

شرح مسئله:

یک سیستم مدیریت کتابخانه طراحی کنید که به کاربران اجازه دهد از طریق API، کتابها، اعضا، و وامهای کتابخانه را مدیریت کنند. هدف از این تمرین آشنایی با طراحی API های RESTful، کار با مدلها و سریالایزرها، و پیاده سازی منطق تجاری با استفاده از DRF است.

ویژگی های مورد نیاز:

۱. مدل ها: (Models)

۳ مدل اصلی زیر را ایجاد کنید:

۱. Book (کتاب):

- title (عنوان): CharField
- author (نویسنده): CharField
- published_date (تاریخ انتشار): DateField
- stock (تعداد موجودی): IntegerField

۲. Member (عضو):

- name (نام): CharField
- email (ایمیل): EmailField
- phone_number (شماره تماس): CharField

۳. Loan (وام):

- book (کتاب): ForeignKey به مدل Book
- member (عضو): ForeignKey به مدل Member
- loan_date (تاریخ وام): DateField
- return_date (تاریخ بازگشت): DateField (می تواند خالی باشد)

۲. API های مورد نیاز:

۱. کتاب ها (Books):

- GET /books/ → نمایش لیست کتابها.
- POST /books/ → اضافه کردن کتاب جدید.
- PUT /books/{id}/ → ویرایش اطلاعات کتاب.
- DELETE /books/{id}/ → حذف کتاب.

۲. اعضا (Members):

- GET /members/ → نمایش لیست اعضا.
- POST /members/ → ثبت نام عضو جدید.
- PUT /members/{id}/ → ویرایش اطلاعات عضو.
- DELETE /members/{id}/ → حذف عضو.

۳. وام‌ها (Loans):

- GET /loans/ → نمایش لیست وام‌ها.
- POST /loans/ → ثبت وام جدید.
- PUT /loans/{id}/ → ویرایش اطلاعات وام (مثلاً ثبت تاریخ بازگشت).
- DELETE /loans/{id}/ → حذف وام.

۳. سریالایزرها: (Serializers)

برای هر مدل، سریالایزرهای زیر را طراحی کنید:

۱. BookSerializer:

- شامل تمام فیلدهای مدل باشد.

۲. MemberSerializer:

- شامل تمام فیلدهای مدل باشد.

۳. LoanSerializer:

- هنگام ثبت وام: (POST)

▪ بررسی کنید که اگر stock کتاب برابر صفر باشد، خطا برگردد.

▪ پس از ثبت موفق وام، تعداد موجودی کتاب (stock) کاهش یابد.

- هنگام ویرایش وام: (PUT)

▪ اگر تاریخ بازگشت (return_date) ثبت شد، موجودی کتاب به صورت خودکار افزایش یابد.

- اعتبارسنجی‌ها:

▪ تاریخ بازگشت نباید قبل از تاریخ وام باشد.

- نمایش اطلاعات اضافی: (read-only)

▪ به جای نمایش شناسه کتاب و عضو، نام کتاب (book_title) و نام عضو (member_name) نمایش داده شود.

۴. احراز هویت: (Authentication)

- کاربران برای افزودن، ویرایش یا حذف منابع باید احراز هویت شوند.
 - از **Token Authentication** استفاده کنید.
-

۵. اعتبارسنجی: (Validation)

۱. بررسی کنید:
 - موجودی کتاب نباید منفی شود.
 - تاریخ بازگشت (return_date) قبل از تاریخ وام (loan_date) نباشد.
 ۲. هنگام ثبت وام، تنها کتاب‌هایی که موجودی دارند، قابل انتخاب باشند.
-

۶. تست‌ها: (Tests)

تست‌های زیر را با استفاده از ابزار **pytest-django** یا تست داخلی **Django** بنویسید:

۱. ثبت کتاب:
 - بررسی کنید که کتاب با اطلاعات صحیح به درستی ثبت شود.
 ۲. ثبت وام:
 - بررسی کنید که اگر موجودی کتاب صفر باشد، وام ثبت نشود.
 - بررسی کاهش موجودی کتاب پس از ثبت وام.
-

هدف تمرین:

- طراحی API های RESTful با استفاده از DRF.
- استفاده از سریالایزرها برای مدیریت داده‌ها.
- کار با احراز هویت و مدیریت دسترسی.
- پیاده‌سازی اعتبارسنجی و منطق تجاری.
- آشنایی با فیلدهای read-only و سریالایزهای سفارشی.