

VR Balance API - Installation

Technical stack

- Language: Java 21
- Framework: Spring Boot
- API documentation: Swagger
- Gradle 8.13
- Database: PostgreSQL

For installation, I recommend using the IntelliJ IDE by JetBrains.

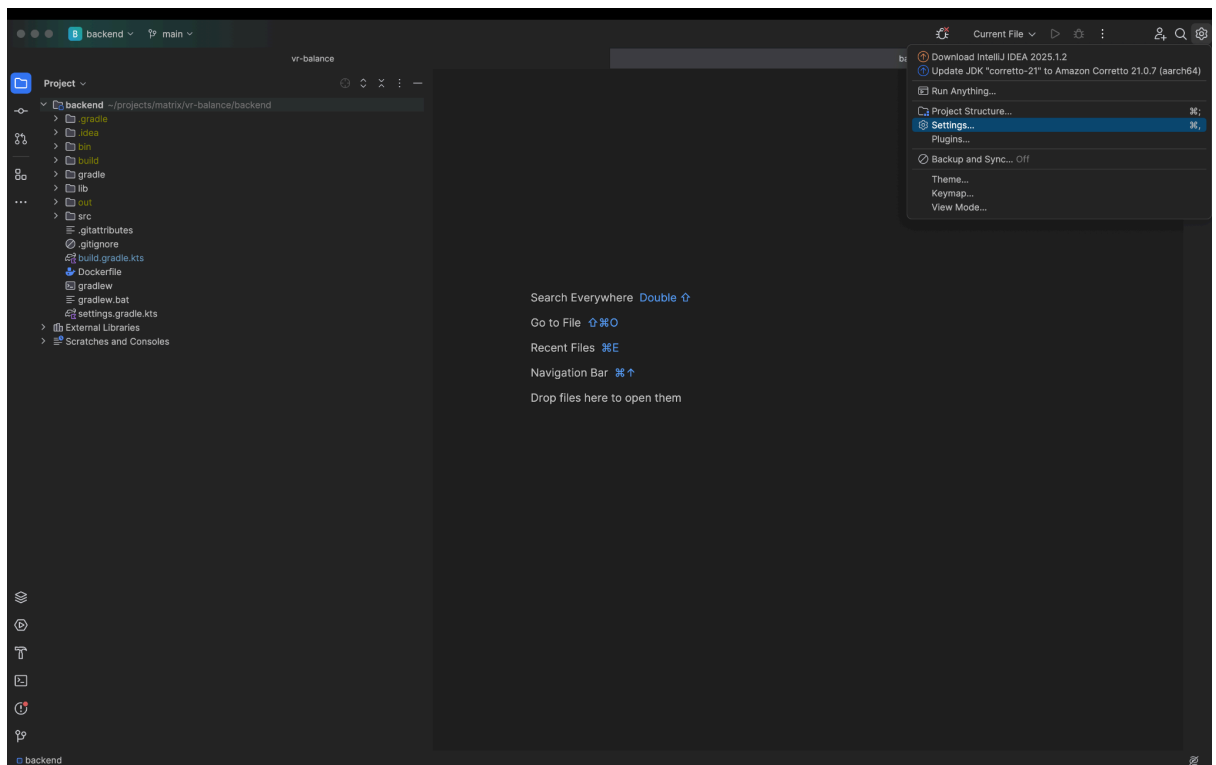
While it is a paid tool, you can get it for free with your student account.

If you prefer not to use IntelliJ, you'll need to install Gradle and Java 21 manually.

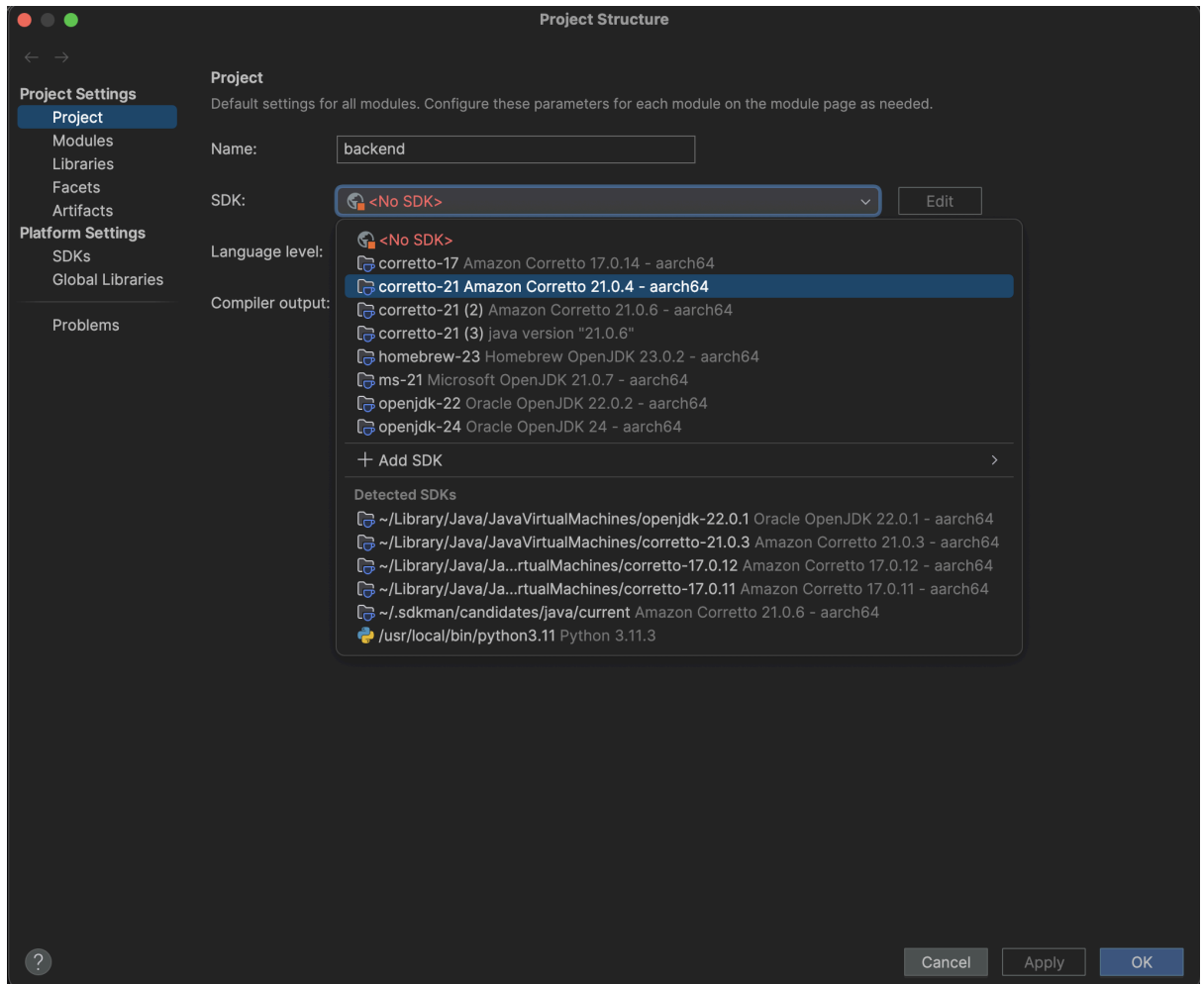
Once installed, navigate to the backend folder in your terminal and run the following command to start the project: **./gradlew bootRun**

IntelliJ installation

1. Open the project (the “backend” folder).
2. Go to “project structure” in the settings-icon

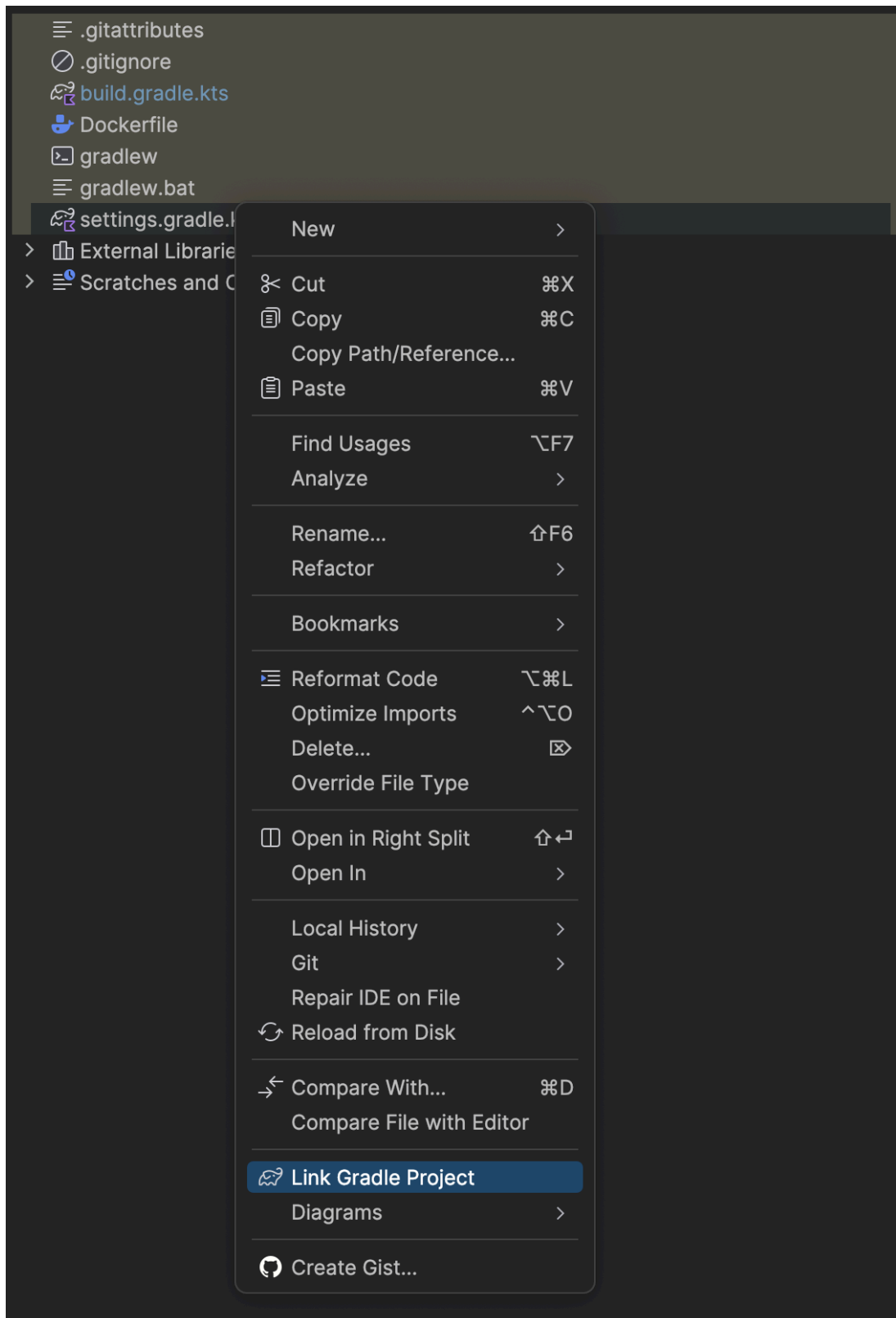


3. Choose an SDK that supports Java 21. I am using Corretto 21.

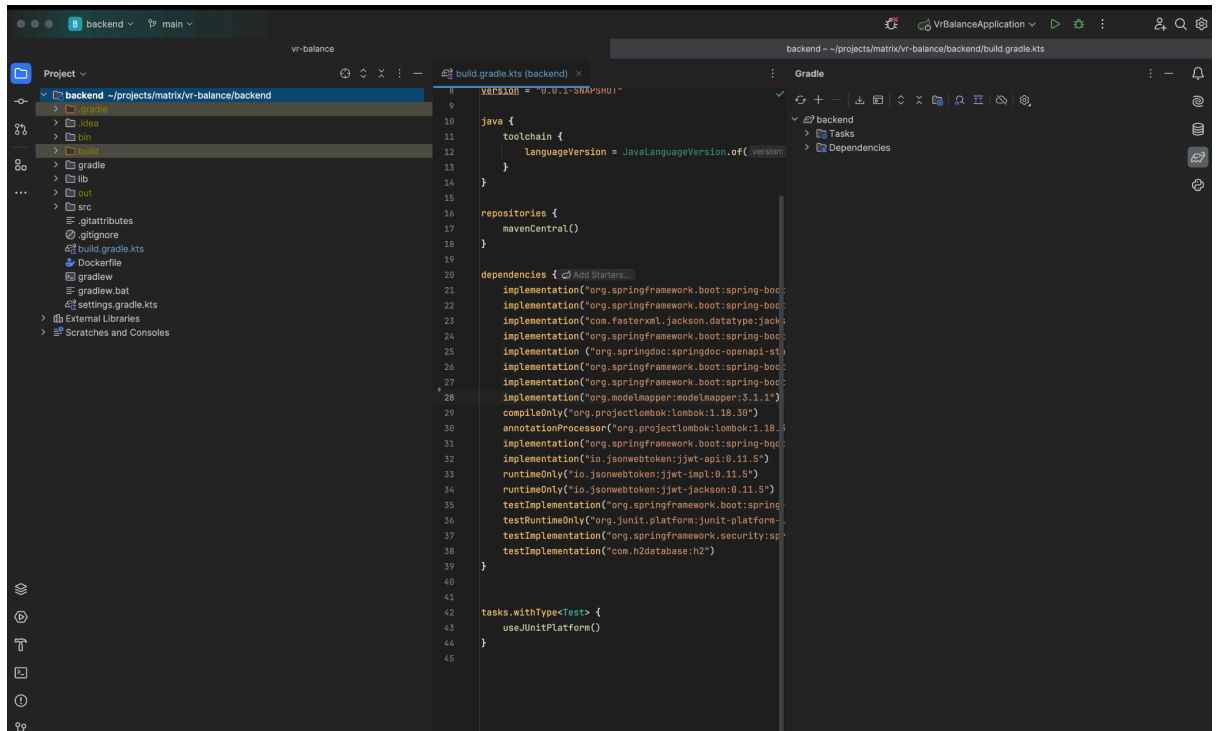


4. Click "apply" and then "ok"

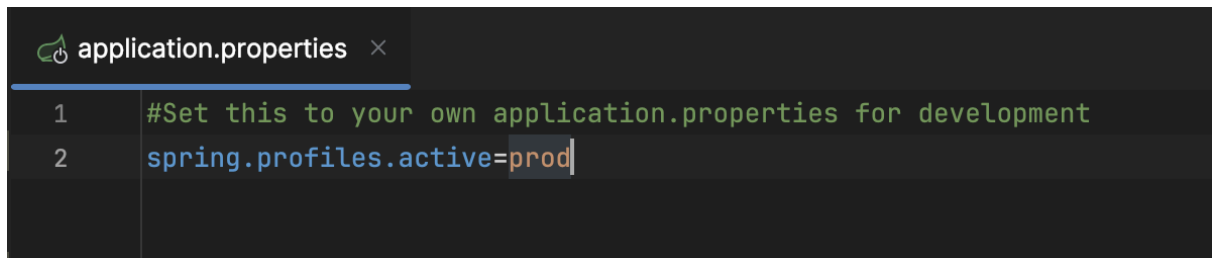
5. Right-click on the build.gradle.kts file and select **Link Gradle Project**.



7. You should now see an elephant icon on the right, along with a button labeled `VrBalanceApplication` to start the app.



8. Go to the `application.properties` file located in `src/main/resources/` and set the profile to `dev`. The `dev` is for development, while `prod` is for production.



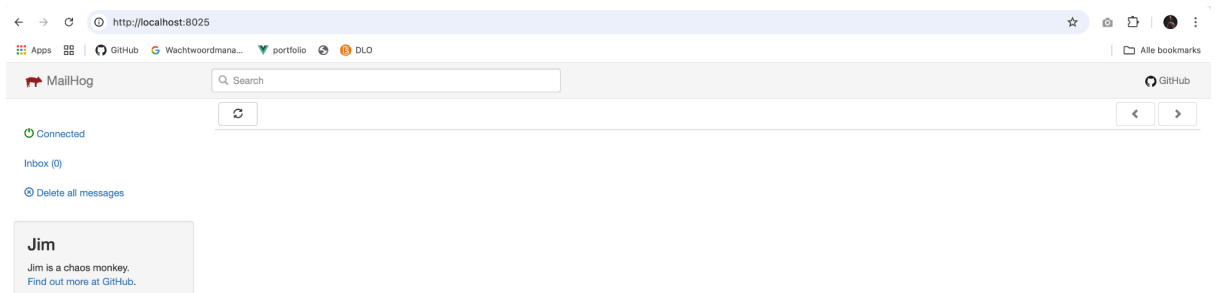
9. For sending emails locally and during development, I use [MailHog](#).

10. Make sure [Docker](#) is installed.

11. Run the following commands in the terminal:

```
docker run -d \
--name mailhog \
-p 8025:8025 \
-p 1025:1025 \
mailhog/mailhog
```

12. Navigate to <http://localhost:8025/>. You should now see a web interface that captures and displays all outgoing emails.

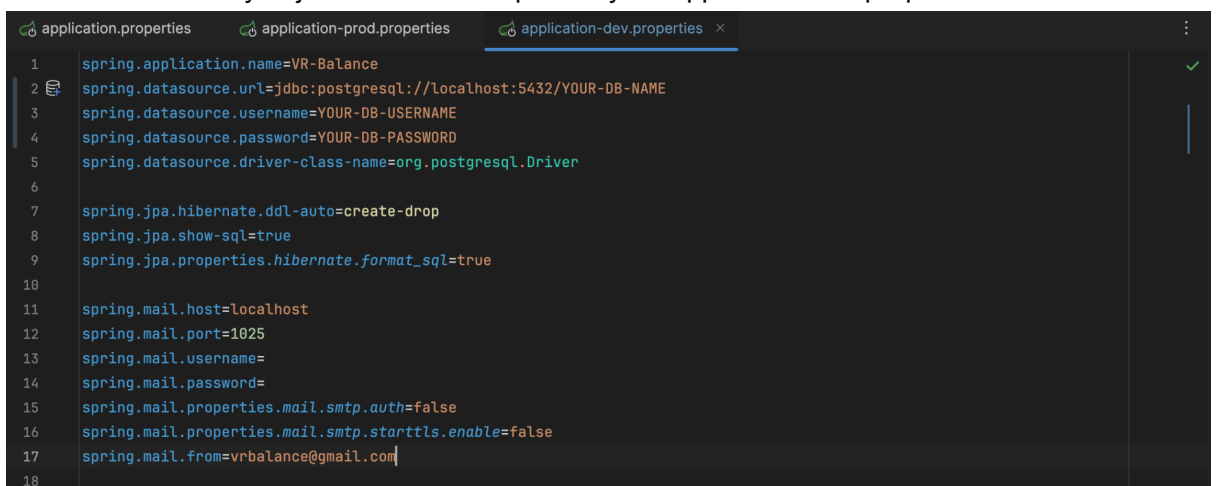


13. Now that you have Docker installed, you can also use it to install PostgreSQL.

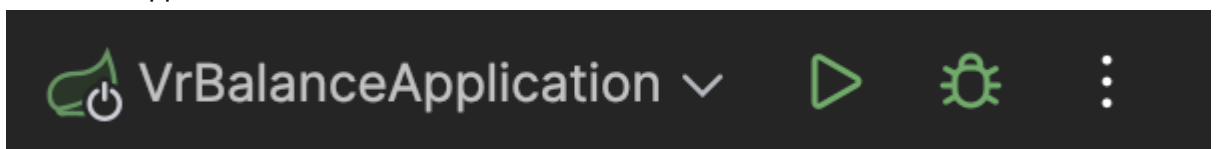
14. Run the following command, replacing `your_password`, `your_user`, and `your_database` with your own values.

```
docker run --name postgres-container -e POSTGRES_PASSWORD=your_password
-e POSTGRES_USER=your_user -e POSTGRES_DB=your_database -p 5432:5432
-d postgres:latest
```

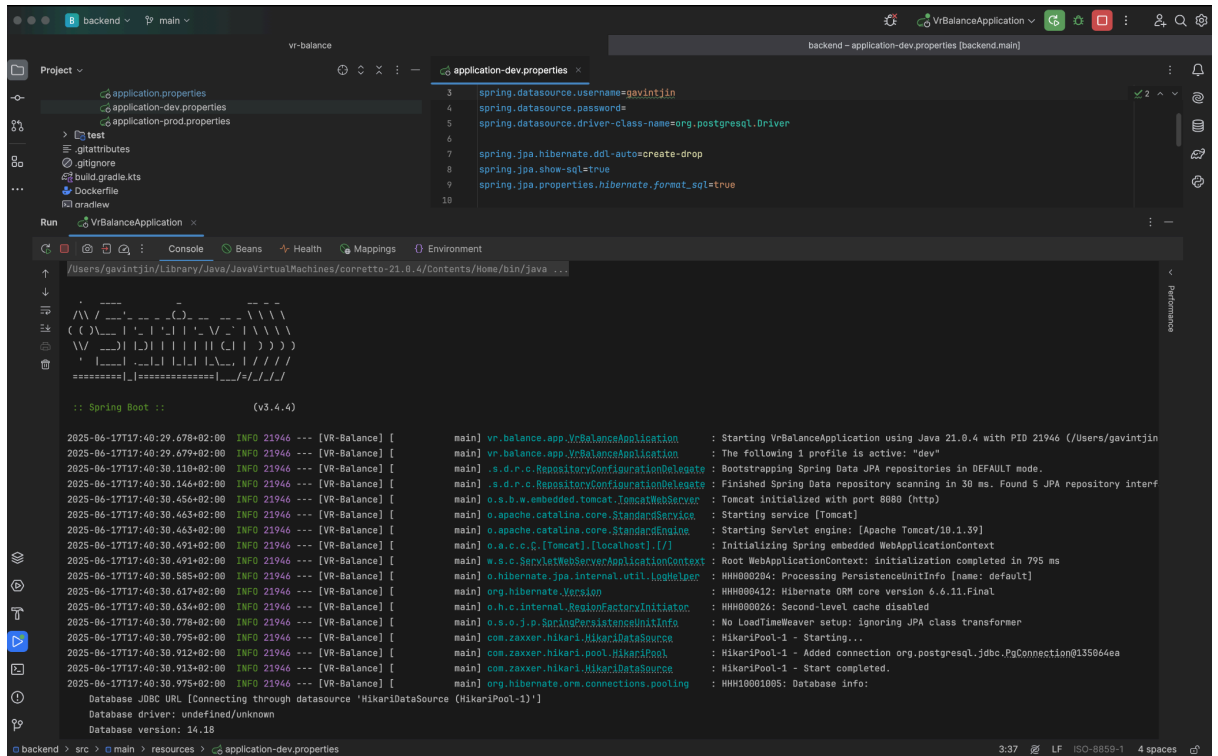
15. Add the values you just defined in step 14 to your `application-dev.properties` file.



16. Now that everything is installed, you can start the project by clicking the VrBalanceApplication button.

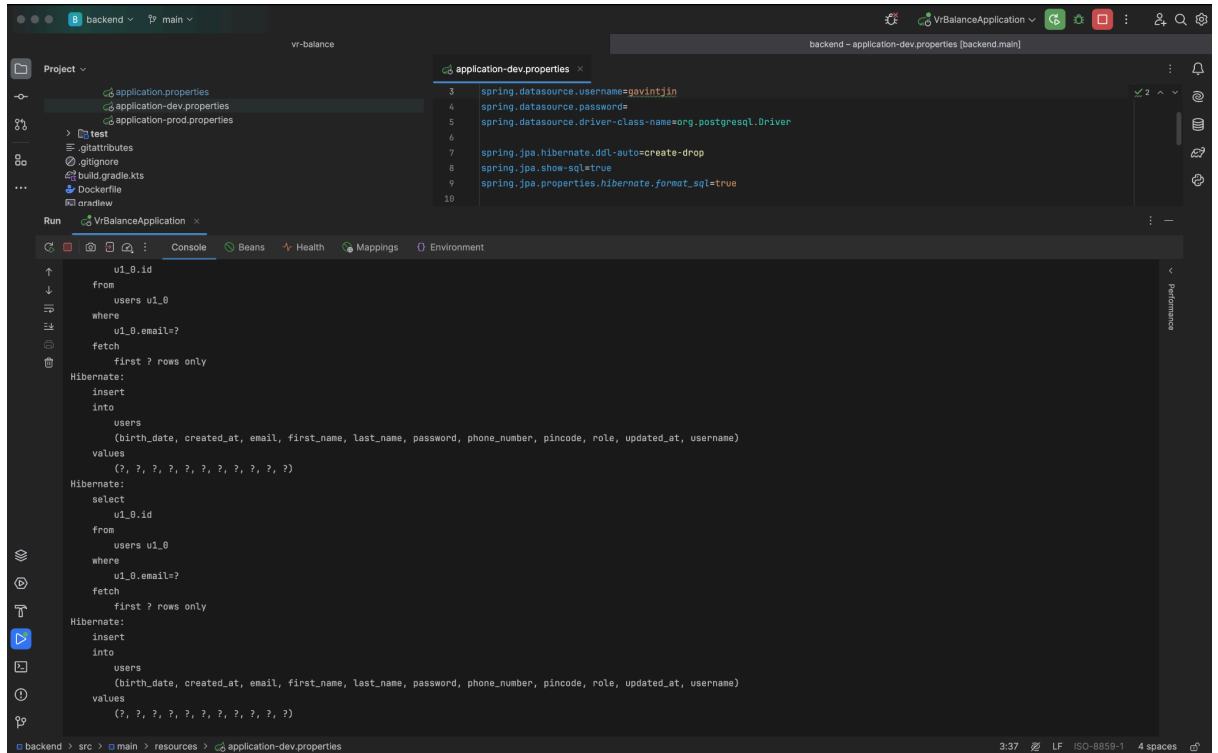


17. When the apps start it looks like this



The screenshot shows an IDE with the 'Run' tab selected, displaying the startup logs of a Spring Boot application. The logs are timestamped and show the application's progress from booting to starting the web server. The application is named 'VrBalanceApplication' and is running on a Java 11.0.4 virtual machine. The logs indicate that the application is using a 'dev' profile and is bootstrapping Spring Data JPA repositories. The application is also using a HikariPool-1 for database connections. The logs show the application is starting the web server on port 8080 (http) and is initializing the Spring embedded WebApplicationContext. The application is also starting the HikariPool-1 and is adding a connection to the database. The logs show the application is starting the HikariPool-1 and is adding a connection to the database. The logs show the application is starting the HikariPool-1 and is adding a connection to the database.

```
2025-06-17T17:40:29.678+02:00 INFO 21946 --- [Vr-Balance] [main] vr.balance.app.VrBalanceApplication : Starting VrBalanceApplication using Java 21.0.4 with PID 21946 (/Users/gavintjin/Library/Java/JavaVirtualMachines/corretto-21.0.4/Contents/Home/bin/java ...)
2025-06-17T17:40:29.679+02:00 INFO 21946 --- [Vr-Balance] [main] vr.balance.app.VrBalanceApplication : The following 1 profile is active: "dev"
2025-06-17T17:40:30.116+02:00 INFO 21946 --- [Vr-Balance] [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2025-06-17T17:40:30.146+02:00 INFO 21946 --- [Vr-Balance] [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 30 ms. Found 3 JPA repository interfaces
2025-06-17T17:40:30.456+02:00 INFO 21946 --- [Vr-Balance] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-06-17T17:40:30.463+02:00 INFO 21946 --- [Vr-Balance] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-06-17T17:40:30.491+02:00 INFO 21946 --- [Vr-Balance] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.39]
2025-06-17T17:40:30.491+02:00 INFO 21946 --- [Vr-Balance] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-06-17T17:40:30.585+02:00 INFO 21946 --- [Vr-Balance] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 795 ms
2025-06-17T17:40:30.617+02:00 INFO 21946 --- [Vr-Balance] [main] org.hibernate.Version : HHH000284: Processing PersistenceUnitInfo [name: default]
2025-06-17T17:40:30.634+02:00 INFO 21946 --- [Vr-Balance] [main] org.h.c.internal.RegionFactoryInitiator : HHH00026: Second-level cache disabled
2025-06-17T17:40:30.778+02:00 INFO 21946 --- [Vr-Balance] [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer
2025-06-17T17:40:30.795+02:00 INFO 21946 --- [Vr-Balance] [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-06-17T17:40:30.912+02:00 INFO 21946 --- [Vr-Balance] [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection org.postgresql.jdbc.PgConnection@135864ea
2025-06-17T17:40:30.913+02:00 INFO 21946 --- [Vr-Balance] [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-06-17T17:40:30.975+02:00 INFO 21946 --- [Vr-Balance] [main] org.hibernate.orm.connections.pooling : HHH10001895: Database info:
Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
Database driver: undefined/unknown
Database version: 14.18
```



The screenshot shows an IDE with the 'Run' tab selected, displaying the SQL logs of a Spring Boot application. The logs show the application's progress from booting to starting the web server. The application is named 'VrBalanceApplication' and is running on a Java 11.0.4 virtual machine. The logs indicate that the application is using a 'dev' profile and is bootstrapping Spring Data JPA repositories. The application is also using a HikariPool-1 for database connections. The logs show the application is starting the web server on port 8080 (http) and is initializing the Spring embedded WebApplicationContext. The application is also starting the HikariPool-1 and is adding a connection to the database. The logs show the application is starting the HikariPool-1 and is adding a connection to the database. The logs show the application is starting the HikariPool-1 and is adding a connection to the database.

```
from
  users u1_0
where
  u1_0.email=?
fetch
  first ? rows only
Hibernate:
  insert
  into
  users
  (birth_date, created_at, email, first_name, last_name, password, phone_number, pincode, role, updated_at, username)
  values
  (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
Hibernate:
  select
  u1_0.id
  from
  users u1_0
where
  u1_0.email=?
fetch
  first ? rows only
Hibernate:
  insert
  into
  users
  (birth_date, created_at, email, first_name, last_name, password, phone_number, pincode, role, updated_at, username)
  values
  (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
```

18. Go to: <http://localhost:8080/swagger-ui/index.html#/> to confirm that the API has started correctly. You should see this page:

