足球运动员数据分析平台

1. 环境:Linux下anaconda新建虚拟环境
2. 技术:Vue,Vue-CLI(版本:4.x)(Vue脚手架),python3,ElementUI,

(脚手架:命令行接口工具)

1. 前端使用代理服务器解决跨域问题
2. 后端使用CORS解决跨域问题
3. 后端采用RESTful(资源状态转换,面向资源开发的接口规范)架构编写URL,统一资源接口(使用标准的HTTP方法对资源进行操作[避免使用动词])(使用缓存),不要使用动词,用HTTP方法去代表动作

(API接口:应用程序编程接口;接口规范:restful,RPC)

1. 后端采用django-rest-framework(DRF)(Web应用开发框架)(重点)，利用序列化器,模型序列化器和相应的视图类(如:APIView,GenericAPIView,ModelViewSet对视图类进行编写),其中运用了大量python的面向对象,继承和封装的知识点
2. 采用了DRF中的路由Routers快速实现路由信息
3. 配置并使用了DRF中的认证组件Authentication和权限组件 Permissions等功能组件
4. django采用类视图函数(CBV[面对对象,self是谁,调用顺序(由下到上的查找)])

(post发请求的数据格式urlencoded(eg:\a\a\a\a=2&b=3)/json(eg:\a\a\a\{“a”:1,”b”:2})....)根据Content-Type指明的请求数据类型  
8.djangorestframework-simplejwt(令牌:用于认证用户身份信息,保持登录状态的作用)(由于原来对token解码后数据太少,所以自定义了令牌,以包含更多的数据)

9.使用了echarts的仪表盘，饼图,进行数据分析后的结果展示

10.利用element ui中的表格,卡片,图片展示,布局进行结果的展示

11.(1)数据分析,(2)足球人才仓库,(3)球员间的数据分析与对比

12.数据仓库:对数据进行存储与排序，利用数组的filter方法对从后端返回的球员数据进行过滤，以实现搜索，完成查询功能

13.条件查询:filter--QuerySet(类数组,集合)容器对象,get--模型对象

14.在对比页时，各项数据分别对比，较大的数据旁使用了Vue中插槽slot(内容分发)(让父组件可以向子组件指定位置插入html结构)会显示相应的差值

15.缓存(把高频读取的数据,放置到更快的存储介质里),使用django-redis缓存数据,利用@cache\_page装饰器完成缓存操作（h还可以有模板缓存）

16.分为访客和工作人员

17.采用django的admin后台作为足球平台的后台

18.celery(类似于生产者和消费者模型)处理算法问题

19.使用MongoDB来存储用户上传的文件(why:易于使用,文档存储;便于获取数据;性能优越,查询快;全索引支持)

20.序列化:使用序列化器,模型序列化器对模型类对象转换为字典,在利用Rsponse对象变为json字符串,通过对生成的序列化器对象的data属性可以获取序列化后的数据

21.反序列化:对数据进行验证后，才能获取验证成功的数据或保存成模型类对象,这个过程可以把数据转成模型类对象,可以通过实现create()和update()两个方法来实现,在反序列化数据的时候，就可以通过save()方法返回一个数据对象实例了,

22.使用Django\_notifications实现网站消息通知

23.对请求和响应的回调函数采用异步编程,使用了async/await关键字简化了异步操作