

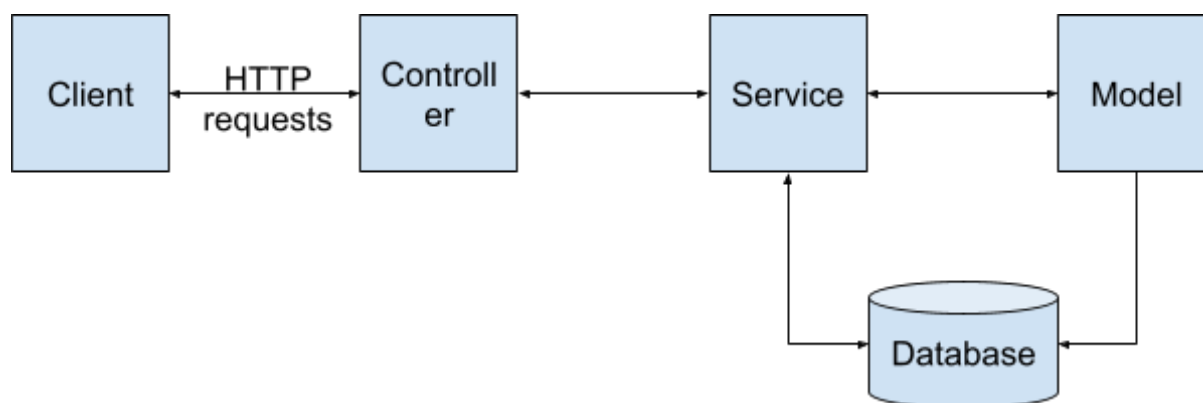


مقدمه

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

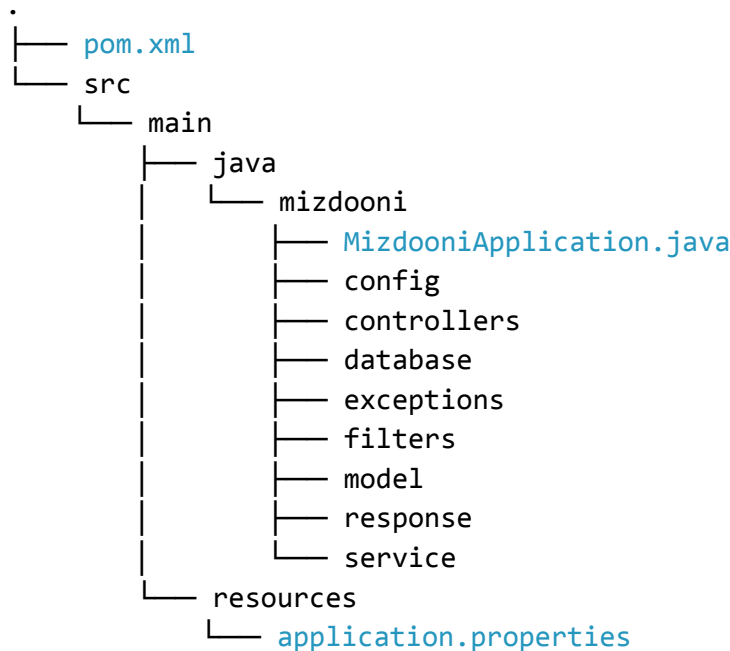
پروژه‌ای که در طول درس بر روی آن تست‌های مختلف را پیاده‌سازی خواهید کرد، یک سیستم رزرو میز در رستوران است. این سیستم با چارچوب Spring پیاده‌سازی شده است. Spring یک چارچوب توسعه نرم‌افزار متن باز است که زیرساخت مورد نیاز برای ساخت برنامه‌های کاربردی که عمدتاً بر پایه جاوا می‌باشند را فراهم می‌کند. برای 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:



1. ابتدا کلاینت یک درخواست HTTP (مانند GET یا POST یا ...) ارسال می‌کند.
2. این درخواست به controller فرستاده می‌شود؛ controller درخواست را پردازش کرده و متناسب با درخواست، سرویس مورد نیاز را فراخوانی می‌کند.
3. 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 پروژه میزدونی که در طول ترم بر روی آن کار می‌کنید، به صورت زیر است:



- فایل pom.xml: این فایل یک بخش ضروری از یک پروژه جاوا مبتنی بر Maven است، که یک ابزار اتوماتیک برای ساخت و مدیریت وابستگی‌های موجود در برنامه به شمار می‌آید. یکی از مواردی که در این فایل مشخص می‌شود، کتابخانه‌های مورد نیاز پروژه است.
- model: در داخل این پوشه، دامنه پروژه مشخص شده است. دامنه یک پروژه به فضایی که پروژه به آن می‌پردازد، اشاره دارد و نشان‌دهنده تمرکز اصلی یا حوزه تخصصی پروژه است. در اینجا، دامنه سیستم شامل موجودیت‌های کاربر، رستوران، میز، نظر و ... می‌باشد.
- database: در این قسمت برای سادگی، تمام اطلاعات مربوط به این اپلیکیشن داخل حافظه اصلی 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 معمولاً 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 دایرکتوری شامل تعدادی سرویس است به طوری که این سرویس‌ها، مسئول برقراری ارتباط بین کنترلرها، دامنه برنامه و دیتابیس هستند.
- فایل MizdooniApplication.java: در این بخش کلاس main این اپلیکیشن قرار داده شده است و برای اجرا کردن اپلیکیشن از این قسمت باید اقدام کنید.
- فایل application.properties: در این فایل بخشی از تنظیمات اپلیکیشن قرار داده شده است.

مراحل اجرای پروژه

1. ابتدا از مخزن مربوط به پروژه‌های درس، **پروژه میزدونی** را clone کنید.
 2. پیشنهاد ما برای انجام پروژه‌های این درس، استفاده از ¹ IntelliJ IDEA است. این ابزار به دلیل سادگی و امکان انجام خودکار بسیاری از فرآیندهای مرتبط با اجرای یک پروژه جاوا، توصیه می‌شود.
 3. اگر جاوا در سیستم شما نصب نشده است، ابتدا باید آن را نصب کنید. توصیه می‌شود از ورژن ۲۱ 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.
 4. ابتدا یک پروژه‌ی 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.
 5. برای اجرای پروژه، کلاس MizdooniApplication در دایرکتوری mizdooni را اجرا کنید. این پروژه روی پورت 8080 اجرا خواهد شد.
 6. 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 کرد.
- برای کسب اطلاعات بیشتر در مورد Maven می‌توانید به **این لینک** و **این لینک** مراجعه کنید.

¹ <https://www.jetbrains.com/idea/>

نکات پایانی

- پروژه‌های درس در صفحه گیت‌هاب درس² آپلود می‌شوند. لطفا قبل از شروع پروژه جدید، کد مربوط به آن را clone کنید و در داخل یک مخزن³ خصوصی⁴ در صفحه گیت‌هاب شخصی خود قرار دهید.
- توجه داشته باشید که پروژه را fork نکنید.
- پروژه در قالب گروه‌های **حداکثر دو نفره** انجام می‌شود.
- کاربر **SWT-UT** را به مخزن خود اضافه کنید.
- پاسخ سوالات بخش تئوری را در قالب یک فایل PDF در صفحه درس بارگذاری کنید. توجه داشته باشید که نیازی به ذکر کدهای بخش پیاده‌سازی در این فایل نیست؛ تنها لازم است در ابتدای این فایل، **آدرس مخزن و شناسه آخرین کامیت** خود را بنویسید.
- برای تحویل کافیست یکی از اعضای گروه فایل PDF را در صفحه درس بارگذاری نماید.
- 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. لطفا در صورت پیدا کردن باگ در هر یک از فازهای پروژه، آن باگ را اصلاح کرده و در هنگام تحویل پروژه، باگ اصلاح شده را گزارش کنید. دقت کنید که **بخشی از نمره** پروژه را اصلاح باگ‌های آن فاز تشکیل می‌دهد.
- هدف از تمرینات این درس، یادگیری شماسست؛ لطفا تمرین‌ها را خودتان انجام دهید. در صورت مشاهده مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاست ذکر شده در کلاس، کسر خواهد شد.

² <https://github.com/UT-Software-Testing-Course/Fall-03>

³ Repository

⁴ Private