是几个关键区域需要优化的部分：

```
{ } hongkong.json
```

```
▷ Apply to hongkong.json...
```

```
// 以元朗区为例，增加更多坐标点来减少与周边区域的空隙
```

很多细节需要精细的调整，有时候还会导致新的错误发生。

# CASE：疫情实时监控

地区之间有很多空隙，比如元朗区和荃湾区，另外大埔区和沙田区有重叠的区域了。帮我查看地图的数据，中间很多没有相连，空隙太大了。

我注意到了香港地图显示的问题，地区之间确实存在较大空隙，且有些区域如大埔区和沙田区出现了重叠。这是因为当前使用的GeoJSON数据精度不够或坐标点可能存在问题。

让我分析一下香港地图数据并提供解决方案：

## 1. 现有GeoJSON数据问题分析：

- 空隙问题：当前数据中各区域的边界点可能不够密集，导致渲染时出现空隙
- 重叠问题：大埔区和沙田区的坐标点可能存在错误，导致两个区域在地图上重合
- 整体精度：目前使用的简化坐标可能过于简化，无法准确表示香港复杂的地理形状

## 2. 解决方案：

我建议获取更高精度的香港地图GeoJSON数据。我可以帮你修改当前的 `hongkong.json` 文件，增加更多坐标点来填补区域间的空隙。下面是几个关键区域需要优化的部分：

```
{ } hongkong.json
```

```
▷ Apply to hongkong.json...
```

```
// 以元朗区为例，增加更多坐标点来减少与周边区域的空间
```

很多细节需要精细的调整，有时候还会导致新的错误发生。

# 打卡：疫情实时监控大屏



使用data.xlsx 搭建疫情实时监控大屏，可以使用Flask + Echarts

1. 核心指标展示（顶部卡片）
2. 地理分布图（中心区域左侧）
3. 每日新增与累计确诊数据（中心区域右侧上部）
4. 各区域疫情对比（右侧中部）
5. 趋势分析图（底部）
6. 风险预警看板（右下角）



# Trae使用

# Trae使用

Trae编辑器 (The Real AI Engineer) :

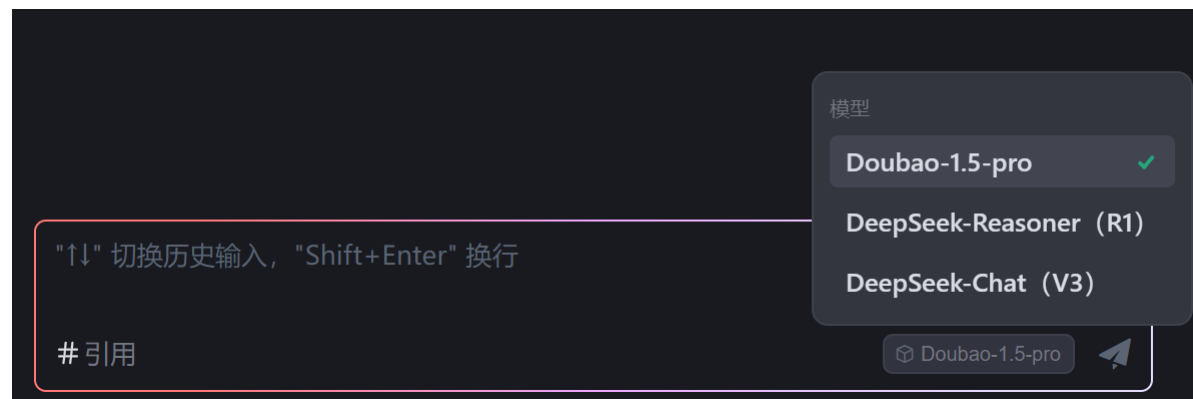
<https://www.trae.com.cn/>




字节推出的一款 AI 驱动的开发环境，特点：

**Builder 模式：**Trae的亮点，它会自主拆解需求并自动完成多轮编码任务，从想法描述到功能实现一气呵成。

**多模态支持：**开发者可以上传图像，Trae 能够理解图像内容并生成相关代码

**免费使用：**目前 Trae 提供免费访问，包括内置的 Doubao-1.5-pro, DeepSeek-R1, DeepSeek-V3





Thank You  
Using data to solve problems