基于配置平台实现游戏业务主机管理

目录

实战开发需求

●**课题目标：**

1、巩固SaaS应用的软件开发&设计能力

2、了解蓝鲸CMDB配置平台的功能、数据结构与使用方法

3、掌握蓝鲸网关/ESB组件API的调用方法与鉴权模式

4、能够通过蓝鲸API联通CMDB平台获取业务主机与架构信息

5、提升SaaS开发技能，能够进行前后端联调并设计接口

6、提升SaaS开发技能，进一步熟悉开发框架与后台建模

●**课题内容：**

基于蓝鲸SaaS开发框架开发一个独立SaaS应用，借助蓝鲸CMDB配置平台实现游戏业务主机资源拉取与查询，通过蓝鲸网关/ESB组件API联通 CMDB平台实现数据获取，并根据CMDB主机数据结构，设计查询条件与对应接口

●**样例展示：**https://apps.ce.bktencent.com/stag--frontend--bk-cmdb-demo1/

需求分析-功能点

- **功能一：**创建SaaS应用，通过蓝鲸ESB组件API调用蓝鲸配置平台CMDB接口，拉取业务->集群->模块列表

- **功能二**：根据业务->集群->模块结构，前端实现级联选框，实现级联联动

- **功能三：**根据主机详情字段，设计查询条件，实现主机列表查询接口

- **功能四：**根据点击的主机ID/名称，实现主机详情查询接口

- **功能五：**设计前端界面，运用各个组件实现数据渲染与用户交互

目录

前言：前端开发环境配置

前端开发—前端环境安装及本地运行

- 开发基础环境准备：
  - nodejs安装： nvm， node.js v20.10.0

- 创建前端模块， 下载前端代码

  本次课程使用环境：https://ce.bktencent.com/

  前端代码在群里下载

- 前端模块依赖安装及运行：
  - 安装：npm install
  - 运行：npm run dev

NVM是Node Version Manager
也就是Node版本管理工具，用于管理和
切换到不同版本的Node.js

后端配置更改

- 数据库配置：
  - 在local_settings文件中进行数据库配置
  - 在gitignore文件中把local_settings文件添加进去

思考：为什么不像第一次那样直接在dev.py中配置数据库

实战步骤

实战步骤

- **功能点：实现主机层级信息的获取**
  - 步骤一：实现业务列表拉取接口
  - 步骤二：实现集群列表拉取接口
  - 步骤三：实现模块列表拉取接口
  - 步骤四：对获取到的层级数据进行联动展示

- **功能点：获取主机详情字段**
  - 步骤五：实现主机列表查询接口
  - 步骤六：对获取到的主机列表数据进行展示

- **功能点：获取主机详细信息**
  - 步骤七：实现主机详情查询接口
  - 步骤八：对获取到的主机详细信息在侧边栏进行展示

- **功能点：部署展示**
  - 步骤九：部署运行项目

步骤一：实现业务列表拉取接口

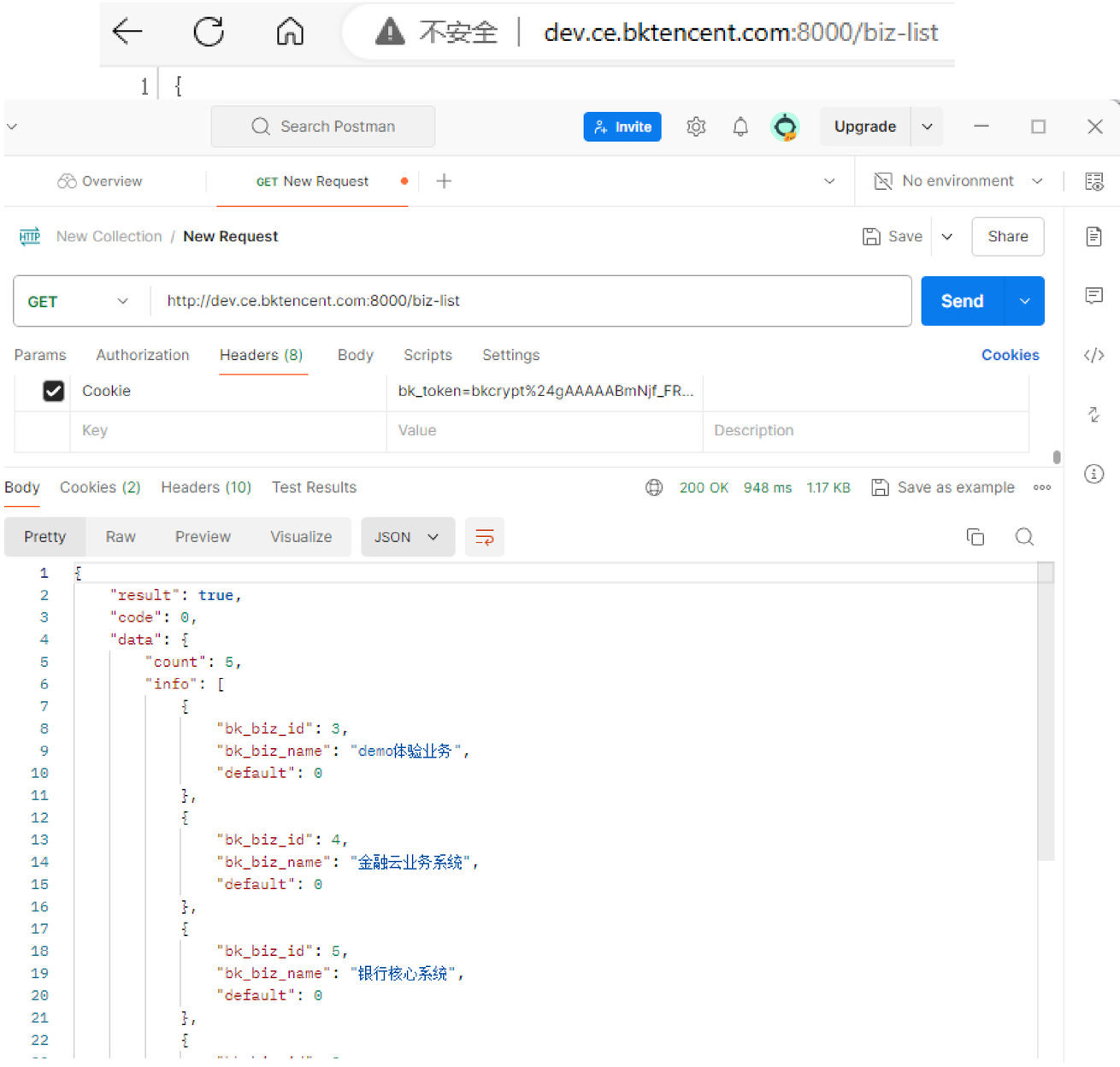
- API组件的访问方式
  - 导入对应的依赖包：
    from blueking.component.shortcuts import get_client_by_request
  - 获取对应的client
    client = get_client_by_request(request)
  - 设置对应API的请求参数
    kwargs = {'bk_biz_id': 1} # 请求参数
  - 调用对应的API
    result = client.cc.get_app_host_list(kwargs)

- 实现业务列表拉取接口
  - 在home_application.views下编写接口，接口代码详见实验手册
  - 接口的文档如下：
    https://apigw.ce.bktencent.com/docs/component-api/default/CC/search_business/doc

步骤一：实现业务列表拉取接口

- 验证接口是否编写成功
  - 启动Django服务，访问地址http://dev.ce.bktencent.com:8000/biz-list，即可看到接口响应数据
  - 也可以使用Postman测试接口，注意在headers里面添加包含bk_token的Cookie
- 前后端联调
  - 前端需要有对应的访问接口的函数

```
},
// 查询业务列表
getBizData(context, params, config：{}  = {}) {
  // eslint-disable-next-line no-undef
  return http.get(BACKEND_API_PREFIX+`/biz-list`, params, config);
},
// 根据业务ID，查询集群列表
```



步骤一：实现业务列表拉取接口

思考：功能已经实现，但是到这里就结束了吗？

- 最佳实践：对代码进行错误处理，提升代码的鲁棒性
  - 在代码中使用try-catch进行异常捕获
  - 蓝鲸API能够为异常的请求返回对应的错误信息，这样就能够很好的利用这些信息来进行错误处理


```
{
    "result": false,
    "code": 1306000,
    "data": null,
    "message": "business id [bk_biz_id] This field is required.",
    "request_id": "1b6b5c0e7c1844cebeddfda87d48e323"
}
```

跨域问题

- 前后端联调跨域问题：
  - 前端在访问对应接口时出现CORS Error
  - 解决方案：在后端添加对应的跨域相关的配置信息

思考：跨域是为了资源的安全性的，在实践中应该如何安全的配置跨域信息？

跨域问题

- 普通实现：允许所有的来源都可以对资源进行访问
  - CORS_ALLOW_ALL_ORIGINS = True

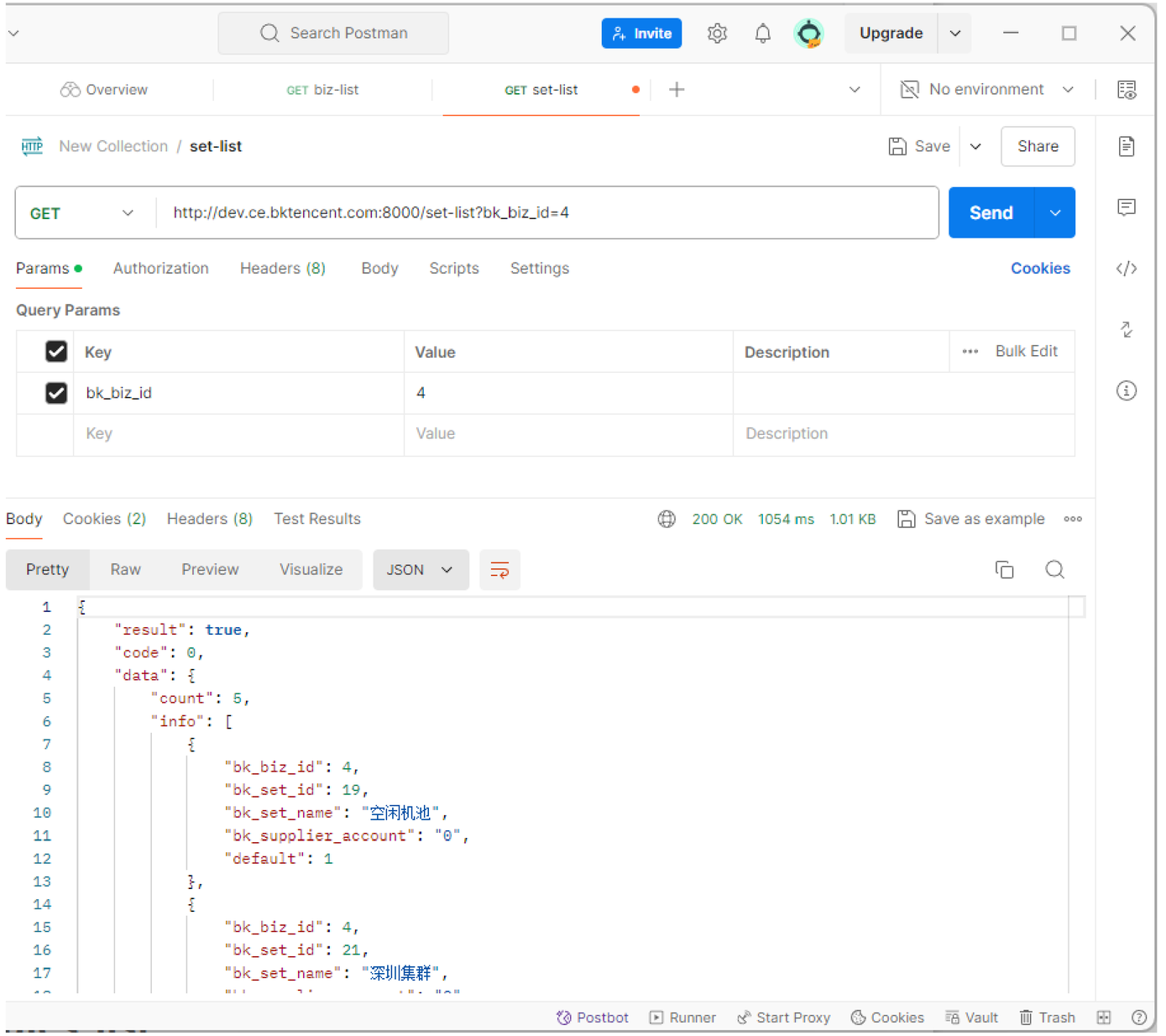
- 最佳实践：只允许我们需要的使用的应用来对后端的资源进行访问

```
CORS_ALLOWED_ORIGINS = [
    os.getenv('CORS_ALLOWED_ORIGIN'),    # 允许跨域的域名
]
```

蓝鲸团队编程最佳实践总结：
https://bk.tencent.com/docs/markdown/ZH/DevelopGuide/DevSpecification/BackendDevStandards/BestPractices/python/README.md

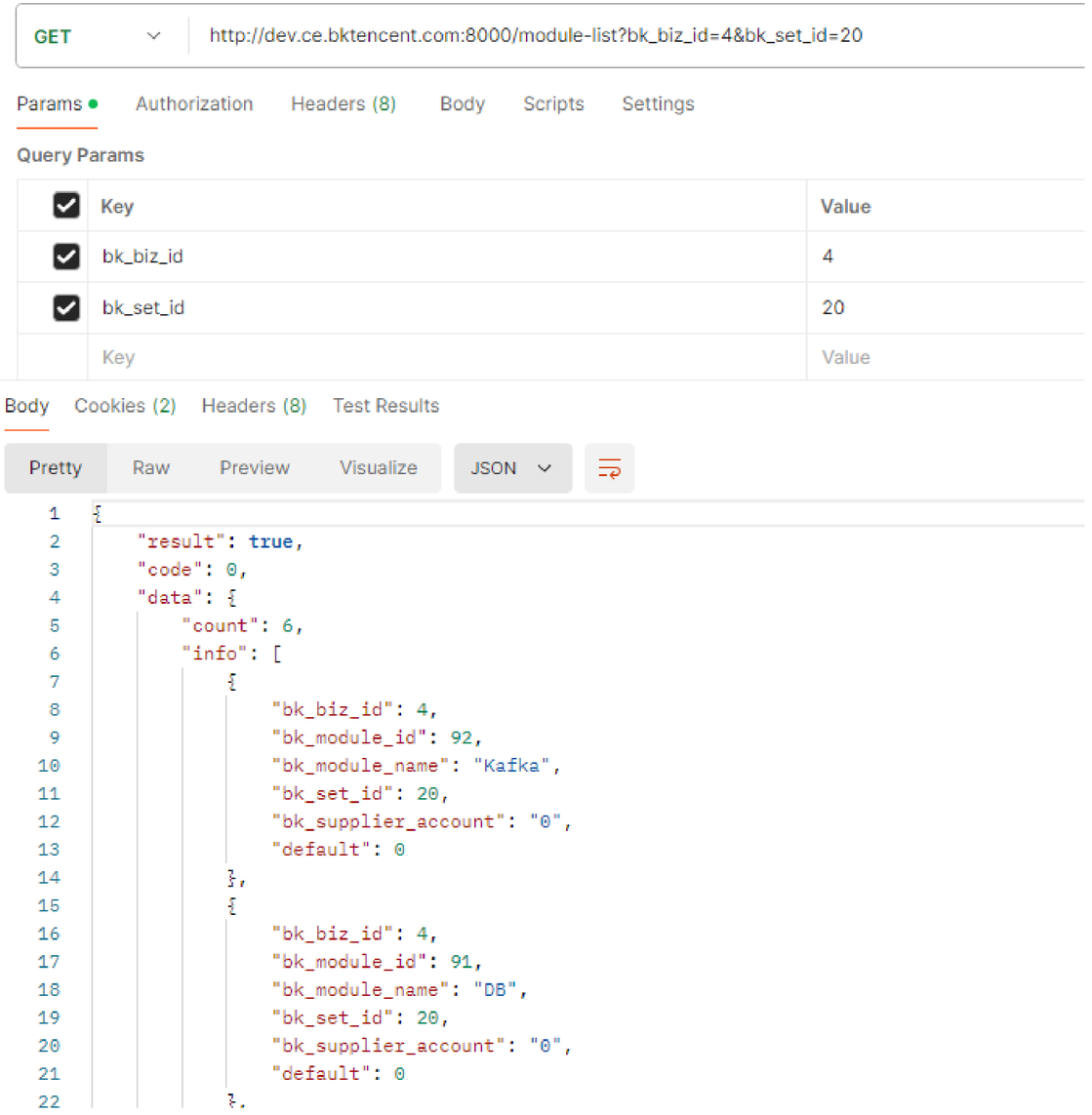
步骤二：实现集群列表拉取接口

- 实现集群列表拉取接口
  - 使用search_set接口来进行实现，具体实现代码详见实验手册
  - 接口文档为：https://apigw.ce.bktencent.com/docs/component-api/default/CC/search_set/doc
  - 添加对应的路由配置
  - 对实现的接口进行测试



步骤三：实现模块列表拉取接口

- 实现集群列表拉取接口
  - 使用search_module接口来进行实现，具体实现代码详见实验手册
  - 接口文档为：https://apigw.ce.bktencent.com/docs/component-api/default/CC/search_module/doc
  - 添加对应的路由配置
  - 对实现的接口进行测试



步骤四：前端实现级联选框

- 前端开发框架目录结构介绍
  - src/views存放vue页面组件文件
  - 在router/index.js下编写路由，新增页面后需要在此编写
  - 在store/modules/example.js下编写js函数，实现与后端接口联通

- 前端实现

  - 采用蓝鲸前端组件进行前端的搭建，查询表单使用bk-form组件，下拉选框使用bk-select组件

  - 蓝鲸MagicBox组件文档：组件文档

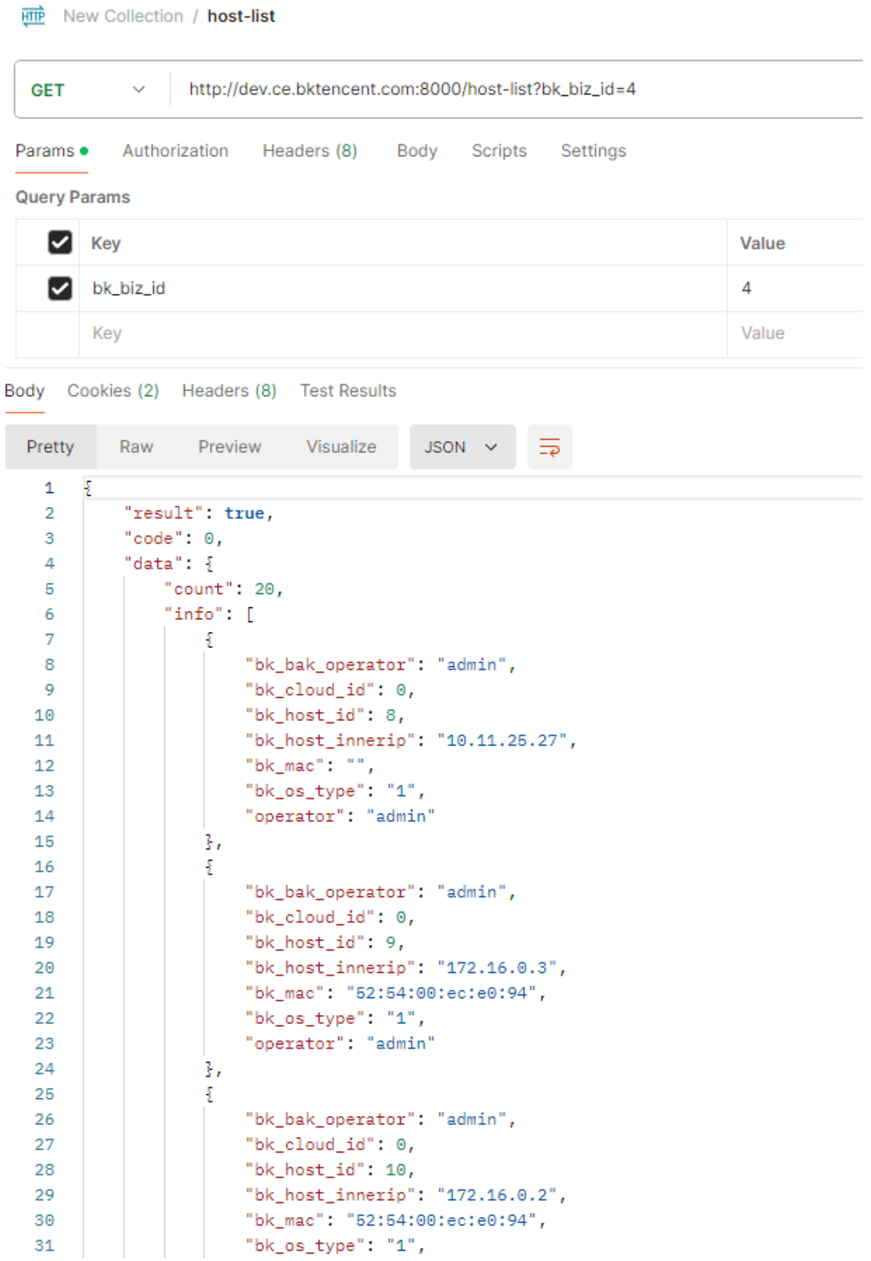
  - 详细实现见前端代码，前端代码已经提供给大家了

- 后端接口联通

  - 在store/modules/examples.js中添加js函数，实现与后端API的联通

步骤五：实现主机查询接口

- 实现主机查询接口
  - 主机查询接口类似于此前的拉取集群和模块的接口，同样需要传递参数，这里使用GET 法，从 request.GET 中获取查询参数，不同的地 在于，这里需要考虑到可选参数，因为主机查询接口的查询参数不固定，只有业务ID为必选参数，其余的如集群ID、模块ID、主机ID、维护 等均为额外可选参数
  - 使用list_biz_hosts接口实现主机查询功能
  - 接口文档为：https://apigw.ce.bktencent.com/docs/component-api/default/CC/list_biz_hosts/doc
  - 添加对应的路由配置
  - 对实现的接口进行测试

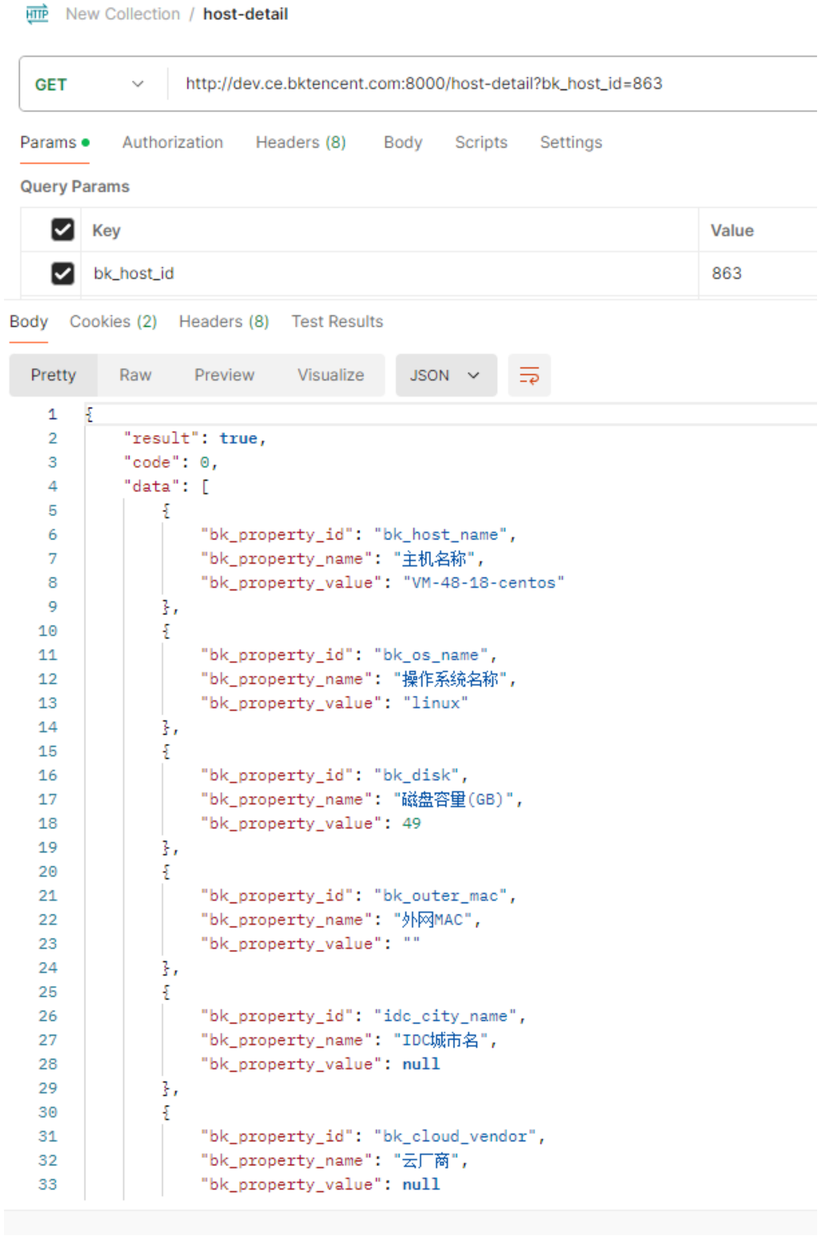


步骤六：实现主机列表数据渲染至前端

- 将数据渲染至前端
  - 添加额外的表单条目
  - 监听数据变化
  - 添加前端与后端的联通接口
  - 进行前端的查询按钮的绑定事件
  - 使用bk-table组件将数据渲染至列表
  - 组件文档：bk-table组件文档
  - 具体实现见发送给大家的前端代码

步骤七：实现主机详情查询接口

- 实现主机详情查询接口
  - 使用get_host_base_info接口来进行实现，具体实现代码详见实验手册
  - 接口文档为：https://apigw.ce.bktencent.com/docs/component-api/default/CC/get_host_base_info/doc
  - 添加对应的路由配置
  - 对实现的接口进行测试

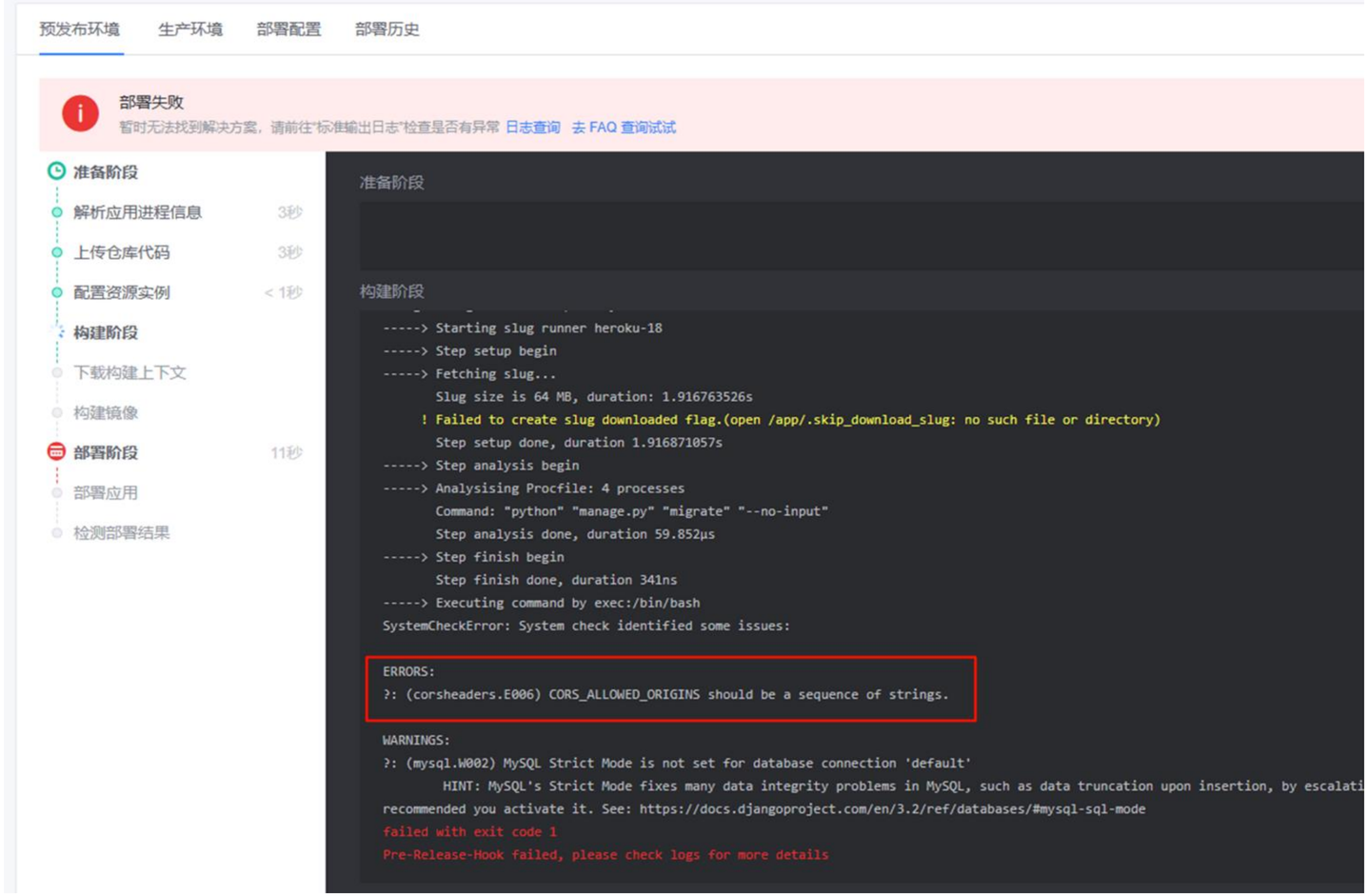


步骤八：实现主机详情渲染至侧边栏

- 将数据渲染至侧边栏
  - 使用bk-sideslider组件来渲染主机详情信息
  - 组件文档：bk-sideslider组件
  - 添加data字段
  - 实现主机详情查询函数

步骤九：部署上线

- 部署后端模块
  - 建议先部署后端模块再部署前端模块，因为本次实验需要通过将后端的上线地址作为环境变量来对前端的环境进行配置
  - **注意在后端部署的时候添加对应跨域的环境变量**



思考：这里部署失败是什么原因?

步骤九：部署上线

- 部署前端模块
  - 依据部署上线的后端模块获取线上后端的URL
  - 修改环境变量BK_BACKEND_API_PREFIX为后端的URL
  - 切换模块为前端模块，进行前端代码的部署
  - 将后端的URL配置成环境变量能够增加代码的鲁棒性，即使环境进行变化，也只需要简单的配置就可以让代码重新运行，而不是修改原始的代码

可选优化项目

- 后端目录结构优化-新建APP，编写L2阶段CMDB相关的接口
- 后端框架改造-使用Django Restframework，重构已有接口，采用CBV类视图实现
- 后端框架改造-路由采用Router进行实现
- **后端代码优化-在接口中添加异常处理以及日志记录，增强代码鲁棒性**
- 后端代码优化-使用pre-commit，优化代码规范
- 后端代码优化-基于pytest等测试框架，为接口编写单元测试
- **后端功能改造-设计数据模型，实现CMDB数据本地存储**
- **后端功能改造-实现数据备份功能，备份至本地数据库**
- 前端界面优化-添加拓扑树，优化用户交互

目录

实验评分标准

整体要求	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，项目开发说明
考点一	创建SaaS应用，通过蓝鲸 ESB组件API联通CMDB配置平台，实现业务、集群、模块级联拉取接口，并在前端进行下拉框组件展示与数据渲染
考点二	添加 根据蓝鲸CMDB配置平台的主机数据结构设计查询条件（包括但不限于主机名称、主机维护人、主机备份人等字段），实现主机查询接口（模糊查询可加分）
考点三	设计前端界面（可参考课程前端样例代码与MagicBox组件库），进行前后端联调，实现前端主机列表数据渲染
考点四	实现主机详情展示接口&界面，要求点击主机后能够查看主机详情信息并通过前端界面进行数据展示
考点五	将实现的后端&前端代码上传至Git代码托管平台，并部署到PaaS平台，数据交互展示无误，不存在CORS、CSRF等问题
其他评分项	1.Python代码符合PEP8规范，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.设计对应数据Model，将CMDB配置信息存储到SaaS应用数据库中，可酌情加分
	4.前端界面优美，用户交互体验良好，可酌情加分
	5.实现数据同步功能，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

蓝鲸智云

官方微信公众号

蓝鲸高校培训群