Risk Engine POC

Setup Guide for Beginners

Contents

1 Introduction **3**

2 Environment Requirements **3**
2.1 Required Versions . . . 3

3 Verify Your Environment **3**
3.1 Check Java . . . 3
3.2 Check Maven (Optional) . . . 3
3.3 Check PostgreSQL . . . 3

4 What This Project Does **4**

5 Project Structure Overview **4**
5.1 Backend Folder . . . 4

6 Database Setup **4**
6.1 Create Database . . . 4
6.2 Tables . . . 5

7 Configure Application **5**

8 Run the Application **5**
8.1 Command Line . . . 5
8.2 Using IntelliJ . . . 5

9 Post Setup Checks **6**
9.1 Check Application Started . . . 6
9.2 Open Browser . . . 6
9.3 Test Login . . . 6
9.4 Test Risk Evaluation . . . 6
9.5 Verify Database Storage . . . 6

10 Common Errors **6**
10.1 Database Connection Error . . . 6
10.2 Port 8080 Already in Use . . . 7
10.3 Java Version Error . . . 7

11 Quick Summary **7**

1 Introduction

This guide explains how to set up and run the **Risk Engine POC** project from scratch.

It is written for beginners. Follow each step in order. Do not skip environment verification.

2 Environment Requirements

2.1 Required Versions

Component	Required Version	Defined In
Java	21	pom.xml
Spring Boot	3.2.0	pom.xml
PostgreSQL	12+ (14+ recommended)	Installed manually
Maven	3.6+ (optional)	Use mvnw wrapper

3 Verify Your Environment

Run these commands before proceeding.

3.1 Check Java

```
java -version
```

Expected output must show **version 21**.
If not:

- Install Java 21
- Set JAVA_HOME correctly

3.2 Check Maven (Optional)

```
mvn -version
```

Must show:

- Maven 3.6+
- Java version: 21

3.3 Check PostgreSQL

```
psql --version
```

Must show version 12 or higher.

4 What This Project Does

This project simulates a post-login runtime risk detection system.

Flow:

1. User logs in
2. Browser collects security signals
3. Signals sent to backend
4. Backend calculates risk score
5. Decision returned:
   - ALLOW
   - MFA
   - TERMINATE
6. Data stored in PostgreSQL

This is a proof-of-concept only.

5 Project Structure Overview

5.1 Backend Folder

- **pom.xml** – Maven configuration
- **RiskEngineApplication.java** – Main application entry
- **controller/** – REST APIs
- **service/** – Business logic
- **entity/** – Database models
- **repository/** – Database access layer
- **application.properties** – Configuration file
- **static/** – Frontend HTML + JavaScript

6 Database Setup

6.1 Create Database

Connect using psql:

```
psql -U postgres
```

Then run:

```
CREATE DATABASE risk_engine;
```

Exit using:

```
\q
```

6.2 Tables

Tables are automatically created if this property exists:

```
spring.jpa.hibernate.ddl-auto=update
```

Tables:

- raw_signals
- risk_decisions

7 Configure Application

Open:
`backend/src/main/resources/application.properties`
Update:

```
spring.datasource.username=postgres
spring.datasource.password=your_password
```

If database runs on different host:

```
spring.datasource.url=jdbc:postgresql://localhost:5432/risk_engine
```

8 Run the Application

8.1 Command Line

Navigate to backend folder:

```
cd backend
```

Run:

```
./mvnw spring-boot:run
```

Windows:

```
mvnw.cmd spring-boot:run
```

8.2 Using IntelliJ

1. Open backend folder
2. Locate RiskEngineApplication.java
3. Click Run

9 Post Setup Checks

9.1 Check Application Started

Console must show:

```
Started RiskEngineApplication
```

9.2 Open Browser

Visit:

```
http://localhost:8080
```

You should see login page.

9.3 Test Login

Enter any username/password.
You should be redirected to dashboard.

9.4 Test Risk Evaluation

Click **Evaluate Risk**.
Expected:

- Risk score displayed
- Decision displayed
- Device signature displayed

9.5 Verify Database Storage

Run:

```
psql -U postgres -d risk_engine
```

Then:

```
SELECT * FROM risk_decisions;
```

You should see records.

10 Common Errors

10.1 Database Connection Error

Cause:

- PostgreSQL not running
- Wrong username/password
- Database not created

10.2 Port 8080 Already in Use

Change:

```
server.port=8081
```

10.3 Java Version Error

Ensure:

```
java -version
```

Shows version 21.

11 Quick Summary

- Install Java 21
- Install PostgreSQL
- Create database: risk_engine
- Update application.properties
- Run backend
- Open http://localhost:8080

This completes the setup of the Risk Engine POC.