



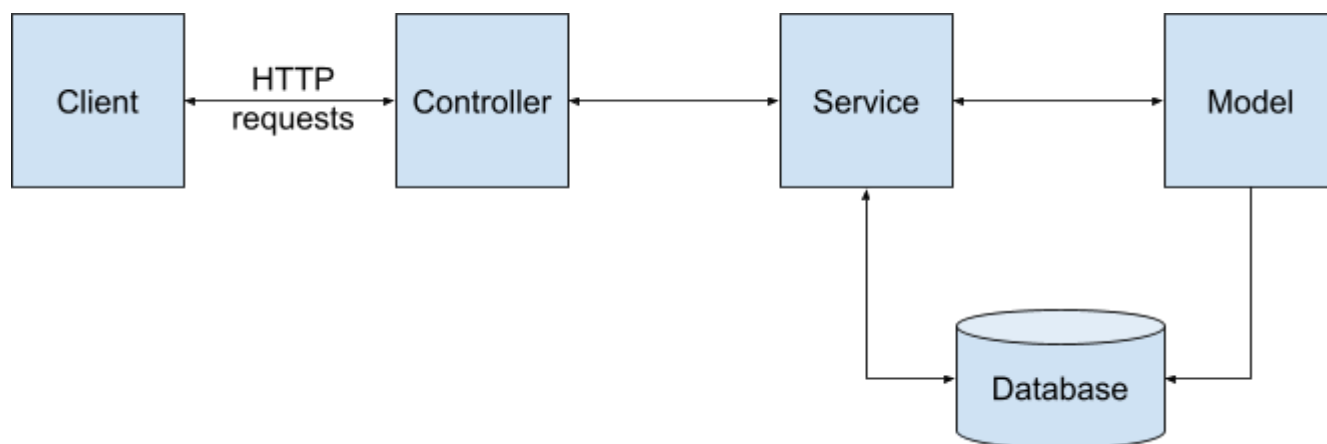
### بلوط

#### مقدمه

در این قسمت، شما با پروژه‌ای که قرار است در تمرین‌های کامپیوتری این درس با آن کار کنید، آشنا می‌شوید. همچنین، تلاش شده است تا مراحل نصب این پروژه توضیح داده شود تا بتوانید به راحتی آن را اجرا کنید. سعی کنید حتماً قبل از تمرین کامپیوتری اول، یکبار پروژه را اجرا کنید تا از درستی اجرا شدن آن روی سیستم خودتان اطمینان حاصل کنید.

#### کلیت پروژه

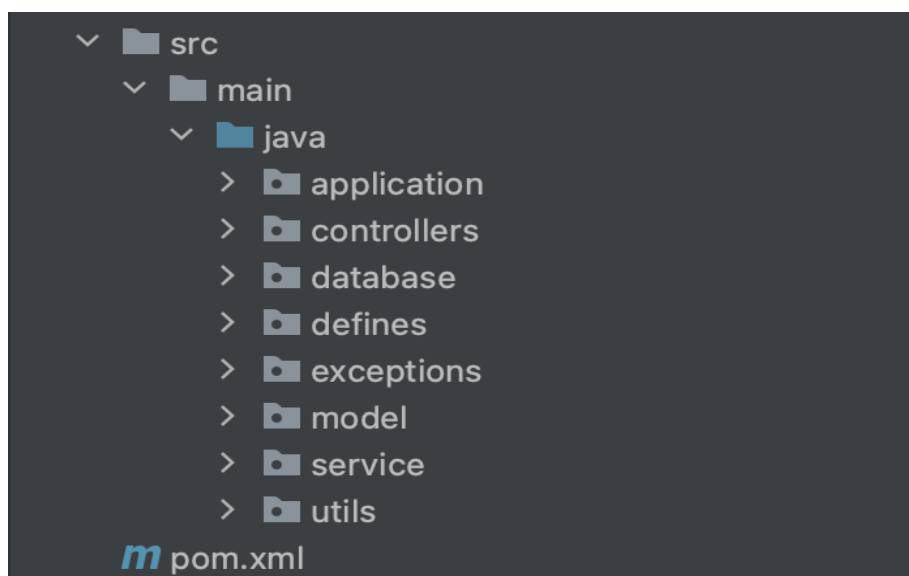
پروژه‌ای که در طول درس بر روی آن تست‌های مختلف را پیاده‌سازی خواهید کرد، یک سیستم خرید کالا است. این سیستم با چارچوب Spring پیاده‌سازی شده است. Spring یک چارچوب توسعه نرم‌افزار متن باز است که زیرساخت مورد نیاز برای ساخت برنامه‌های کاربردی که عمدتاً بر پایه جاوا می‌باشند را فراهم می‌کند. برای کسب اطلاعات بیشتر می‌توانید به این [لینک](#) مراجعه کنید. معماری یک اپلیکیشن Spring، از چندین لایه تشکیل شده است. معماری پروژه‌ای که در اختیار شما گذاشته شده است، به صورت زیر می‌باشد:



- ابتدا کلاینت یک درخواست HTTP می‌دهد. (مانند GET، POST، PUT و ...)
- این درخواست به controller فرستاده می‌شود. controller درخواست را پردازش می‌کند و متناسب با درخواست، سرویس مورد نیاز را فراخوانی می‌کند.
- در لایه Service، تمام منطق مربوط به برنامه هندل می‌شود. این لایه مسئول اجرای محاسبات و عملیات روی دیتابیس (به طور مثال اضافه کردن، کم کردن، یا تغییر عنصری در دیتابیس) می‌باشد.

## ساختار پروژه

ساختار پروژه‌ای که قرار است در طول ترم برای آن تست بنویسید، به این صورت است:



**فایل pom.xml:** این فایل یک بخش ضروری از یک پروژه جاوا مبتنی بر Maven می‌باشد. Maven یک ابزار اتوماتیک برای ساخت و مدیریت وابستگی‌های موجود در برنامه می‌باشد. یکی از مواردی که در این فایل مشخص می‌شود، کتابخانه‌های مورد نیاز پروژه است.

**model:** در داخل این دایرکتوری، دامنه پروژه مشخص شده است. دامنه یک پروژه به فضایی که پروژه به آن می‌پردازد، اشاره دارد. این نشان دهنده تمرکز اصلی یا حوزه تخصصی پروژه است. در اینجا، دامنه سیستم شامل موجودیت‌های کاربر، کالا، فراهم کننده کالا و نظر می‌باشد.

**database:** در این قسمت برای سادگی، تمام اطلاعات مربوط به این اپلیکیشن داخل مموری ذخیره‌سازی می‌شوند. در دایرکتوری data از این بخش، می‌توانید اطلاعات دلخواه خود را برای دامنه این سیستم مشخص کنید که در ابتدای اجرا شدن برنامه، این اطلاعات ذخیره خواهند شد.

**controllers:** در پروژه Spring، دایرکتوری controllers معمولا شامل کلاس‌هایی است که منطق رسیدگی به درخواست‌های HTTP و تعریف RESTful endpoints را پیاده‌سازی می‌کنند. کلاس‌های موجود در این بخش، مسئول دریافت درخواست‌های دریافتی، پردازش آنها و بازگرداندن پاسخ‌های مناسب هستند.

**service:** این بخش شامل کلاس Baloot می‌باشد. این کلاس مسئول برقراری ارتباط بین کنترلرها، دامنه برنامه و دیتابیس است.

**application:** در این بخش کلاس main این اپلیکیشن قرار داده شده است. برای اجرا کردن اپلیکیشن از این قسمت باید اقدام کنید.

## مراحل اجرای پروژه

- ۱) ابتدا از مخزن مربوط به پروژه‌های درس، پروژه اول<sup>1</sup> را clone کنید.
- ۲) پیشنهاد ما برای انجام پروژه‌های این درس، استفاده از IntelliJ IDEA<sup>2</sup> است. این ابزار به دلیل سادگی و امکان انجام خودکار بسیاری از فرآیندهای مرتبط با اجرای یک پروژه جاوا، توصیه می‌شود.
- ۳) اگر جاوا در سیستم شما نصب نشده است، ابتدا باید آن را نصب کنید. توصیه می‌شود از ورژن ۱۹ جاوا استفاده کنید، زیرا این پروژه با این ورژن آزمایش شده است و ممکن است با ورژن‌های پایین‌تر، مشکلاتی در ارتباط با ورژن کتابخانه‌ها ایجاد شود.
- ۴) ابتدا یک پروژه Maven بسازید و با ساختار ایجاد شده توسط آن آشنا شوید. همچنین، فایل pom.xml و اطلاعات داخل آن را مشاهده کنید. برای ایجاد پروژه Maven در محیط توسعه IntelliJ IDEA، می‌توانید از این [لینک](#) استفاده کنید.
- ۵) برای اجرای پروژه، کلاس BalootApplication در دایرکتوری application را اجرا کنید. این پروژه روی پورت 8080 اجرا خواهد شد.
- ۶) برای ارسال درخواست، می‌توانید از دستور curl استفاده نمایید. تمامی این دستورها در فایل README.md پروژه موجود است. همچنین می‌توانید از ابزار Postman نیز استفاده کنید. Postman ابزاری برای ارسال درخواست با متدهای HTTP و فرمت دلخواه به یک سرور است. از این ابزار می‌توان برای آزمایش و مشاهده خروجی سرویس‌ها استفاده کنید.

---

<sup>1</sup> <https://github.com/UT-Software-Testing-Course/Fall-02>

<sup>2</sup> <https://www.jetbrains.com/idea/>

برای اطلاعات بیشتر در مورد Maven می‌توانید به لینک‌های زیر مراجعه کنید.

[Maven – Maven in 5 Minutes \(apache.org\)](#)

[Maven – Maven Getting Started Guide \(apache.org\)](#)

## نکات پایانی

- پروژه‌های درس در صفحه گیت‌هاب درس<sup>3</sup> آپلود می‌شوند. لطفا قبل از شروع پروژه جدید، کد مربوط به آن را **clone** کنید و در داخل صفحه گیت‌هاب شخصی خود قرار دهید.
- توجه داشته باشید که پروژه‌ها را **fork** نکنید.
- پروژه‌های این درس در قالب گروه‌های حداکثر دو نفره انجام می‌شوند.
- برای تحویل کافیت یکی از اعضای گروه گزارش پروژه که شامل آخرین شناسه کامیت نیز می‌باشد را در صفحه درس بارگزاری نمایید.
- گزارش کار پروژه‌ها در قالب یک pdf در صفحه درس آپلود شود. توجه داشته باشید که لازم نیست کد آزمون‌های پیاده‌سازی شده را در گزارش بیاورید.
- لطفا کاربر **SoftTest-ut** را به repositoryهای مربوط به فازهای مختلف پروژه خود اضافه کنید.
- با توجه به اینکه در هنگام پیاده‌سازی این پروژه، تست برای آن نوشته نشده است، احتمالا برخی از باگ‌ها و یا سناریوهایی که به درستی پوشش داده نشده‌اند، در آن وجود دارد که با ایجاد تست می‌توان آنها را پیدا کرد.
- هدف پروژه‌های این درس، یادگیری شماست. لطفا تمرین‌ها را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.

---

<sup>3</sup> <https://github.com/UT-Software-Testing-Course>