BACK-END

Prof. Bruno Kurzawe

Servidores de aplicação

Um servidor de aplicação é uma estrutura de software que hospeda e opera aplicações e serviços de back-end para clientes ou usuários finais. Esse tipo de servidor desempenha um papel essencial na arquitetura client-server, permitindo que os usuários acessem e usem aplicativos de rede através de suas interfaces de cliente, enquanto a lógica de negócios, o processamento de dados e a execução do aplicativo ocorrem principalmente no servidor de aplicação.

Características e funções chave de um servidor de aplicação:

Intermediação entre o usuário e os sistemas back-end: O servidor de aplicação atua como um intermediário entre o usuário final (ou interface do cliente) e os bancos de dados ou outros recursos de back-end.

Execução da lógica de negócios: Enquanto as interfaces de cliente podem lidar com a apresentação e interação do usuário, a lógica de negócios real (como cálculos, processamento de transações ou operações de banco de dados) é frequentemente executada no servidor de aplicação.

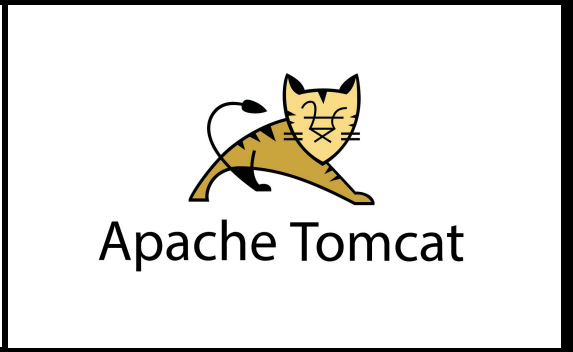
Escalabilidade: Servidores de aplicação são frequentemente projetados para lidar com um grande número de usuários simultâneos e podem ser escalados horizontalmente (adicionando mais servidores) ou verticalmente (aumentando os recursos em um servidor existente) conforme a demanda cresce.

Gestão de transações: Muitos servidores de aplicação oferecem mecanismos para gerenciar transações, garantindo que as operações sejam completadas com sucesso ou, em caso de falha, revertidas para manter a integridade dos dados.

Conexões com bancos de dados: Servidores de aplicação frequentemente interagem com bancos de dados, e eles podem oferecer pooling de conexões, onde as conexões com o banco de dados são mantidas e reutilizadas, melhorando o desempenho.

Segurança: Muitos servidores de aplicação oferecem camadas de segurança para proteger os dados e garantir que apenas usuários autorizados possam acessar certos recursos ou executar certas ações.

Exemplos de servidores de aplicação



Frameworks com servidores de aplicação

Python:

Django: Enquanto é mais frequentemente referido como um framework web, o servidor de desenvolvimento embutido e a capacidade de conectar-se a servidores de produção, como Gunicorn ou uWSGI, lhe permitem funcionar efetivamente como um servidor de aplicação.

Flask: Um micro-framework que, quando combinado com servidores como Gunicorn ou uWSGI, pode servir aplicações em ambiente de produção.

Frameworks com servidores de aplicação

Ruby:

Ruby on Rails (RoR): Embora RoR seja um framework, ele vem com o WEBrick, um servidor web simples. No entanto, em produção, servidores como Puma, Unicorn ou Passenger são comumente usados para servir aplicações RoR.

Frameworks com servidores de aplicação

JavaScript (Node.js):

Express.js: Um framework web para Node.js que, quando combinado com o próprio servidor interno de Node, atua como um servidor de aplicação.

Koa: Criado pelos mesmos desenvolvedores do Express, oferece uma base mais leve para web applications.

Frameworks com servidores de aplicação

.NET (C#):

IIS (Internet Information Services): É o servidor web da Microsoft usado para hospedar aplicações .NET.

ASP.NET Core: Uma plataforma cross-platform para construção de aplicações web modernas usando C#.

Frameworks com servidores de aplicação

PHP:

PHP-FPM (FastCGI Process Manager): É uma implementação alternativa do PHP FastCGI que pode ser usada com vários servidores web, como Nginx ou Apache.

Laravel: Um popular framework PHP que vem com um servidor de desenvolvimento embutido e pode ser usado em produção com servidores como Apache ou Nginx.

Frameworks com servidores de aplicação

Go (Golang):

net/http: A biblioteca padrão em Go inclui tudo o que é necessário para construir aplicações web, tornando-se efetivamente um servidor de aplicação simplificado.

Frameworks com servidores de aplicação

Elixir:

Phoenix: Um framework para construir aplicações web e em tempo real usando a linguagem Elixir.

Frameworks com servidores de aplicação

Rust:

Rocket: Um framework web para Rust.

Actix Web: Outro framework web para Rust com foco em desempenho.

Em resumo, um servidor de aplicação é uma ferramenta poderosa que centraliza a execução de aplicativos, facilita a gestão e escalação, e fornece recursos essenciais para aplicações empresariais modernas.


spring
boot

Spring Boot é uma extensão do framework Spring, criado para simplificar o processo de configuração e desenvolvimento de novas aplicações Spring. Ele foi desenvolvido para eliminar a complexidade associada à configuração de aplicações Spring, oferecendo uma maneira mais convencional e rápida de iniciar projetos Spring/Java.

Pontos-chave sobre o Spring Boot:

Configuração Automática: Spring Boot pode configurar automaticamente sua aplicação com base nas bibliotecas que você tem no seu projeto. Por exemplo, se o Spring Boot detectar uma biblioteca de banco de dados no classpath, ele irá configurar automaticamente uma conexão de banco de dados.

Produção Pronta: O Spring Boot tem recursos integrados como saúde, métricas e verificações de aplicação, o que o torna fácil de monitorar e gerenciar em produção.

Sem Código Gerado: A abordagem do Spring Boot é diferente de muitos outros frameworks de "scaffolding". Ele não gera código e não há requisito de configuração XML.

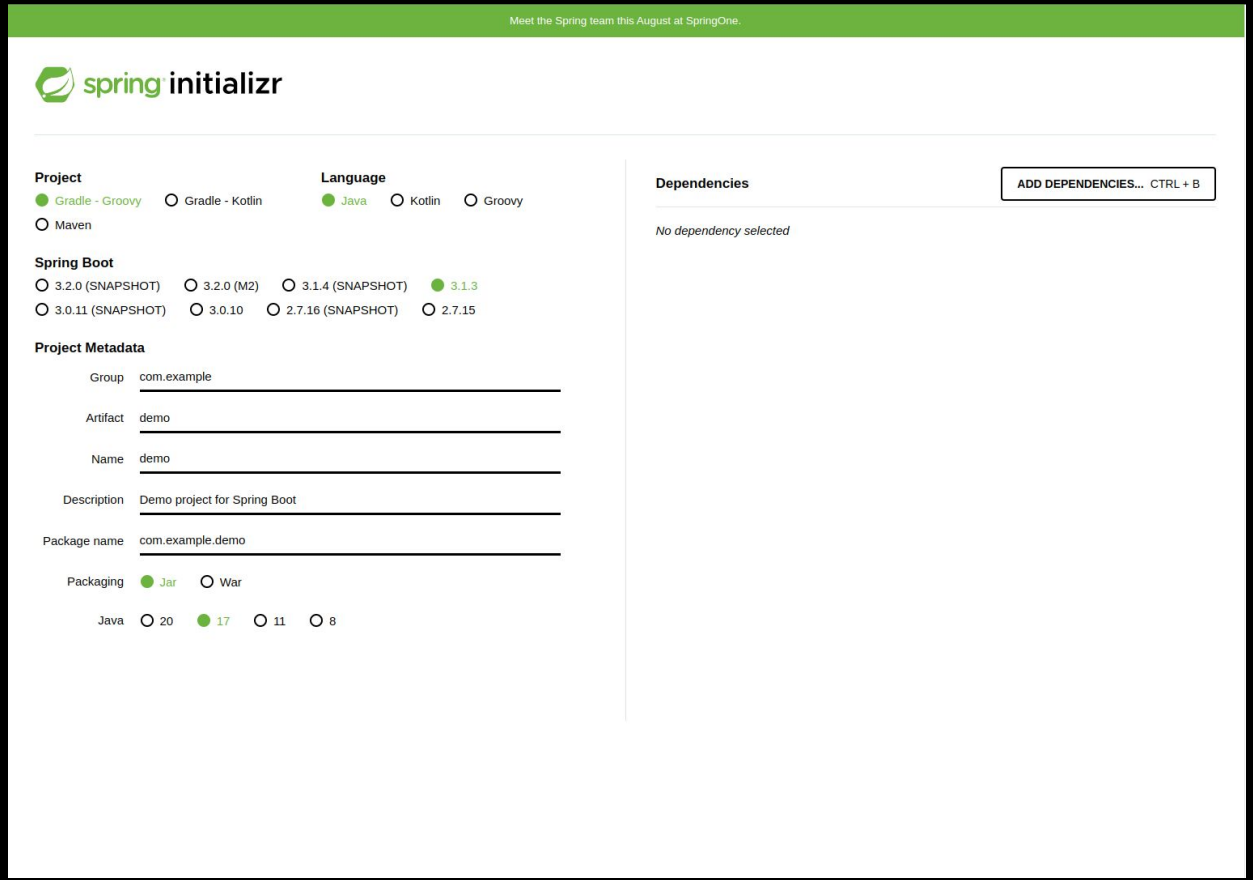
Flexível: Embora o Spring Boot seja "opinionated", ele não sacrifica a flexibilidade. Se você não concordar com as opiniões padrão, pode facilmente substituí-las.

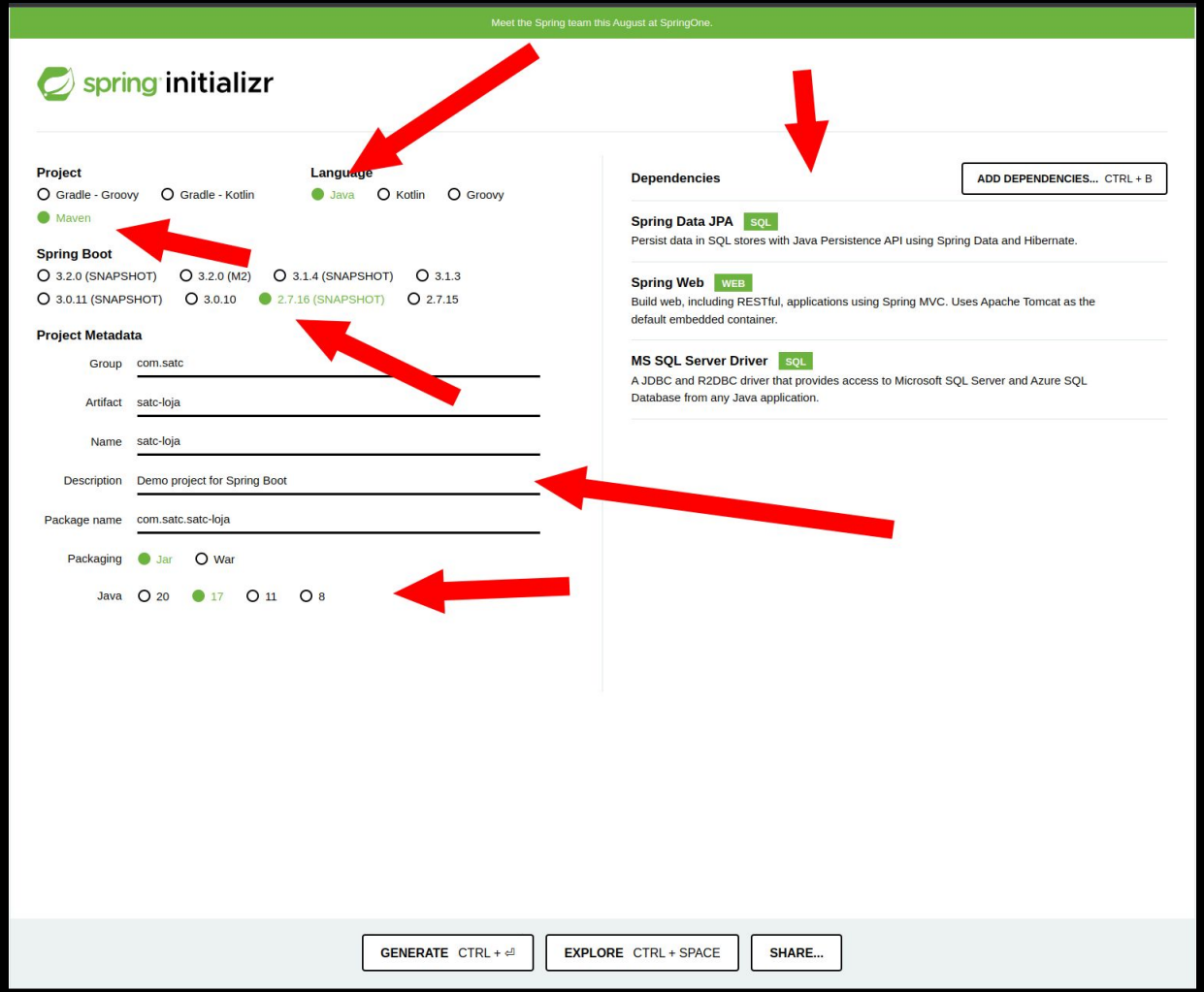
Embed Servers: Spring Boot pode embedar servidores de aplicação como Tomcat, Jetty e Undertow diretamente no JAR final, permitindo que aplicações sejam executadas como aplicações Java autônomas.

Gerenciador de Dependências: Com o Spring Boot Starter POMs, as dependências podem ser gerenciadas de forma centralizada, assegurando que as versões de bibliotecas sejam compatíveis entre si.

Spring Boot Initializr: Uma ferramenta web que permite criar rapidamente um novo projeto Spring Boot com as dependências que você escolher.

https://start.spring.io/


Meet the Spring team this August at SpringOne.
spring initializr
Project
Gradle - Groovy
Gradle - Kotlin
Maven
Language
Java
Kotlin
Groovy
Spring Boot
3.2.0 (SNAPSHOT)
3.2.0 (M2)
3.1.4 (SNAPSHOT)
3.1.3
3.0.11 (SNAPSHOT)
3.0.10
2.7.16 (SNAPSHOT)
2.7.15
Project Metadata
Group
com.example
Artifact
demo
Name
demo
Description
Demo project for Spring Boot
Package name
com.example.demo
Packaging
Jar
War
Java
20
17
11
8
Dependencies
ADD DEPENDENCIES... CTRL + B
No dependency selected


Meet the Spring team this August at SpringOne.
spring initializr
Project
Gradle - Groovy
Gradle - Kotlin
Maven
Language
Java
Kotlin
Groovy
Spring Boot
3.2.0 (SNAPSHOT)
3.2.0 (M2)
3.1.4 (SNAPSHOT)
3.1.3
3.0.11 (SNAPSHOT)
3.0.10
2.7.16 (SNAPSHOT)
2.7.15
Project Metadata
Group
com.satc
Artifact
satc-loja
Name
satc-loja
Description
Demo project for Spring Boot
Package name
com.satc.satc-loja
Packaging
Jar
War
Java
20
17
11
8
Dependencies
ADD DEPENDENCIES... CTRL + B
Spring Data JPA
SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
Spring Web
WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
MS SQL Server Driver
SQL
A JDBC and R2DBC driver that provides access to Microsoft SQL Server and Azure SQL Database from any Java application.
GENERATE CTRL + ⏎
EXPLORE CTRL + SPACE
SHARE...

Dependencies ADD DEPENDENCIES... CTRL + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

MS SQL Server Driver SQL

A JDBC and R2DBC driver that provides access to Microsoft SQL Server and Azure SQL Database from any Java application.

Meet the Spring team this August at SpringOne.

spring initializr

Project

- Gradle - Groovy
- Gradle - Kotlin
- Maven

Language

- Java
- Kotlin
- Groovy

Spring Boot

- 3.2.0 (SNAPSHOT)
- 3.2.0 (M2)
- 3.1.4 (SNAPSHOT)
- 3.1.3
- 3.0.11 (SNAPSHOT)
- 3.0.10
- 2.7.16 (SNAPSHOT)
- 2.7.15

Project Metadata

Group	com.satc
Artifact	satc-loja
Name	satc-loja
Description	Exemplo para os alunos da 4 fase SATC
Package name	com.satc.satc-loja

Packaging: Jar / War

Java: 20 / 17 / 11 / 8

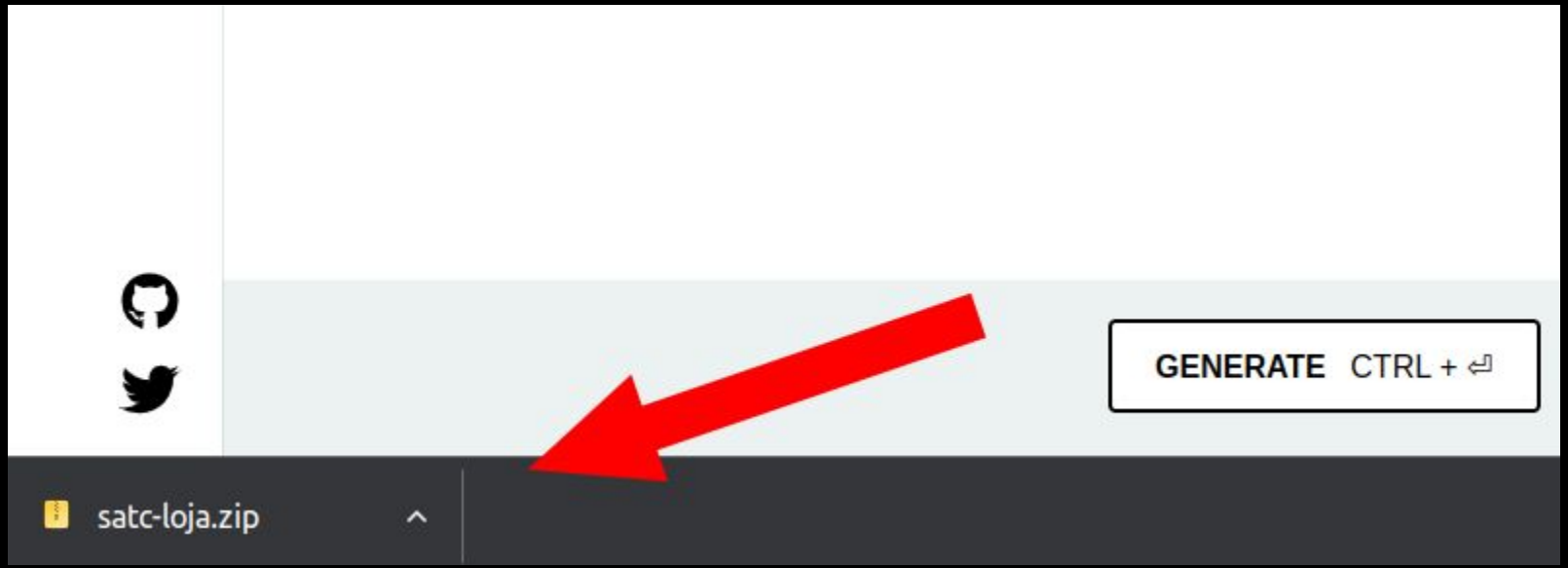
Dependencies — ADD DEPENDENCIES... CTRL + B

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

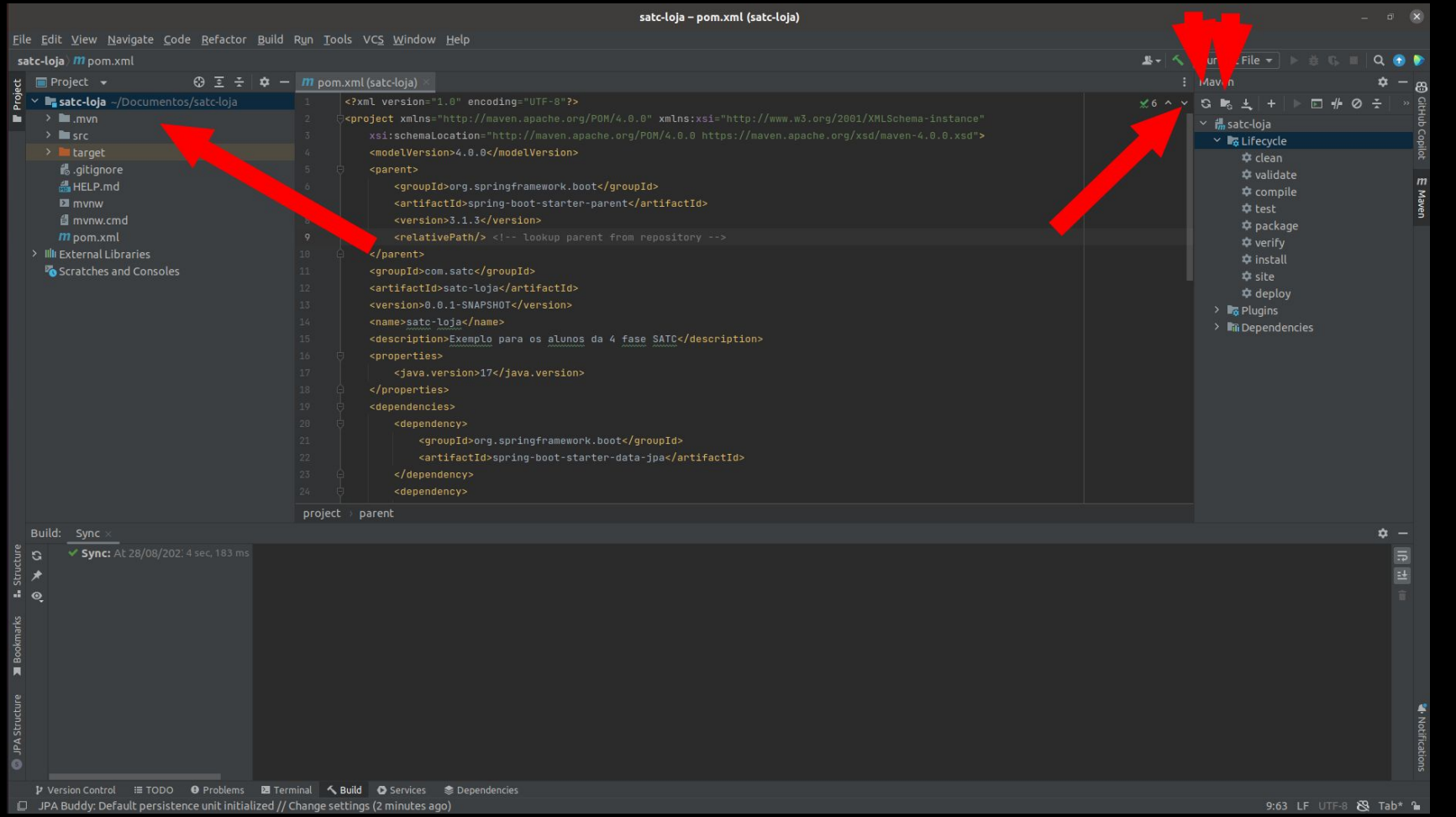
Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

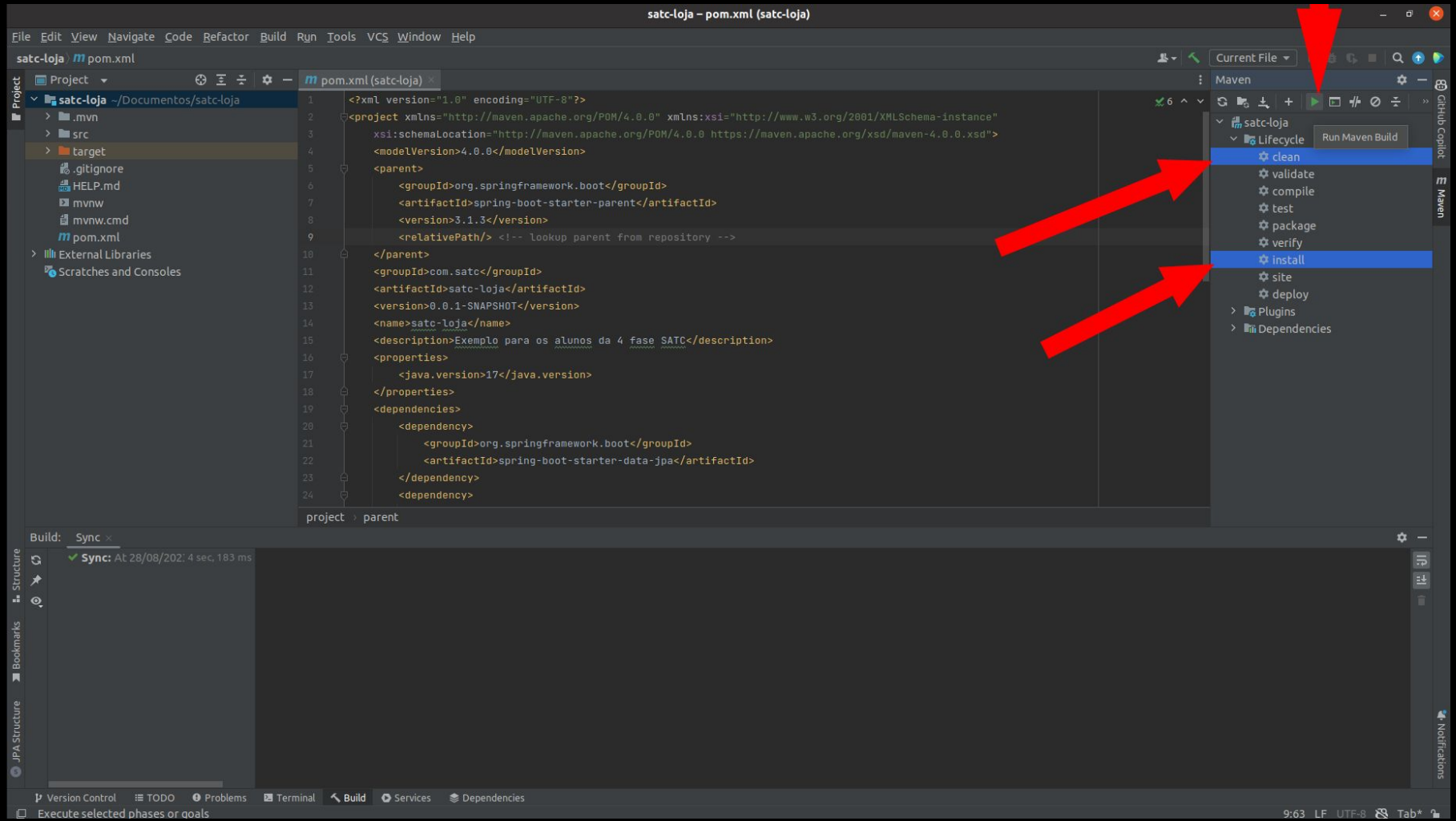
MS SQL Server Driver SQL
A JDBC and R2DBC driver that provides access to Microsoft SQL Server and Azure SQL Database from any Java application.

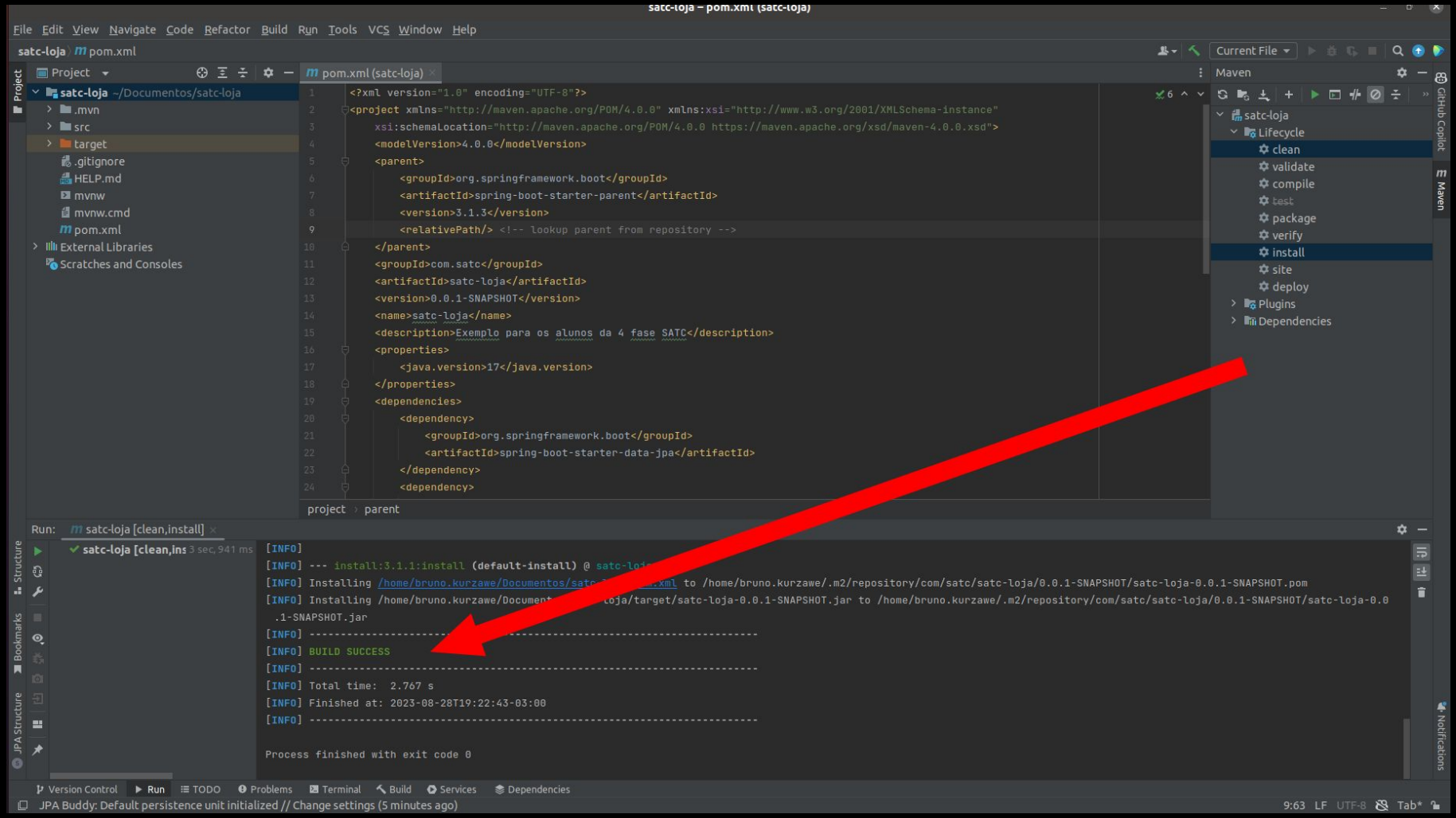
GENERATE CTRL + ⏎ | EXPLORE CTRL + SPACE | SHARE...


GENERATE CTRL + ⏎
satc-loja.zip

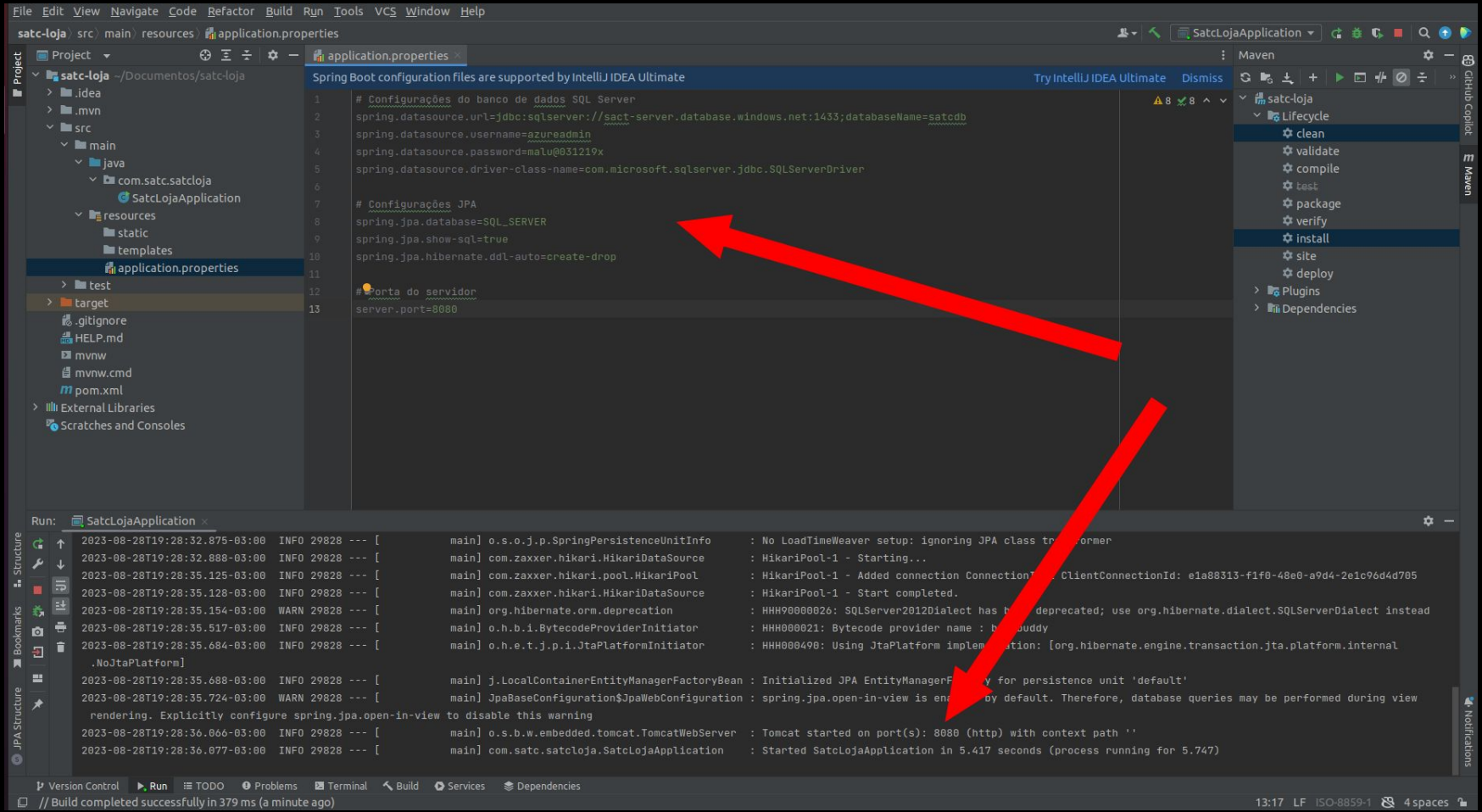
Vamos extrair esse arquivo em uma pasta para podermos abrir com o Intellij.


satc-loja – pom.xml (satc-loja)
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
satc-loja pom.xml
Project
satc-loja ~/Documentos/satc-loja
.mvn
src
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
pom.xml (satc-loja)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.1.3</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.satc</groupId>
<artifactId>satc-loja</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>satc-loja</name>
<description>Exemplo para os alunos da 4 fase SATC</description>
<properties>
<java.version>17</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
project parent
Maven
satc-loja
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Dependencies
Build: Sync
Sync: At 28/08/202: 4 sec, 183 ms
Version Control TODO Problems Terminal Build Services Dependencies
JPA Buddy: Default persistence unit initialized // Change settings (2 minutes ago)
9:63 LF UTF-8 Tab*


satc-loja – pom.xml (satc-loja)
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
satc-loja pom.xml
Current File
Project
satc-loja ~/Documentos/satc-loja
.mvn
src
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
pom.xml (satc-loja)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.1.3</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.satc</groupId>
<artifactId>satc-loja</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>satc-loja</name>
<description>Exemplo para os alunos da 4 fase SATC</description>
<properties>
<java.version>17</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
project parent
Maven
satc-loja
Lifecycle
Run Maven Build
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Dependencies
Build: Sync
Sync: At 28/08/202: 4 sec, 183 ms
Version Control TODO Problems Terminal Build Services Dependencies
Execute selected phases or goals
9:63 LF UTF-8 Tab*


satc-loja – pom.xml (satc-loja)
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
satc-loja pom.xml
Current File
Project
satc-loja ~/Documentos/satc-loja
.mvn
src
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
pom.xml (satc-loja)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4 <modelVersion>4.0.0</modelVersion>
5 <parent>
6 <groupId>org.springframework.boot</groupId>
7 <artifactId>spring-boot-starter-parent</artifactId>
8 <version>3.1.3</version>
9 <relativePath/> <!-- lookup parent from repository -->
10 </parent>
11 <groupId>com.satc</groupId>
12 <artifactId>satc-loja</artifactId>
13 <version>0.0.1-SNAPSHOT</version>
14 <name>satc-loja</name>
15 <description>Exemplo para os alunos da 4 fase SATC</description>
16 <properties>
17 <java.version>17</java.version>
18 </properties>
19 <dependencies>
20 <dependency>
21 <groupId>org.springframework.boot</groupId>
22 <artifactId>spring-boot-starter-data-jpa</artifactId>
23 </dependency>
24 <dependency>
project parent
Maven
satc-loja
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Dependencies
Run: satc-loja [clean,install]
satc-loja [clean,ins 3 sec, 941 ms
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ satc-loj
[INFO] Installing /home/bruno.kurzawe/Documentos/satc .xml to /home/bruno.kurzawe/.m2/repository/com/satc/satc-loja/0.0.1-SNAPSHOT/satc-loja-0.0.1-SNAPSHOT.pom
[INFO] Installing /home/bruno.kurzawe/Document loja/target/satc-loja-0.0.1-SNAPSHOT.jar to /home/bruno.kurzawe/.m2/repository/com/satc/satc-loja/0.0.1-SNAPSHOT/satc-loja-0.0
.1-SNAPSHOT.jar
[INFO] ----------
[INFO] BUILD SUCCESS
[INFO] ----------
[INFO] Total time: 2.767 s
[INFO] Finished at: 2023-08-28T19:22:43-03:00
[INFO] ----------
Process finished with exit code 0
Version Control Run TODO Problems Terminal Build Services Dependencies
JPA Buddy: Default persistence unit initialized // Change settings (5 minutes ago)
9:63 LF UTF-8 Tab*

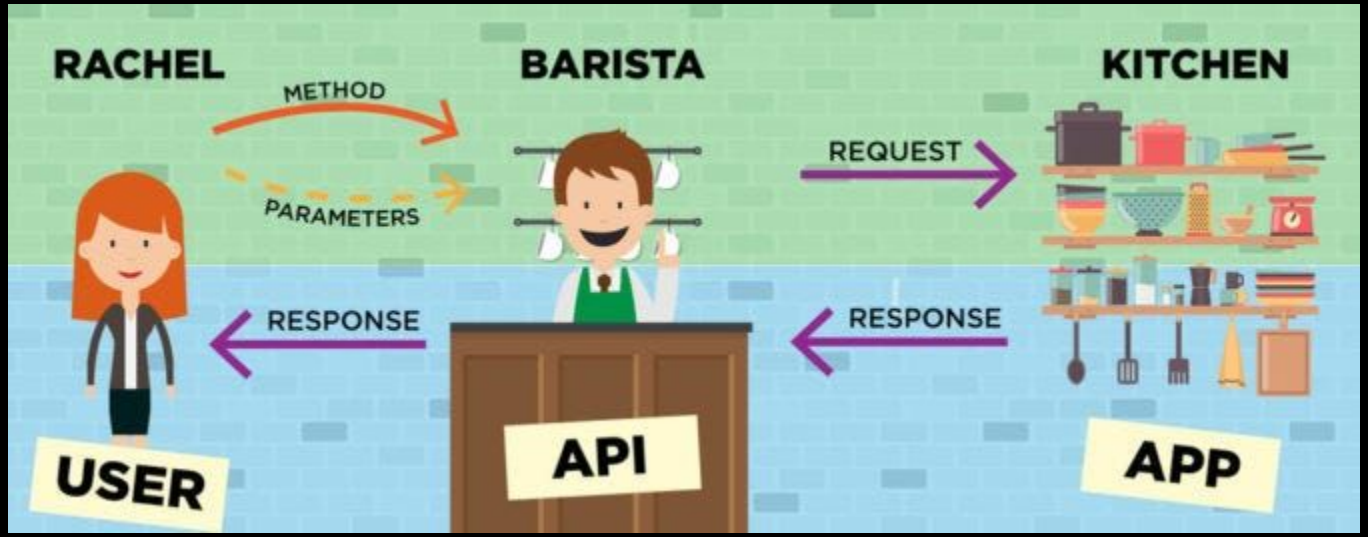
Agora vamos configurar nosso application.properties


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
satc-loja src main resources application.properties
SatcLojaApplication
Project
satc-loja ~/Documentos/satc-loja
.idea
.mvn
src
main
java
com.satc.satcloja
SatcLojaApplication
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
application.properties
Spring Boot configuration files are supported by IntelliJ IDEA Ultimate
Try IntelliJ IDEA Ultimate
Dismiss
1 # Configurações do banco de dados SQL Server
2 spring.datasource.url=jdbc:sqlserver://sact-server.database.windows.net:1433;databaseName=satcdb
3 spring.datasource.username=azureadmin
4 spring.datasource.password=malu@031219x
5 spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver
6
7 # Configurações JPA
8 spring.jpa.database=SQL_SERVER
9 spring.jpa.show-sql=true
10 spring.jpa.hibernate.ddl-auto=create-drop
11
12 #Porta do servidor
13 server.port=8080
Maven
satc-loja
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Dependencies
Run: SatcLojaApplication
2023-08-28T19:28:32.875-03:00 INFO 29828 --- [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer
2023-08-28T19:28:32.888-03:00 INFO 29828 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-08-28T19:28:35.125-03:00 INFO 29828 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection ConnectionID, ClientConnectionId: e1a88313-f1f0-48e0-a9d4-2e1c96d4d705
2023-08-28T19:28:35.128-03:00 INFO 29828 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-08-28T19:28:35.154-03:00 WARN 29828 --- [main] org.hibernate.orm.deprecation : HHH90000026: SQLServer2012Dialect has been deprecated; use org.hibernate.dialect.SQLServerDialect instead
2023-08-28T19:28:35.517-03:00 INFO 29828 --- [main] o.h.b.i.BytecodeProviderInitiator : HHH000021: Bytecode provider name : bytebuddy
2023-08-28T19:28:35.684-03:00 INFO 29828 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2023-08-28T19:28:35.688-03:00 INFO 29828 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-08-28T19:28:35.724-03:00 WARN 29828 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2023-08-28T19:28:36.066-03:00 INFO 29828 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-08-28T19:28:36.077-03:00 INFO 29828 --- [main] com.satc.satcloja.SatcLojaApplication : Started SatcLojaApplication in 5.417 seconds (process running for 5.747)
Version Control
Run
TODO
Problems
Terminal
Build
Services
Dependencies
// Build completed successfully in 379 ms (a minute ago)
13:17 LF ISO-8859-1 4 spaces

Até aqui, configuramos uma aplicação SringBoot

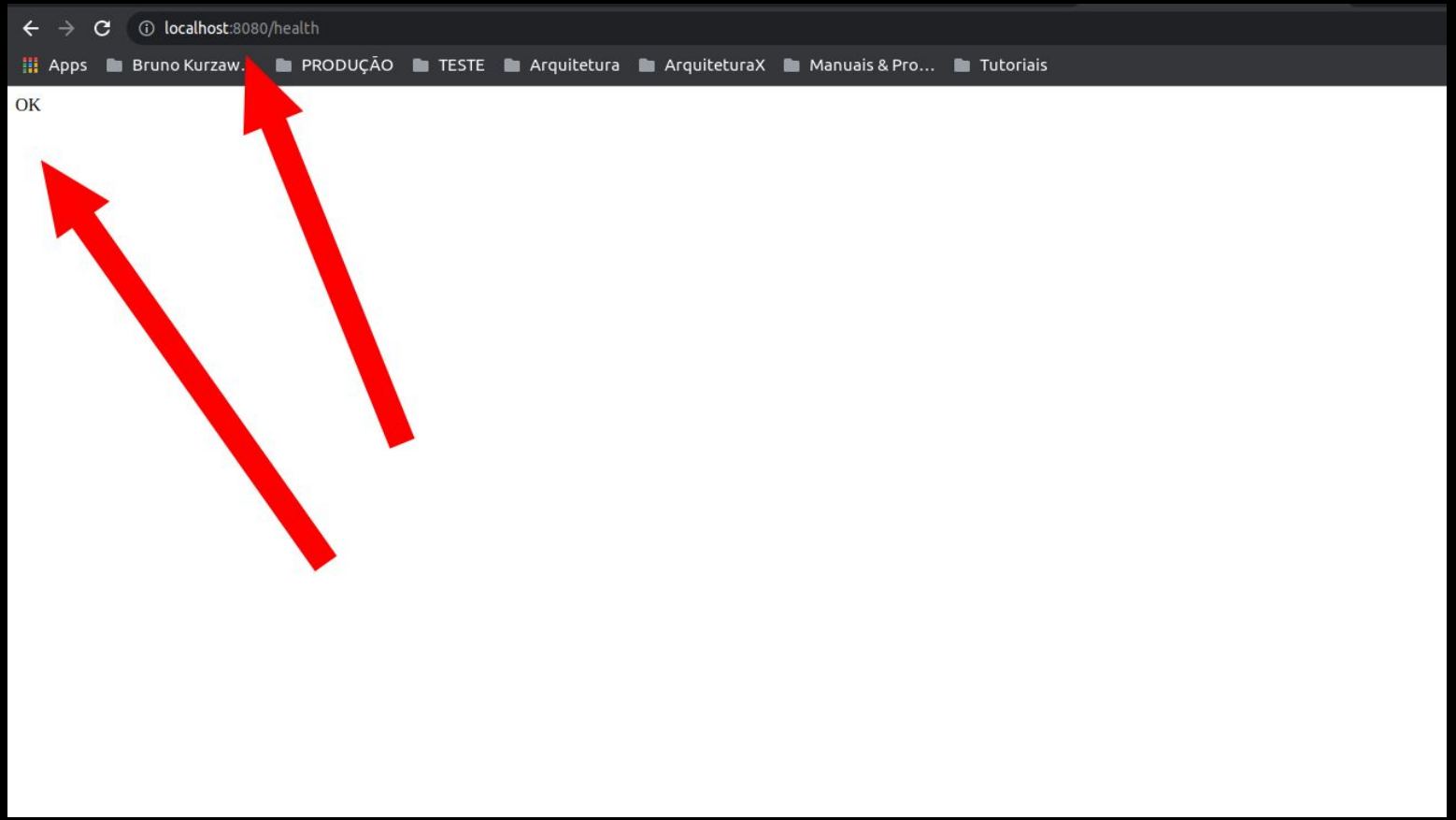
Vamos criar uma API de healthCheck, criem um pacote chamado health, dentro deste pacote vamos criar uma classe chamada HealthCheckController.

API significa "Interface de Programação de Aplicativos" (em inglês, "Application Programming Interface"). É um conjunto de regras, protocolos e ferramentas que permite que diferentes componentes de software interagem entre si. Em outras palavras, uma API define como os diferentes sistemas de software devem se comunicar e trocar informações.


RACHEL
METHOD
PARAMETERS
BARISTA
KITCHEN
REQUEST
RESPONSE
RESPONSE
USER
API
APP

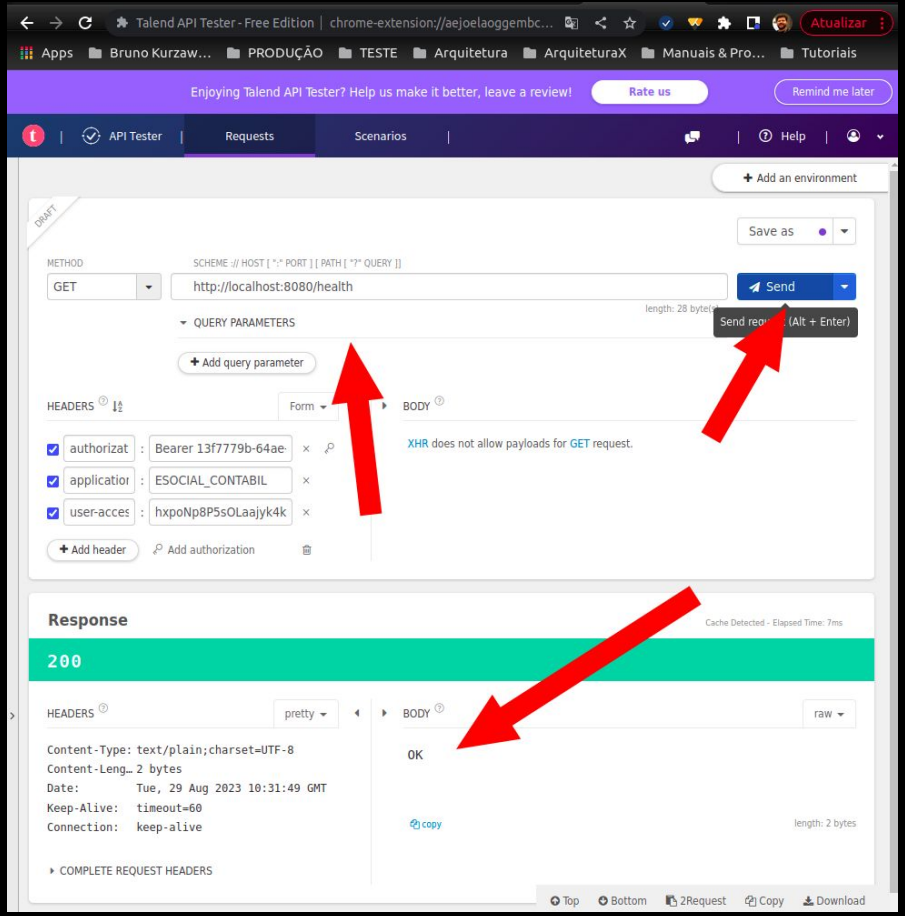
Vamos reiniciar a nossa aplicação

Vamos testar nossa API, como é uma api de GET (logo mais vamos aprender mais sobre isso) podemos testar direto no navegador.

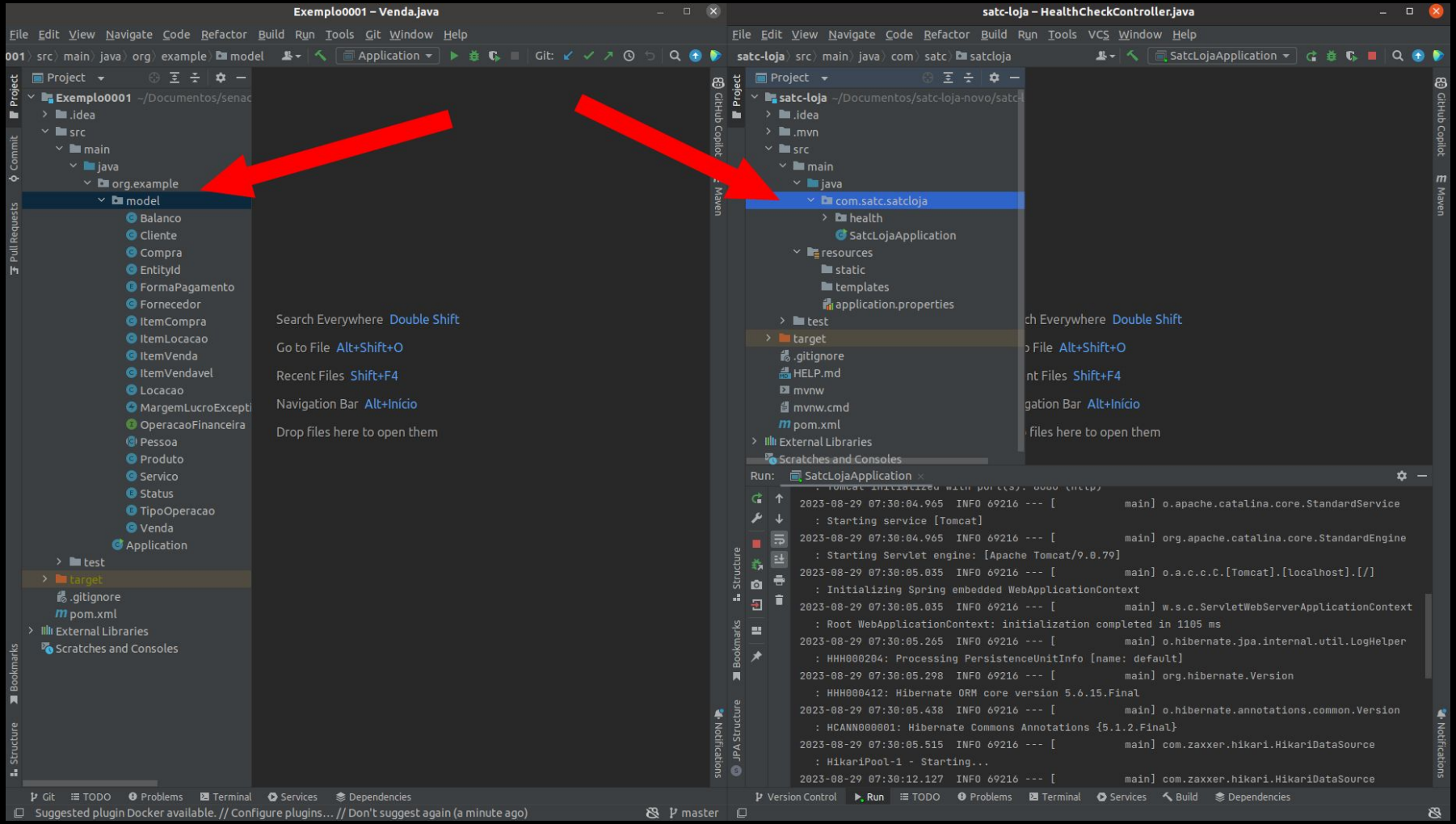

localhost:8080/health
Apps
Bruno Kurzaw...
PRODUÇÃO
TESTE
Arquitetura
ArquiteturaX
Manuais & Pro...
Tutoriais
OK

Vamos testar nossa API, como é uma api de GET (logo mais vamos aprender mais sobre isso) podemos testar direto no navegador.

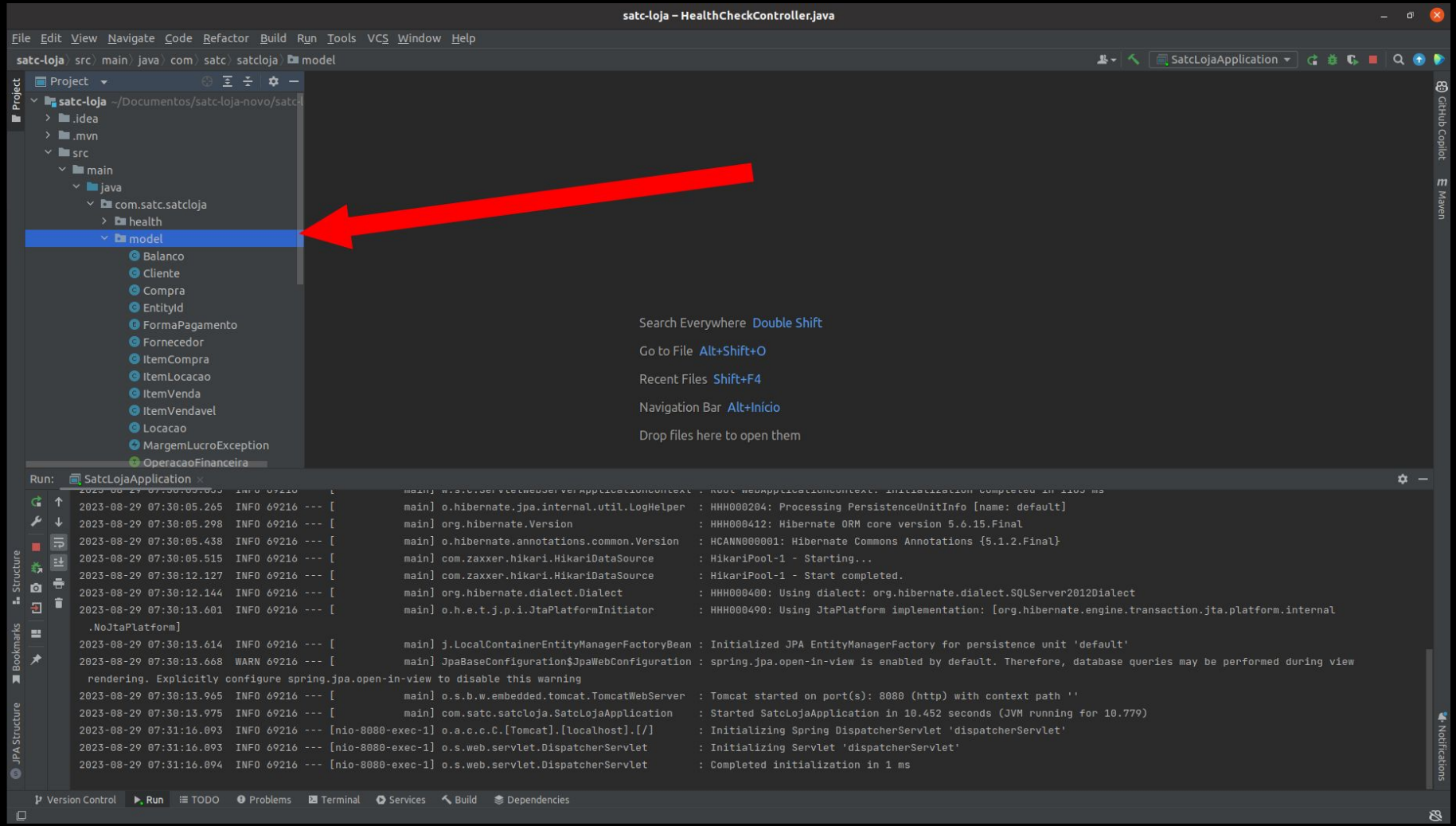
Podemos testar usando ferramentas de test de API



Agora vamos pegar o nosso projeto da loja


Exemplo0001 – Venda.java
File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Project
Exemplo0001 ~/Documentos/senac
.idea
src
main
java
org.example
model
Balanco
Cliente
Compra
EntityId
FormaPagamento
Fornecedor
ItemCompra
ItemLocacao
ItemVenda
ItemVendavel
Locacao
MargemLucroExcepti
OperacaoFinanceira
Pessoa
Produto
Servico
Status
TipoOperacao
Venda
Application
test
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
Search Everywhere Double Shift
Go to File Alt+Shift+O
Recent Files Shift+F4
Navigation Bar Alt+Início
Drop files here to open them
Git TODO Problems Terminal Services Dependencies
Suggested plugin Docker available. // Configure plugins... // Don't suggest again (a minute ago)
master
satc-loja – HealthCheckController.java
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
satc-loja ~/Documentos/satc-loja-novo/satc-
.idea
.mvn
src
main
java
com.satc.satcloja
health
SatcLojaApplication
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
Run: SatcLojaApplication
2023-08-29 07:30:04.965 INFO 69216 --- [main] o.apache.catalina.core.StandardService
: Starting service [Tomcat]
2023-08-29 07:30:04.965 INFO 69216 --- [main] org.apache.catalina.core.StandardEngine
: Starting Servlet engine: [Apache Tomcat/9.0.79]
2023-08-29 07:30:05.035 INFO 69216 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/]
: Initializing Spring embedded WebApplicationContext
2023-08-29 07:30:05.035 INFO 69216 --- [main] w.s.c.ServletWebServerApplicationContext
: Root WebApplicationContext: initialization completed in 1105 ms
2023-08-29 07:30:05.265 INFO 69216 --- [main] o.hibernate.jpa.internal.util.LogHelper
: HHH000204: Processing PersistenceUnitInfo [name: default]
2023-08-29 07:30:05.298 INFO 69216 --- [main] org.hibernate.Version
: HHH000412: Hibernate ORM core version 5.6.15.Final
2023-08-29 07:30:05.438 INFO 69216 --- [main] o.hibernate.annotations.common.Version
: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2023-08-29 07:30:05.515 INFO 69216 --- [main] com.zaxxer.hikari.HikariDataSource
: HikariPool-1 - Starting...
2023-08-29 07:30:12.127 INFO 69216 --- [main] com.zaxxer.hikari.HikariDataSource
Version Control Run TODO Problems Terminal Services Build Dependencies

Vamos copiar o conteúdo do model para dentro da nossa aplicação SpringBoot.


satc-loja – HealthCheckController.java
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
satc-loja src main java com satc satcloja model
SatcLojaApplication
Project
satc-loja ~/Documentos/satc-loja-novo/satc
.idea
.mvn
src
main
java
com.satc.satcloja
health
model
Balanco
Cliente
Compra
EntityId
FormaPagamento
Fornecedor
ItemCompra
ItemLocacao
ItemVenda
ItemVendavel
Locacao
MargemLucroException
OperacaoFinanceira
Search Everywhere Double Shift
Go to File Alt+Shift+O
Recent Files Shift+F4
Navigation Bar Alt+Início
Drop files here to open them
Run: SatcLojaApplication
2023-08-29 07:30:05.265 INFO 69216 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-08-29 07:30:05.298 INFO 69216 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.15.Final
2023-08-29 07:30:05.438 INFO 69216 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2023-08-29 07:30:05.515 INFO 69216 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-08-29 07:30:12.127 INFO 69216 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-08-29 07:30:12.144 INFO 69216 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.SQLServer2012Dialect
2023-08-29 07:30:13.601 INFO 69216 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2023-08-29 07:30:13.614 INFO 69216 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-08-29 07:30:13.668 WARN 69216 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2023-08-29 07:30:13.965 INFO 69216 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-08-29 07:30:13.975 INFO 69216 --- [main] com.satc.satcloja.SatcLojaApplication : Started SatcLojaApplication in 10.452 seconds (JVM running for 10.779)
2023-08-29 07:31:16.093 INFO 69216 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-08-29 07:31:16.093 INFO 69216 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-08-29 07:31:16.094 INFO 69216 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
Version Control Run TODO Problems Terminal Services Build Dependencies
GitHub Copilot
Maven
Structure
Bookmarks
JPA Structure
Notifications

Agora temos no nosso projeto todas as classes que criamos nas últimas aulas. Vamos rodar novamente!

Fim da aula 02…