第四次作业-基于 BKVision 图表平台实现用户画像与行为分析

# 实验目的

1. 巩固SaaS应用的软件开发&设计能力
2. 了解蓝鲸BKVision图表平台的产品功能与使用方法
3. 掌握基本Django中间件的开发技能与数据采集
4. 掌握蓝鲸图表平台的嵌入方式与SDK使用
5. 提升SaaS开发技能，巩固基础数据分析能力与数据采集技能
6. 提升SaaS开发技能，进一步熟悉开发框架与后台建模

# 实验环境

1. 硬件环境需求： PC或笔记本， 支持外网访问
2. 软件环境需求

系统： Windows, MacOS, Linux

安装 Python 3.6.12

安装 MySQL 8.3

安装 Git (最新版本即可)

安装 pre-commit代码检查工具（可选）

安装 VSCode，PyCharm 或其它 IDE

# 实验内容

在此前两期SaaS开发作业的基础上，借助蓝鲸BKVision图表平台实现用户行为可视化分析与前端嵌入，通过设计并开发Django中间件，实现用户行为数据埋点采集并存储至数据库，通过BKVision实现仪表盘嵌入。

# 实验评分标准

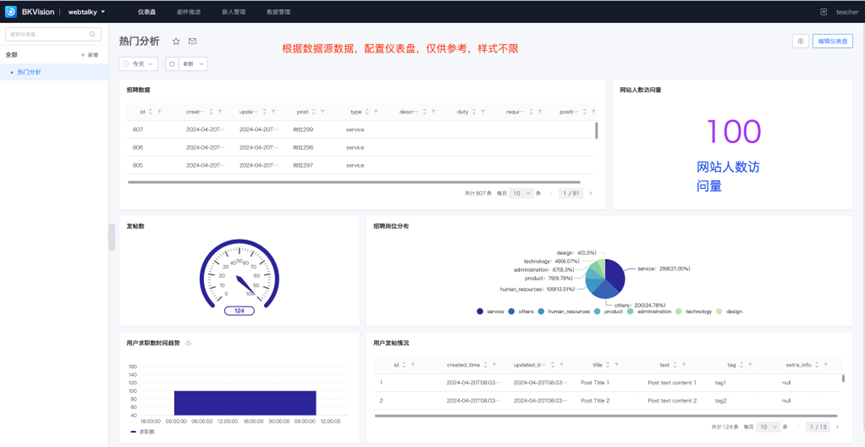
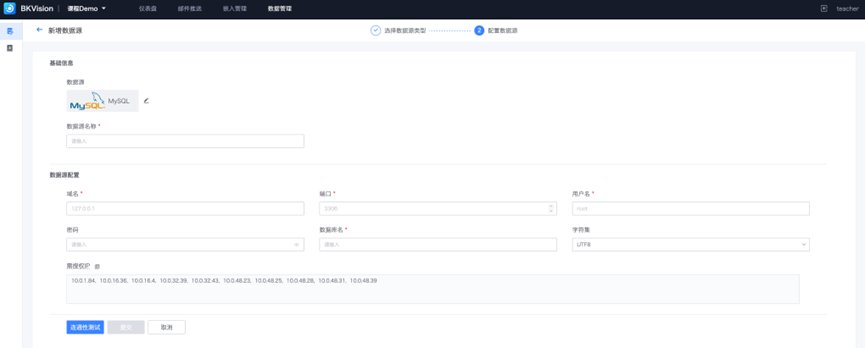
整体要求：请同学们采用迭代方式进行需求分析、面向对象设计和编程实现，实训课报告中需包含相应的需求规约、设计规约、接口文档，项目开发说明

考点一： 在此前两期课程SaaS作业的基础上，通过Django中间件实现用户行为采集并存储到SaaS数据库，比如：登录行为、查询业务列表行为、执行作业行为等

相关资料：

1. Django中间件开发：[Django中间件开发](https://docs.djangoproject.com/zh-hans/4.1/topics/http/middleware/)
2. Django中间件原理及示例：[Django中间件原理及示例](https://pythondjango.cn/django/advanced/9-middleware/)

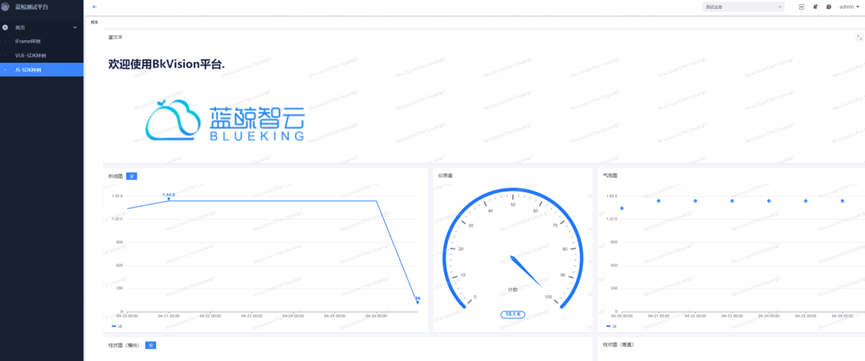
考点二：在BKVision图表平台创建空间，接入对应的SaaS数据库，并对采集的数据进行仪表盘配置（仪表盘样式不限，鼓励大家自由发挥），并发布仪表盘，参考图如下：



相关资料：

1. 蓝鲸图表平台：[蓝鲸BKVision图表平台](https://apps.ce.bktencent.com/bk-vision/#/space)

考点三：设计通过iFrame或BKVision-SDK方式，实现仪表盘发布并嵌入到对应的SaaS前端界面中，参考图如下：



其他评分项：

1.Python代码符合[PEP8规范](https://www.python.org/dev/peps/pep-0008/)，可酌情加分

2.系统边界考虑完善，系统性能优良，可酌情加分

3.Django中间件实现出色，采集覆盖大部分接口场景，可酌情加分

4.Django中间件在实现数据存储时，能够通过Celery异步任务实现，可酌情加分

5.前端界面优美，用户交互体验良好，可酌情加分

6.后端代码能够实现单元测试以及日志、异常处理等，可酌情加分

# 实验过程与结果

* + - 1. **设计实现用户行为数据存储Model**

1. **在home\_application/models.py中实现API请求次数记录表数据Model ApiRequestCount如下：**

**文本

描述已自动生成**

1. **执行数据库迁移**
   * + 1. **设计自定义中间件，实现用户行为埋点记录**
2. **编写core/middleware.py如下：**

**文本

描述已自动生成**

1. **添加自定义中间件到config/default.py中：**



* + - 1. **在BKVision平台添加并配置数据源**

**图形用户界面, 应用程序

描述已自动生成**

* + - 1. **创建并配置仪表盘**

1. **配置实现API调用类别柱状统计图**

**图形用户界面, 应用程序

描述已自动生成**

1. **配置实现API总调用次数仪表盘**

**图形用户界面, 应用程序

描述已自动生成**

1. **配置实现用户备份文件次数统计饼图**

**图形用户界面, 应用程序

描述已自动生成**

1. **配置实现API调用次数明细图**

图形用户界面, 应用程序

描述已自动生成

* + - 1. **将仪表盘嵌入到自己的前端应用**

1. **新增iFrame嵌入**

**图形用户界面, 应用程序, Teams

描述已自动生成**

**图形用户界面, 应用程序, 网站

描述已自动生成**

1. **将iframe代码块添加到前端的src/views/DashBoard/index.vue中，如下：**

**文本

描述已自动生成**

* + - 1. **部署上线**

**最终效果如下：**

**图形用户界面, 应用程序, Teams

描述已自动生成**

# 实验心得与体会

通过这次基于BKVision图表平台实现用户画像与行为分析的实验，我深入了解了数据可视化的重要性和实现方法。开发Django中间件进行用户行为数据采集，不仅提升了我的后端开发技能，也让我理解了数据埋点的原理和价值。在BKVision平台上创建和配置仪表盘的过程，让我掌握了数据可视化的基本技巧。将仪表盘嵌入SaaS应用的实践，使我对前后端集成有了更深入的认识，同时也体会到了数据驱动决策的重要性。