



## MioBook

### مقدمه

در این فاز از پروژه، منطق برنامه را به فریم‌ورک Spring منتقل می‌کنیم تا سرور بتواند به درخواست‌های HTTP پاسخ دهد. برای این منظور، شما بر اساس تصمیمات خود، مجموعه‌ای از API-های RESTful را برای سیستم تعریف می‌کنید تا تمامی عملیاتی که در فاز قبلی از طریق خط فرمان انجام می‌شد، اکنون از طریق درخواست‌های HTTP قابل اجرا باشد.

علاوه بر این، در این فاز برخی تغییرات در منطق برنامه اعمال شده است که سیستم باید با آنها سازگار شود. در نهایت، انتظار می‌رود که یک بک‌اند با معماری تمیز و قابلیت توسعه‌پذیری بالا ارائه دهید.

### بخش اول: تغییرات منطق برنامه

#### بازخورد کاربر به کتاب

- فقط کاربرانی می‌توانند برای یک کتاب بازخورد ثبت کنند که آن را خریده باشند.
- در صورت قرض کردن کتاب، نظر کاربر فقط در زمان وجود آن کتاب در دارایی‌هایش قابل ثبت است و در صورت گذشتن مهلت قرض، دیگر قابلیت ثبت بازخورد را ندارد.
- کاربر در صورتی که از قبل برای یک کتاب بازخورد ثبت کرده باشد، می‌تواند بازخورد جدیدی ثبت کند و در این صورت، بازخورد جدید جایگزین بازخورد قبلی می‌شود (یعنی بازخورد قبلی حذف شده و یک بازخورد با تاریخ جدید اضافه می‌شود).
- میانگین امتیاز کتابی که هیچ بازخوردی ندارد، صفر در نظر گرفته می‌شود.

#### جست‌وجو کتاب‌ها

در فاز قبلی، کاربر امکان جست‌وجو بر اساس نام کتاب، نام نویسنده، ژانر کتاب و سال انتشار کتاب را داشت که برای نام کتاب و نویسنده، صرفاً نیاز بود که عبارت جست‌وجو شامل نام کامل باشد. در این فاز، قابلیت ترکیب این جست‌وجوها را نیز خواهیم داشت و به طور مثال، کاربر می‌تواند لیست کتاب‌ها را بر اساس نام نویسنده‌ای خاص در سال 2014 جست‌وجو کند. همچنین، کتاب‌های بازگردانده شده باید قابلیت مرتب‌سازی بر حسب میانگین امتیاز و یا تعداد بازخوردها، به صورت نزولی و یا صعودی را داشته باشند. به این منظور خوب است که از Query Parameters استفاده کنید.

## صفحه‌بندی نتایج

تا اینجا، دو قابلیت مربوط به لیست کردن کتاب‌ها (به همراه فیلتر و جست‌وجوی آنها) و دریافت بازخوردهای یک کتاب، لیست تمامی آنها را برمی‌گردانند. از آنجا که حجم این دو مورد می‌تواند بسیار زیاد باشد، دریافت تمامی آنها لود زیادی را به سرور وارد می‌کند. برای همین سیستم‌ها از صفحه‌بندی یا Pagination استفاده می‌کنند. شما نیز باید دریافت کتاب‌ها و بازخوردهایشان را به صورت صفحه‌ای پیاده‌سازی کنید تا با دریافت پارامتر شماره صفحه، لیستی با تعدادی ثابت (یا یک پارامتر محدود شده) از کتاب یا بازخورد برگردانده شود.

## بخش دوم: امکانات جدید سیستم

### قابلیت لاگین کاربر

کاربر باید بتواند با وارد کردن نام کاربری و رمز عبور خود، وارد سیستم شده و در صورت لزوم از آن خارج شود. در این فاز، سیستم را تک کاربر فرض می‌کنیم و برای راحتی صرفاً آخرین کاربری که وارد سیستم شده را در کد خود نگه می‌داریم. جهت تغییر کاربر، کاربر کنونی باید لاگ‌اوت کرده و کاربر جدید لاگین کند. برخی دستورات نیازمند کاربر admin و برخی نیازمند customer بودند و این بررسی را انجام داده‌اید. در اینجا باید همچنین حالت لاگین نبودن کاربر را بررسی کرده و فقط برخی دستورات بدون لاگین هم قابل دسترسی باشند.

### دریافت داده از سرویس خارجی

برای داشتن کمی داده‌های اولیه پس از اجرای برنامه، جهت راحتی کار (و هر بار اضافه نکردن آنها پس از هر اجرا)، یک سری داده را در ابتدای اجرای برنامه وارد لیست‌های خود می‌کنیم. این داده‌های کاربران، نویسندگان، کتاب‌ها و بازخوردها در یک سرویس خارجی برای شما قرار داده شده‌اند. این سرویس دارای چهار endpoint بوده که هر کدام یکی از داده‌ها را به فرمت JSON به شما می‌دهند. شما باید با انجام یک ریکوئست به این [لینک](#) در داخل کد برنامه، داده‌ها را گرفته و وارد سیستم خود کنید. همچنین ارورهای احتمالی کانکشن را بررسی کنید و در صورت خطا، صرفاً لاگ کرده و برنامه بدون داده‌های اولیه باید اجرا بشود. برای ارسال ریکوئست، ابزارهای کتابخانه استاندارد جاوا کافی می‌باشند.



انتقال از CL به

Http Req

## بخش سوم: مهاجرت به فریم‌ورک Spring

### استانداردسازی API

در ادامه تمرین لازم است که API سیستم خود را طراحی کنید. از منابع گفته شده در کلاس و سرچ‌های خود برای دیدن استایل‌های مختلف API-ها استفاده کنید و در نهایت، بر روی URL-های API خود و همچنین متد HTTP درخواست‌ها به نتیجه رسیده و آنها را استانداردسازی کنید.

در قدم بعدی، باید روی فرمت JSON ریکوئست و ریسپانس‌ها نیز به نتیجه برسید. به عنوان یک پیشنهاد، خوب است که برای همه ریسپانس‌ها، فیلدهایی از جمله status و timestamp و message مشترک باشد و در صورت نیاز در فیلد data داده مدنظر آن endpoint قرار بگیرد. برای این کار باید یک کلاس wrapper برای Response بسازید که همه متدهای کنترلر شما آن را ریترن می‌کنند.

### آشنایی با CORS

وقتی که در یک صفحه وب هستیم، در صورتی که آن سایت در مرورگر بخواهد به یک سایت دیگر ریکوئست بزند، سایت مقصد باید اجازه دسترسی به منابعش را به ما داده باشد. در غیر این صورت، مرورگر برای دلایل امنیتی، از همه ریکوئست‌ها به دامنه‌هایی غیر از خود سایت مبدأ جلوگیری می‌کند. به این مکانیزم CORS گفته می‌شود که از طریق چند پارامتر در هدر HTTP مشخص می‌شود. در ابتدا با آن آشنا شوید و در نهایت، برای دسترسی فرانت‌اند برنامه به سیستم شما در آینده، بک‌اند خود را طوری تنظیم کنید که مرورگر اجازه ریکوئست از دامنه‌های دیگر را به شما داشته باشد.

### فریم‌ورک Spring

فریم‌ورک اسپرینگ یک فریم‌ورک بسیار گسترده برای جاوا است که به خصوص در ساخت سرویس‌های بک‌اند از آن استفاده می‌شود. از این [لینک](#) جهت آشنایی با این فریم‌ورک استفاده کنید. در جلوتر احتمالاً با کلمه Spring Boot نیز برخورد خواهید کرد. اسپرینگ بوت یک ابزار برای راحت کردن پروسه ساخت و اجرای برنامه‌ها در بستر فریم‌ورک اسپرینگ است که بسیاری از تنظیمات را به طور خودکار انجام داده و از Apache Tomcat که به همراه خود می‌آید به عنوان کاننتینر Servlet مورد نیاز استفاده می‌کند. توضیحات مربوط به Spring Boot نیز در همان لینک آمده است و خوب است که مطالعه کنید. در نهایت برای ساخت یک پروژه می‌توانید از این [لینک](#) نیز استفاده کنید. تنها dependency مورد نیاز شما به احتمال زیاد فقط Spring Web خواهد بود (spring-boot-starter-web).

جهت مهاجرت به اسپرینگ، باید با مفاهیم اصلی آن از جمله نحوه Dependency Injection و انواع annotation-های آن آشنا شوید که برخی در زیر آورده شده است:

@Bean, @AutoWired, @Configuration, @Value, @Component, @Service, @Repository, @RestController, @GetMapping, @PostMapping, @DeleteMapping, @PutMapping, @RequestBody, @RequestParam, @PathVariable, @SpringBootApplication

از آنجا که فریم‌ورک اسپرینگ در طی سال‌ها بسیار تغییر کرده است، راه‌های بسیار زیادی جهت اعمال تنظیمات مختلف (مثلاً ارور هندلینگ در صورت رخداد اکسپشن catch نشده) در آن وجود دارد. خوب است که همیشه جهت یافتن بهترین راه‌حل سرچ کنید. معمولاً پاسخ‌هایی که از xml استفاده کنند قدیمی‌تر هستند. همچنین خوب است که endpoint-های شما در بک‌اند همه ذیل مسیر api/ باشند. برای این کار می‌توانید از تنظیمات server.servlet.context-path در application.properties استفاده کنید.

## بخش چهارم: استفاده از ابزار Postman

Postman ابزاری برای ارسال درخواست با متدهای HTTP و فرمت دلخواه به یک سرور است. از این ابزار برای آزمایش و مشاهده خروجی سرویس‌هایتان استفاده کنید. لازم است تا این ابزار را در هنگام تحویل تمرین نیز روی سیستم خود داشته باشید. API-های شما باید به طور کامل در Postman وارد شده باشد و از آن طریق آزموده خواهد شد.

## Best Practice-ها

شما باید همواره همه نکات گفته شده در پروژه‌های قبلی را در پیاده‌سازی‌ها رعایت کنید.

## نکات پایانی

- این تمرین در گروه‌های حداکثر دو نفره انجام می‌شود. برای تحویل آن کافی است که یکی از اعضای گروه، لینک مخزن گیت‌هاب و Hash مربوط به آخرین کامیت پروژه را در سایت درس آپلود کند. پروژه شما بر روی این کامیت مورد ارزیابی قرار می‌گیرد.
- حتماً کاربر [IE-S04](#) را به پروژه خود اضافه کنید.
- ساختار مناسب و تمیزی کد برنامه، بخشی از نمره همه پروژه‌های شما خواهد بود. بنابراین در طراحی ساختار برنامه و همچنین خوانایی کد دقت زیادی به خرج دهید.
- هدف این تمرین یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده شباهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.
- سوالات خود را تا حد ممکن در گروه درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آنها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این تمرین ارتباط برقرار کنید.