Payment Service – Event Ticketing System

Overview

This documentation describes the design, development, and deployment of the Payment Service, which is one of the key microservices in the Event Ticketing and Seat Reservation System (ETS). The Payment Service manages financial transactions, including processing user charges, issuing refunds, and maintaining idempotent request handling to prevent duplicate operations.
It operates as an independent Spring Boot microservice following the database-per-service pattern for full data isolation and scalability.

Service Responsibilities

The Payment Service is responsible for:

- Processing new payment charges from confirmed orders.
- Recording and updating transaction statuses – PENDING, SUCCESS, FAILED, REFUNDED.
- Handling refund requests for successful payments.
- Enforcing idempotency using an Idempotency-Key to prevent duplicate charges.
- Providing RESTful APIs for order and finance modules to initiate or verify payments.
- Persisting all transactions in a dedicated PostgreSQL database.

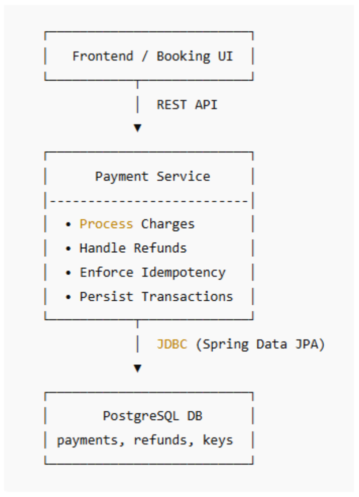
Each payment record is uniquely linked to an Order ID from the Order Service.

Architecture and Design

The Payment Service follows a four-layered architecture:

- Controller Layer → Exposes REST endpoints to clients and other services.
- Service Layer → Implements business logic for charge processing, refunds, and idempotency.
- Repository Layer → Handles database persistence via Spring Data JPA.
- Model Layer → Defines entity classes for Payment, Refund, and Idempotency Key.

Architecture Diagram-



Entity Relationship Diagram (ERD)-

Entities:

1. Payment
   - Tracks each charge request.
   - Linked to an orderId.
   - Has status, amount, method, and reference.
2. Refund
   - Linked to a paymentId.
   - Tracks refund amount and reference number.
3. IdempotencyKey
   - Stores request fingerprints and cached responses to ensure idempotent POST operations.

Relationships:

- One Payment can have multiple Refunds.
- Each IdempotencyKey is unique per API call.

REST API Endpoints -

Method	Endpoint	Description
POST	/v1/payments/charge	Process a payment charge (requires Idempotency-Key header)
POST	/v1/payments/refund	Refund a completed payment
GET	/v1/payments/{id}	Retrieve details of a specific payment
GET	/v1/payments/health	Health check endpoint
GET	/actuator/prometheus	Prometheus metrics for monitoring

Containerization with Docker –

The Payment Service is fully containerized using Docker.

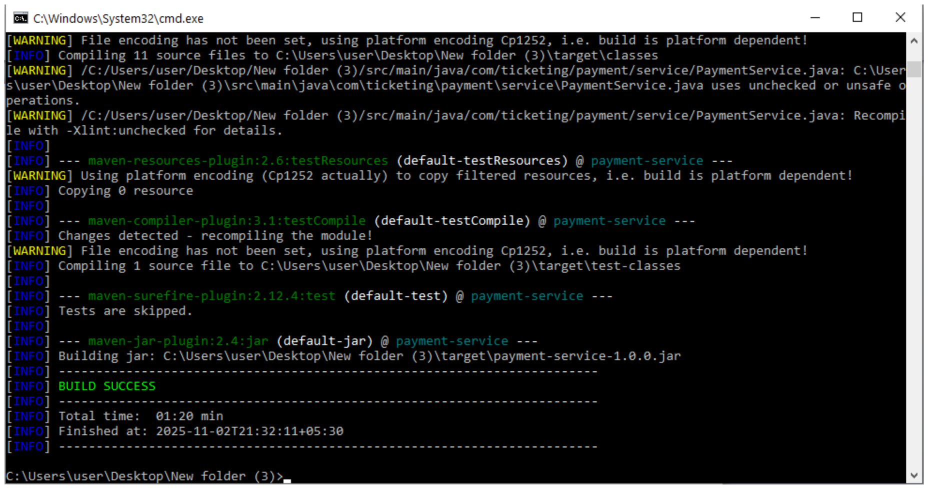
- Dockerfile: Builds a lightweight image from eclipse-temurin:17-jdk-alpine.
- docker-compose.yml: Deploys both the application and a dedicated PostgreSQL database.

Build and Run Commands:

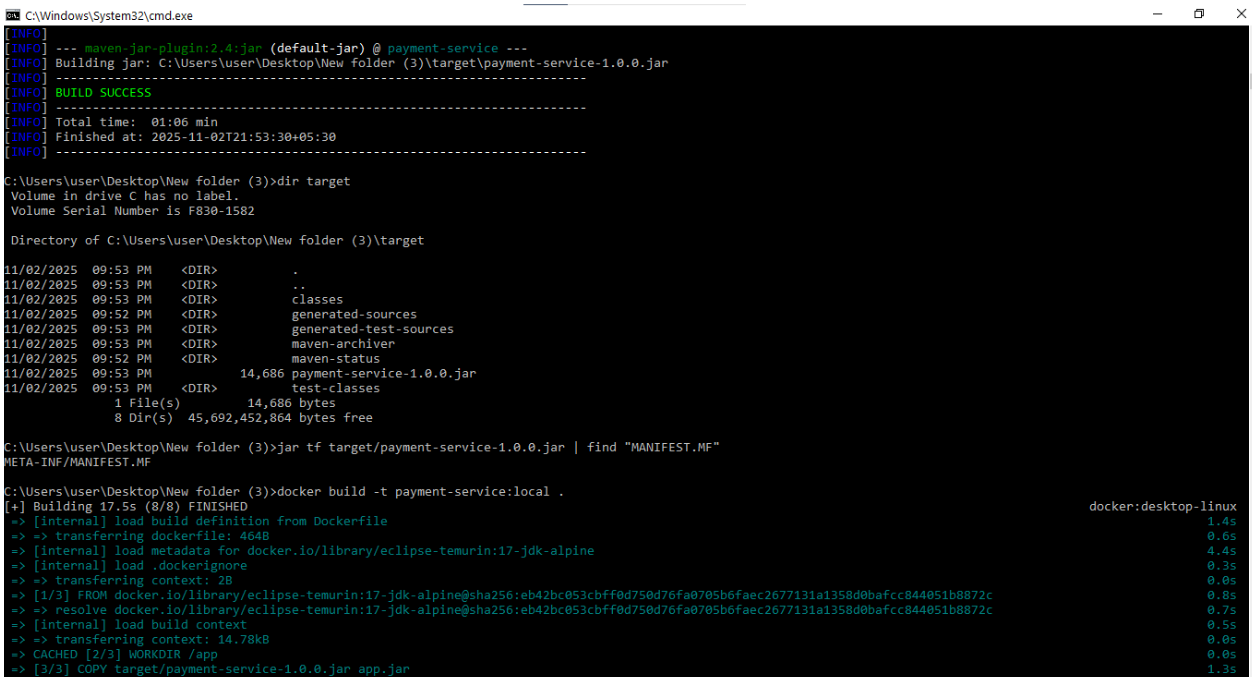
```
mvn clean package -DskipTests
docker-compose up --build
```

Once running:

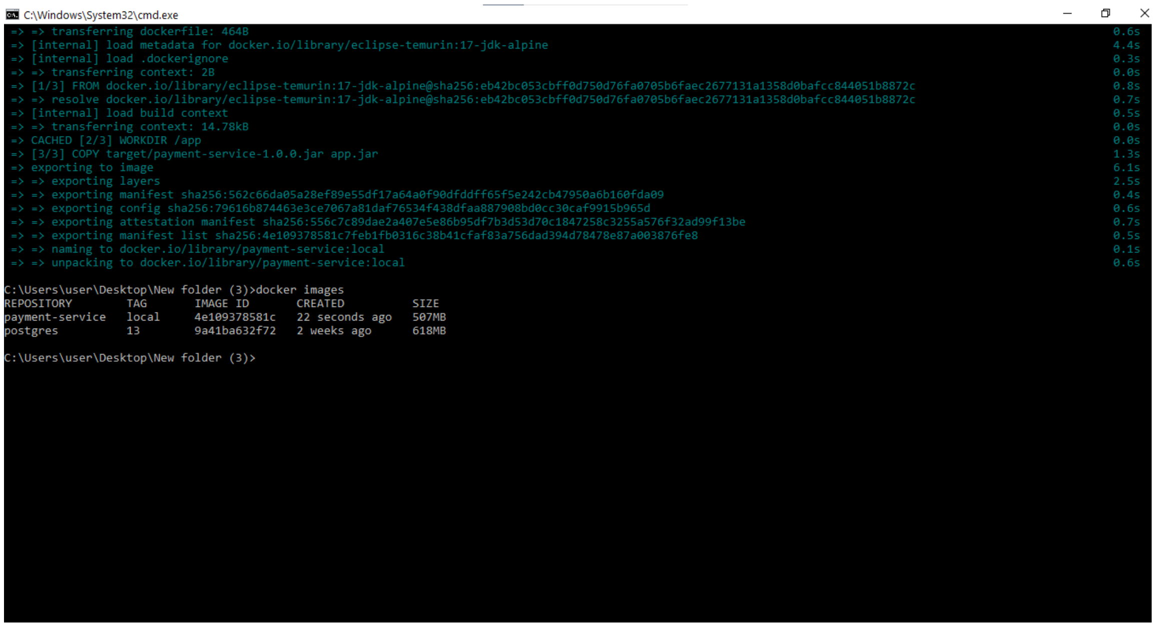
- Application: http://localhost:8080/v1/payments/health
- Database: localhost:5432 (paymentsdb)


C:\Windows\System32\cmd.exe

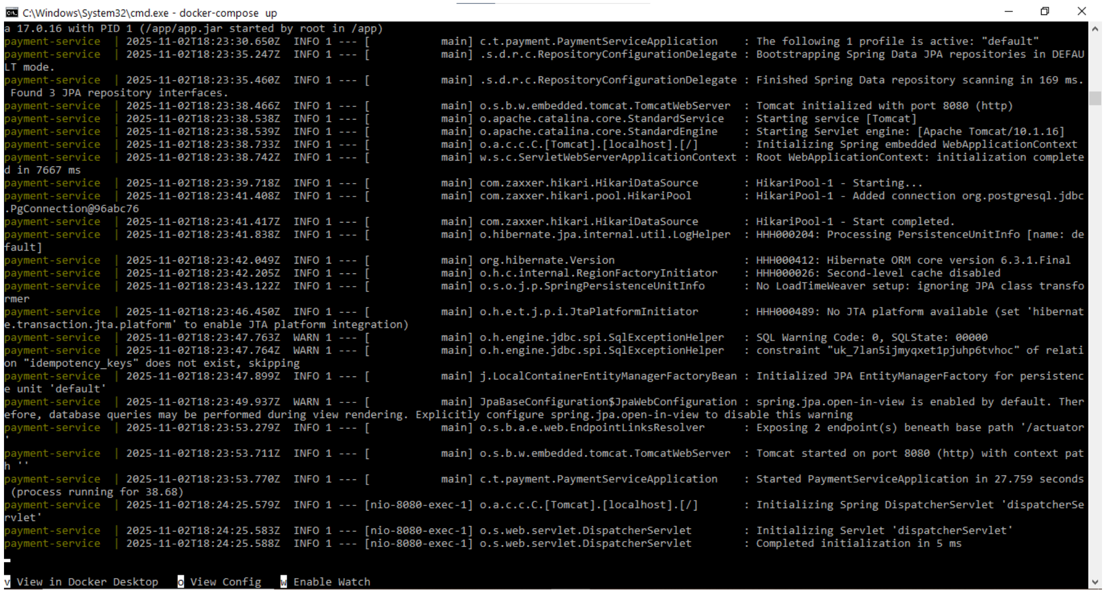
```
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 11 source files to C:\Users\user\Desktop\New folder (3)\target\classes
[WARNING] /C:/Users/user/Desktop/New folder (3)/src/main/java/com/ticketing/payment/service/PaymentService.java: C:\User
s\user\Desktop\New folder (3)\src\main\java\com\ticketing\payment\service\PaymentService.java uses unchecked or unsafe o
perations.
[WARNING] /C:/Users/user/Desktop/New folder (3)/src/main/java/com/ticketing/payment/service/PaymentService.java: Recompi
le with -Xlint:unchecked for details.
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ payment-service ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ payment-service ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to C:\Users\user\Desktop\New folder (3)\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ payment-service ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ payment-service ---
[INFO] Building jar: C:\Users\user\Desktop\New folder (3)\target\payment-service-1.0.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:20 min
[INFO] Finished at: 2025-11-02T21:32:11+05:30
[INFO] ------------------------------------------------------------------------

C:\Users\user\Desktop\New folder (3)>
```


C:\Windows\System32\cmd.exe

```
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ payment-service ---
[INFO] Building jar: C:\Users\user\Desktop\New folder (3)\target\payment-service-1.0.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:06 min
[INFO] Finished at: 2025-11-02T21:53:30+05:30
[INFO] ------------------------------------------------------------------------

C:\Users\user\Desktop\New folder (3)>dir target
 Volume in drive C has no label.
 Volume Serial Number is F830-1582

 Directory of C:\Users\user\Desktop\New folder (3)\target

11/02/2025  09:53 PM    <DIR>          .
11/02/2025  09:53 PM    <DIR>          ..
11/02/2025  09:53 PM    <DIR>          classes
11/02/2025  09:52 PM    <DIR>          generated-sources
11/02/2025  09:53 PM    <DIR>          generated-test-sources
11/02/2025  09:53 PM    <DIR>          maven-archiver
11/02/2025  09:52 PM    <DIR>          maven-status
11/02/2025  09:53 PM            14,686 payment-service-1.0.0.jar
11/02/2025  09:53 PM    <DIR>          test-classes
               1 File(s)         14,686 bytes
               8 Dir(s)  45,692,452,864 bytes free

C:\Users\user\Desktop\New folder (3)>jar tf target/payment-service-1.0.0.jar | find "MANIFEST.MF"
META-INF/MANIFEST.MF

C:\Users\user\Desktop\New folder (3)>docker build -t payment-service:local .
[+] Building 17.5s (8/8) FINISHED                                                                        docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                     1.4s
 => => transferring dockerfile: 464B                                                                                     0.6s
 => [internal] load metadata for docker.io/library/eclipse-temurin:17-jdk-alpine                                         4.4s
 => [internal] load .dockerignore                                                                                        0.3s
 => => transferring context: 2B                                                                                          0.0s
 => [1/3] FROM docker.io/library/eclipse-temurin:17-jdk-alpine@sha256:eb42bc053cbff0d750d76fa0705b6faec2677131a1358d0bafcc844051b8872c  0.8s
 => => resolve docker.io/library/eclipse-temurin:17-jdk-alpine@sha256:eb42bc053cbff0d750d76fa0705b6faec2677131a1358d0bafcc844051b8872c  0.7s
 => [internal] load build context                                                                                        0.5s
 => => transferring context: 14.78kB                                                                                     0.0s
 => CACHED [2/3] WORKDIR /app                                                                                            0.0s
 => [3/3] COPY target/payment-service-1.0.0.jar app.jar                                                                  1.3s
```

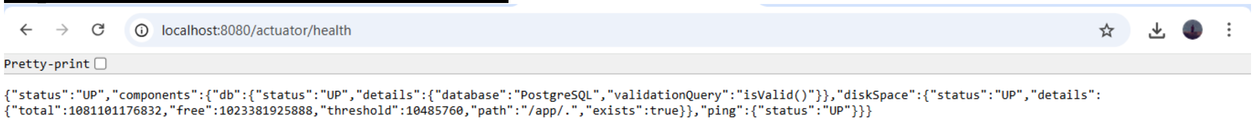


```
=> => transferring dockerfile: 464B                                                                          0.6s
=> [internal] load metadata for docker.io/library/eclipse-temurin:17-jdk-alpine                              4.4s
=> [internal] load .dockerignore                                                                             0.3s
=> => transferring context: 2B                                                                               0.0s
=> [1/3] FROM docker.io/library/eclipse-temurin:17-jdk-alpine@sha256:eb42bc053cbff0d750d76fa0705b6faec2677131a1358d0bafcc844051b8872c  0.8s
=> => resolve docker.io/library/eclipse-temurin:17-jdk-alpine@sha256:eb42bc053cbff0d750d76fa0705b6faec2677131a1358d0bafcc844051b8872c  0.7s
=> [internal] load build context                                                                             0.5s
=> => transferring context: 14.78kB                                                                          0.0s
=> CACHED [2/3] WORKDIR /app                                                                                 0.0s
=> [3/3] COPY target/payment-service-1.0.0.jar app.jar                                                       1.3s
=> exporting to image                                                                                        6.1s
=> => exporting layers                                                                                       2.5s
=> => exporting manifest sha256:562c66da05a28ef89e55df17a64a0f90dfddff65f5e242cb47950a6b160fda09             0.4s
=> => exporting config sha256:79616b874463e3ce7067a81daf76534f438dfaa887908bd0cc30caf9915b965d               0.6s
=> => exporting attestation manifest sha256:556c7c89dae2a407e5e86b95df7b3d53d70c1847258c3255a576f32ad99f13be 0.7s
=> => exporting manifest list sha256:4e109378581c7feb1fb0316c38b41cfaf83a756dad394d78478e87a003876fe8        0.5s
=> => naming to docker.io/library/payment-service:local                                                      0.1s
=> => unpacking to docker.io/library/payment-service:local                                                   0.6s

C:\Users\user\Desktop\New folder (3)>docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
payment-service   local     4e109378581c   22 seconds ago   507MB
postgres          13        9a41ba632f72   2 weeks ago      618MB

C:\Users\user\Desktop\New folder (3)>
```



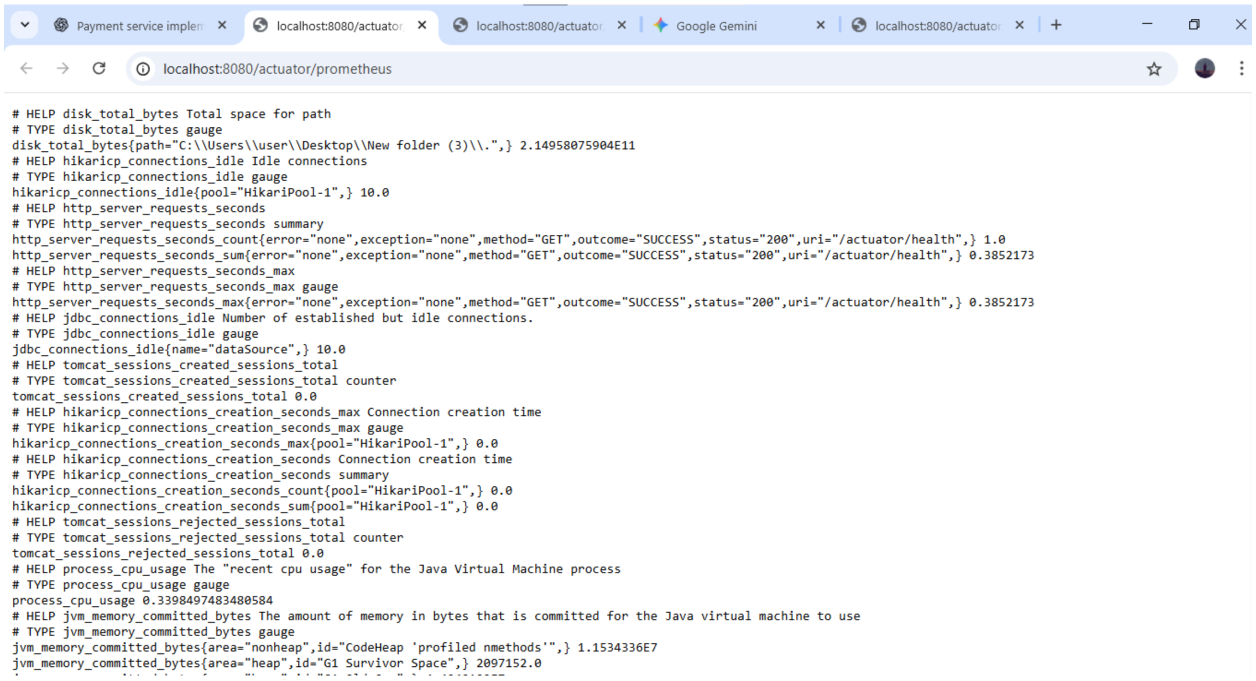
```
a 17.0.16 with PID 1 (/app/app.jar started by root in /app)
payment-service  | 2025-11-02T18:23:30.650Z  INFO 1 --- [           main] c.t.payment.PaymentServiceApplication    : The following 1 profile is active: "default"
payment-service  | 2025-11-02T18:23:35.247Z  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAU
LT mode.
payment-service  | 2025-11-02T18:23:35.460Z  INFO 1 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 169 ms.
 Found 3 JPA repository interfaces.
payment-service  | 2025-11-02T18:23:38.466Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
payment-service  | 2025-11-02T18:23:38.538Z  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
payment-service  | 2025-11-02T18:23:38.539Z  INFO 1 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.16]
payment-service  | 2025-11-02T18:23:38.733Z  INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
payment-service  | 2025-11-02T18:23:38.742Z  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization complete
d in 7667 ms
payment-service  | 2025-11-02T18:23:39.718Z  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
payment-service  | 2025-11-02T18:23:41.408Z  INFO 1 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection org.postgresql.jdbc
.PgConnection@96abc76
payment-service  | 2025-11-02T18:23:41.417Z  INFO 1 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
payment-service  | 2025-11-02T18:23:41.838Z  INFO 1 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: de
fault]
payment-service  | 2025-11-02T18:23:42.049Z  INFO 1 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.3.1.Final
payment-service  | 2025-11-02T18:23:42.205Z  INFO 1 --- [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Second-level cache disabled
payment-service  | 2025-11-02T18:23:43.122Z  INFO 1 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup: ignoring JPA class transfo
rmer
payment-service  | 2025-11-02T18:23:46.450Z  INFO 1 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA platform available (set 'hibernat
e.transaction.jta.platform' to enable JTA platform integration)
payment-service  | 2025-11-02T18:23:47.763Z  WARN 1 --- [           main] o.h.engine.jdbc.spi.SqlExceptionHelper   : SQL Warning Code: 0, SQLState: 00000
payment-service  | 2025-11-02T18:23:47.764Z  WARN 1 --- [           main] o.h.engine.jdbc.spi.SqlExceptionHelper   : constraint "uk_7lan5ijmyqxet1pjuhp6tvhoc" of relati
on "idempotency_keys" does not exist, skipping
payment-service  | 2025-11-02T18:23:47.899Z  INFO 1 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistenc
e unit 'default'
payment-service  | 2025-11-02T18:23:49.937Z  WARN 1 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Ther
efore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
payment-service  | 2025-11-02T18:23:53.279Z  INFO 1 --- [           main] o.s.b.a.e.web.EndpointLinksResolver      : Exposing 2 endpoint(s) beneath base path '/actuator
'
payment-service  | 2025-11-02T18:23:53.711Z  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context pat
h ''
payment-service  | 2025-11-02T18:23:53.770Z  INFO 1 --- [           main] c.t.payment.PaymentServiceApplication    : Started PaymentServiceApplication in 27.759 seconds
 (process running for 38.68)
payment-service  | 2025-11-02T18:24:25.579Z  INFO 1 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherSe
rvlet'
payment-service  | 2025-11-02T18:24:25.583Z  INFO 1 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
payment-service  | 2025-11-02T18:24:25.588Z  INFO 1 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 5 ms

v View in Docker Desktop   o View Config   w Enable Watch
```

<u>http://localhost:8080/actuator/health -</u>



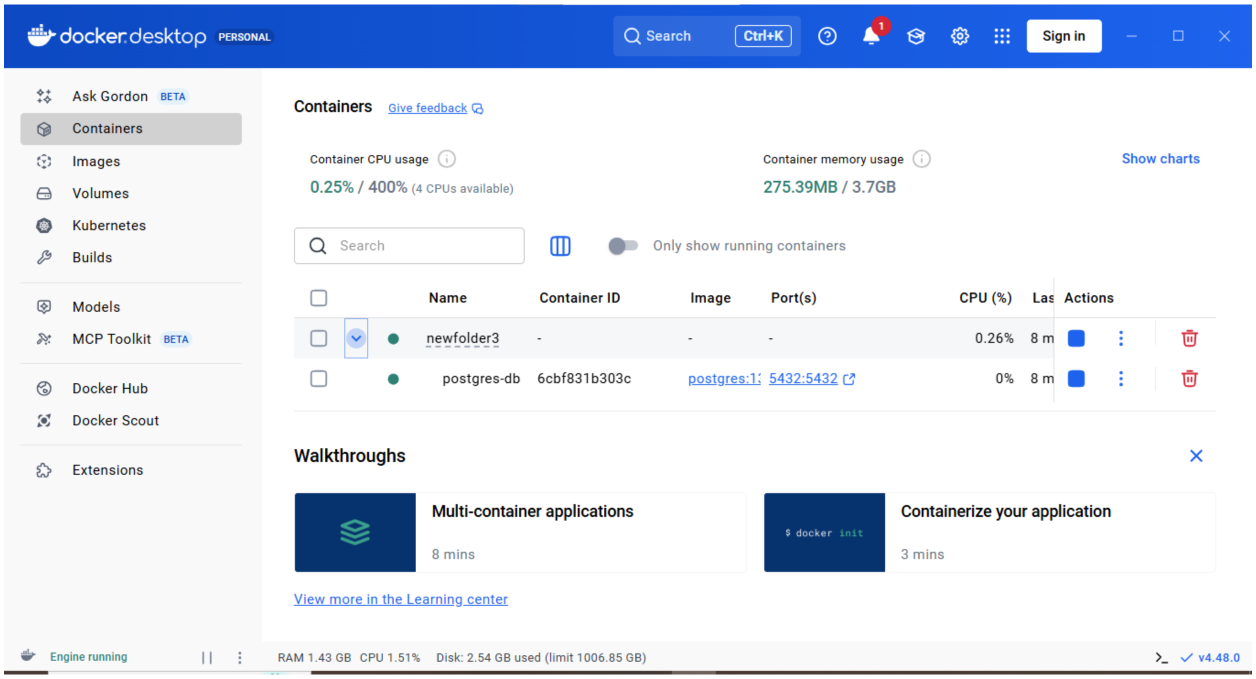
```
{"status":"UP","components":{"db":{"status":"UP","details":{"database":"PostgreSQL","validationQuery":"isValid()"}},"diskSpace":{"status":"UP","details":
{"total":1081101176832,"free":1023381925888,"threshold":10485760,"path":"/app/.","exists":true}},"ping":{"status":"UP"}}}
```

***<u>http://localhost:8080/actuator/prometheus</u>* -**



```
# HELP disk_total_bytes Total space for path
# TYPE disk_total_bytes gauge
disk_total_bytes{path="C:\\Users\\user\\Desktop\\New folder (3)\\.",} 2.14958075904E11
# HELP hikaricp_connections_idle Idle connections
# TYPE hikaricp_connections_idle gauge
hikaricp_connections_idle{pool="HikariPool-1",} 10.0
# HELP http_server_requests_seconds
# TYPE http_server_requests_seconds summary
http_server_requests_seconds_count{error="none",exception="none",method="GET",outcome="SUCCESS",status="200",uri="/actuator/health",} 1.0
http_server_requests_seconds_sum{error="none",exception="none",method="GET",outcome="SUCCESS",status="200",uri="/actuator/health",} 0.3852173
# HELP http_server_requests_seconds_max
# TYPE http_server_requests_seconds_max gauge
http_server_requests_seconds_max{error="none",exception="none",method="GET",outcome="SUCCESS",status="200",uri="/actuator/health",} 0.3852173
# HELP jdbc_connections_idle Number of established but idle connections.
# TYPE jdbc_connections_idle gauge
jdbc_connections_idle{name="dataSource",} 10.0
# HELP tomcat_sessions_created_sessions_total
# TYPE tomcat_sessions_created_sessions_total counter
tomcat_sessions_created_sessions_total 0.0
# HELP hikaricp_connections_creation_seconds_max Connection creation time
# TYPE hikaricp_connections_creation_seconds_max gauge
hikaricp_connections_creation_seconds_max{pool="HikariPool-1",} 0.0
# HELP hikaricp_connections_creation_seconds Connection creation time
# TYPE hikaricp_connections_creation_seconds summary
hikaricp_connections_creation_seconds_count{pool="HikariPool-1",} 0.0
hikaricp_connections_creation_seconds_sum{pool="HikariPool-1",} 0.0
# HELP tomcat_sessions_rejected_sessions_total
# TYPE tomcat_sessions_rejected_sessions_total counter
tomcat_sessions_rejected_sessions_total 0.0
# HELP process_cpu_usage The "recent cpu usage" for the Java Virtual Machine process
# TYPE process_cpu_usage gauge
process_cpu_usage 0.3398497483480584
# HELP jvm_memory_committed_bytes The amount of memory in bytes that is committed for the Java virtual machine to use
# TYPE jvm_memory_committed_bytes gauge
jvm_memory_committed_bytes{area="nonheap",id="CodeHeap 'profiled nmethods'",} 1.1534336E7
jvm_memory_committed_bytes{area="heap",id="G1 Survivor Space",} 2097152.0
```

***<u>Docker</u>* -**



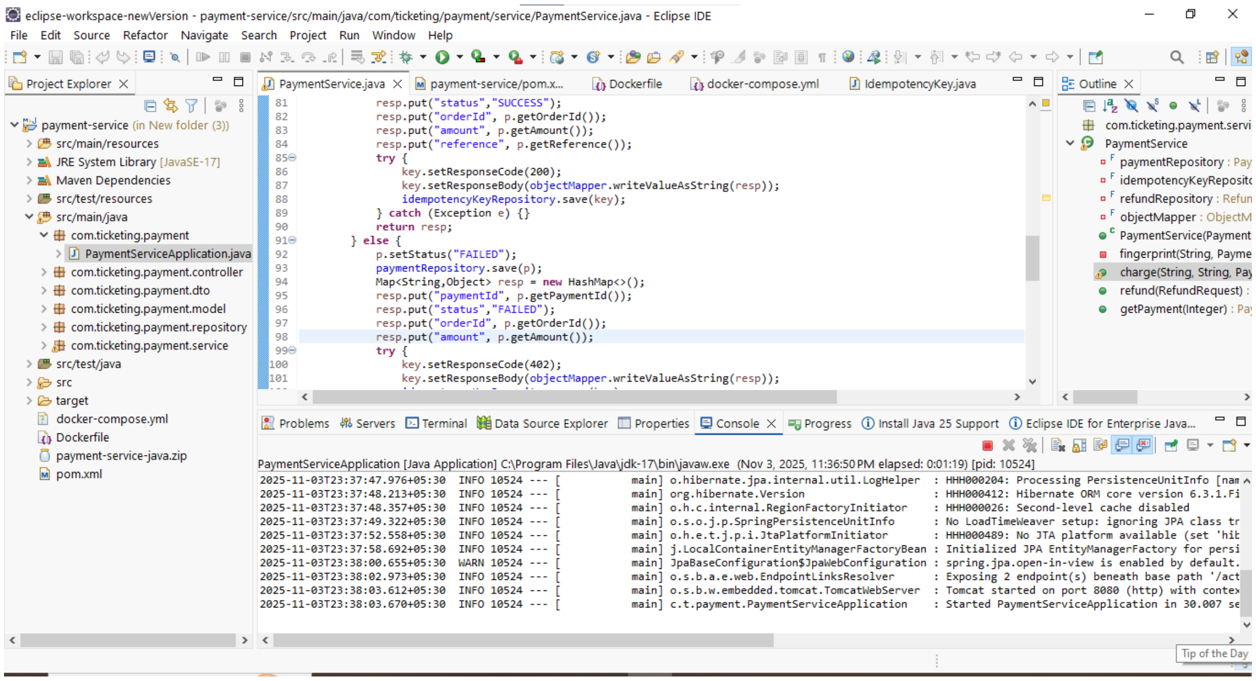
Containers

Container CPU usage
0.25% / 400% (4 CPUs available)

Container memory usage
275.39MB / 3.7GB

Name	Container ID	Image	Port(s)	CPU (%)	Las
newfolder3	-	-	-	0.26%	8 m
postgres-db	6cbf831b303c	postgres:1:	5432:5432	0%	8 m

RAM 1.43 GB CPU 1.51% Disk: 2.54 GB used (limit 1006.85 GB)


eclipse-workspace-newVersion - payment-service/src/main/java/com/ticketing/payment/service/PaymentService.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
payment-service (in New folder (3))
src/main/resources
JRE System Library [JavaSE-17]
Maven Dependencies
src/test/resources
src/main/java
com.ticketing.payment
PaymentServiceApplication.java
com.ticketing.payment.controller
com.ticketing.payment.dto
com.ticketing.payment.model
com.ticketing.payment.repository
com.ticketing.payment.service
src/test/java
src
target
docker-compose.yml
Dockerfile
payment-service-java.zip
pom.xml
PaymentService.java
payment-service/pom.x...
Dockerfile
docker-compose.yml
IdempotencyKey.java
Outline
com.ticketing.payment.servi
PaymentService
paymentRepository : Pay
idempotencyKeyReposito
refundRepository : Refun
objectMapper : ObjectM
PaymentService(Payment
fingerprint(String, Payme
charge(String, String, Pay
refund(RefundRequest) :
getPayment(Integer) : Pa
Problems Servers Terminal Data Source Explorer Properties Console Progress Install Java 25 Support Eclipse IDE for Enterprise Java...
Tip of the Day

```java
            resp.put("status","SUCCESS");
            resp.put("orderId", p.getOrderId());
            resp.put("amount", p.getAmount());
            resp.put("reference", p.getReference());
            try {
                key.setResponseCode(200);
                key.setResponseBody(objectMapper.writeValueAsString(resp));
                idempotencyKeyRepository.save(key);
            } catch (Exception e) {}
            return resp;
        } else {
            p.setStatus("FAILED");
            paymentRepository.save(p);
            Map<String,Object> resp = new HashMap<>();
            resp.put("paymentId", p.getPaymentId());
            resp.put("status","FAILED");
            resp.put("orderId", p.getOrderId());
            resp.put("amount", p.getAmount());
            try {
                key.setResponseCode(402);
                key.setResponseBody(objectMapper.writeValueAsString(resp));
```

```
PaymentServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 3, 2025, 11:36:50 PM elapsed: 0:01:19) [pid: 10524]
2025-11-03T23:37:47.976+05:30  INFO 10524 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [nam
2025-11-03T23:37:48.213+05:30  INFO 10524 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.3.1.Fi
2025-11-03T23:37:48.357+05:30  INFO 10524 --- [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Second-level cache disabled
2025-11-03T23:37:49.322+05:30  INFO 10524 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup: ignoring JPA class tr
2025-11-03T23:37:52.558+05:30  INFO 10524 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA platform available (set 'hib
2025-11-03T23:37:58.692+05:30  INFO 10524 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persi
2025-11-03T23:38:00.655+05:30  WARN 10524 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default.
2025-11-03T23:38:02.973+05:30  INFO 10524 --- [           main] o.s.b.a.e.web.EndpointLinksResolver      : Exposing 2 endpoint(s) beneath base path '/act
2025-11-03T23:38:03.612+05:30  INFO 10524 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with contex
2025-11-03T23:38:03.670+05:30  INFO 10524 --- [           main] c.t.payment.PaymentServiceApplication    : Started PaymentServiceApplication in 30.007 se
```

<u>Sample Request and Response -</u>

POST – Charge Payment

Request:

```
{
 "orderId": "ORD1001",
 "amount": 500.0,
 "currency": "INR",
 "method": "CARD"
}
```

Headers:
Idempotency-Key: charge-101
Response (Success):

```
{
 "paymentId": 1,
 "status": "SUCCESS",
 "orderId": "ORD1001",
 "amount": 500.0,
 "reference": "ETP-a1b2c3d4"
}
```

Response (Duplicate Key):

```
{
 "error": "Idempotency conflict"
}
```

POST – Refund Payment

Request:

```
{
 "paymentId": 1,
 "amount": 500.0
}
```

Response:

```
{
 "refundId": 1,
 "status": "SUCCESS",
 "paymentId": 1,
 "amount": 500.0
}
```

GET – Retrieve Payment

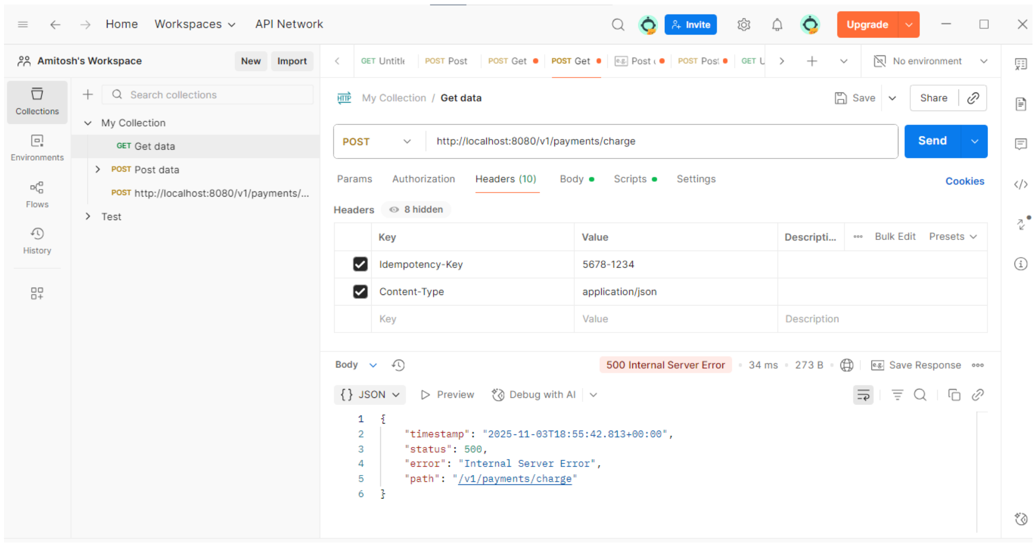
Request:

GET /v1/payments/1

Response:

```
{
 "paymentId": 1,
 "orderId": "ORD1001",
 "amount": 500.0,
 "currency": "INR",
 "status": "SUCCESS",
 "method": "CARD",
 "reference": "ETP-a1b2c3d4"
}
```

<u>Idempotency-</u>

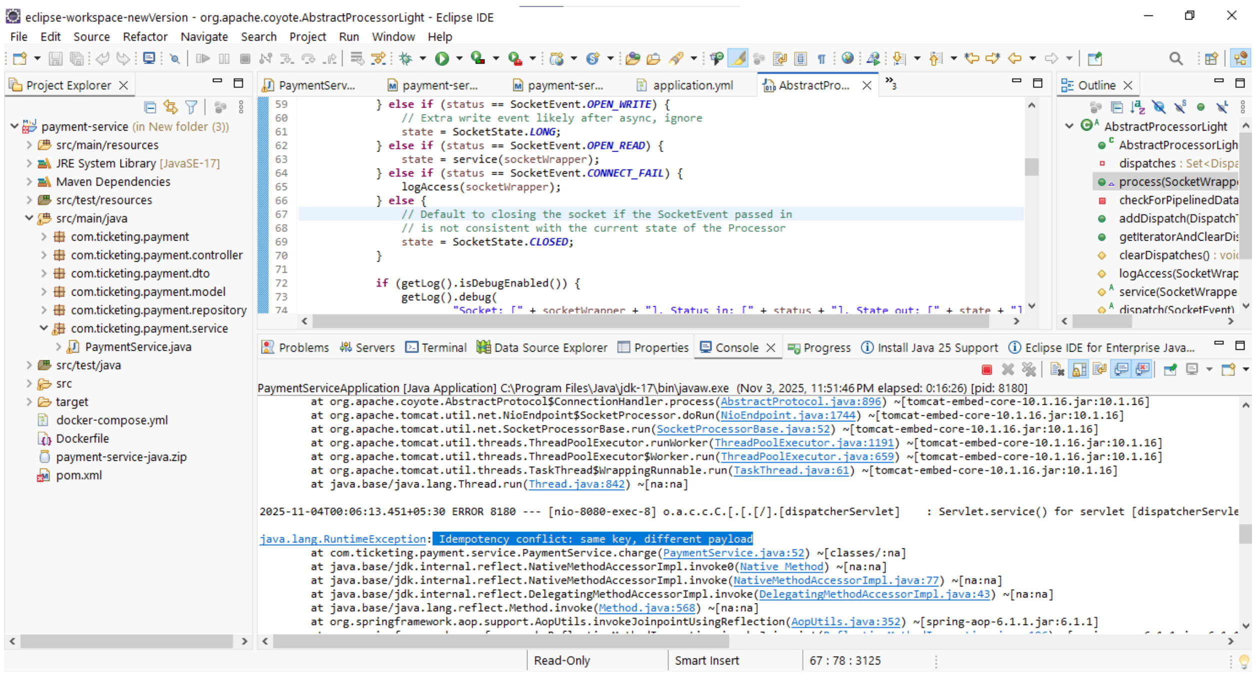


POST http://localhost:8080/v1/payments/charge

	Key	Value	Descripti...
✓	Idempotency-Key	5678-1234	
✓	Content-Type	application/json	
	Key	Value	Description

500 Internal Server Error · 34 ms · 273 B

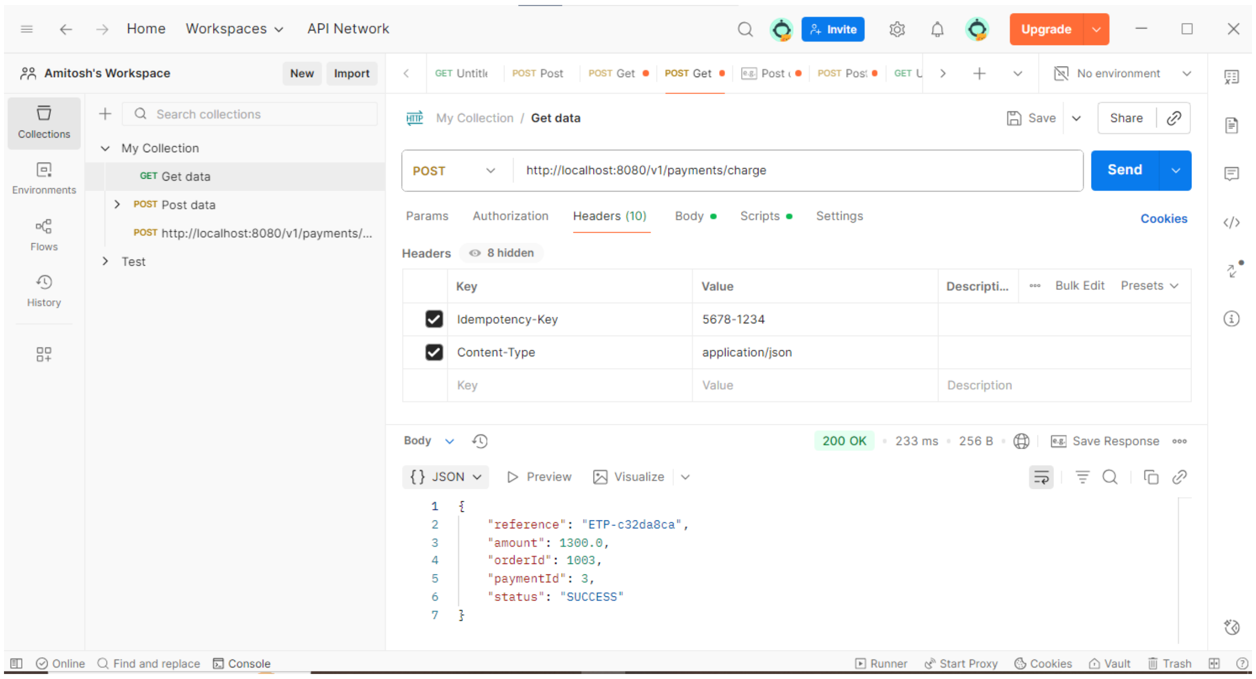
```
{
    "timestamp": "2025-11-03T18:55:42.813+00:00",
    "status": 500,
    "error": "Internal Server Error",
    "path": "/v1/payments/charge"
}
```



```
} else if (status == SocketEvent.OPEN_WRITE) {
    // Extra write event likely after async, ignore
    state = SocketState.LONG;
} else if (status == SocketEvent.OPEN_READ) {
    state = service(socketWrapper);
} else if (status == SocketEvent.CONNECT_FAIL) {
    logAccess(socketWrapper);
} else {
    // Default to closing the socket if the SocketEvent passed in
    // is not consistent with the current state of the Processor
    state = SocketState.CLOSED;
}

if (getLog().isDebugEnabled()) {
    getLog().debug(
            "Socket: [" + socketWrapper + "], Status in: [" + status + "], State out: [" + state + "]
```

```
PaymentServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 3, 2025, 11:51:46 PM elapsed: 0:16:26) [pid: 8180]
	at org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:896) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1744) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at java.base/java.lang.Thread.run(Thread.java:842) ~[na:na]

2025-11-04T00:06:13.451+05:30 ERROR 8180 --- [nio-8080-exec-8] o.a.c.c.C.[.[.[/].[dispatcherServlet]    : Servlet.service() for servlet [dispatcherServle

java.lang.RuntimeException: Idempotency conflict: same key, different payload
	at com.ticketing.payment.service.PaymentService.charge(PaymentService.java:52) ~[classes/:na]
	at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[na:na]
	at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77) ~[na:na]
	at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:na]
	at java.base/java.lang.reflect.Method.invoke(Method.java:568) ~[na:na]
	at org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.java:352) ~[spring-aop-6.1.1.jar:6.1.1]
```

http://localhost:8080/v1/payments/charge -



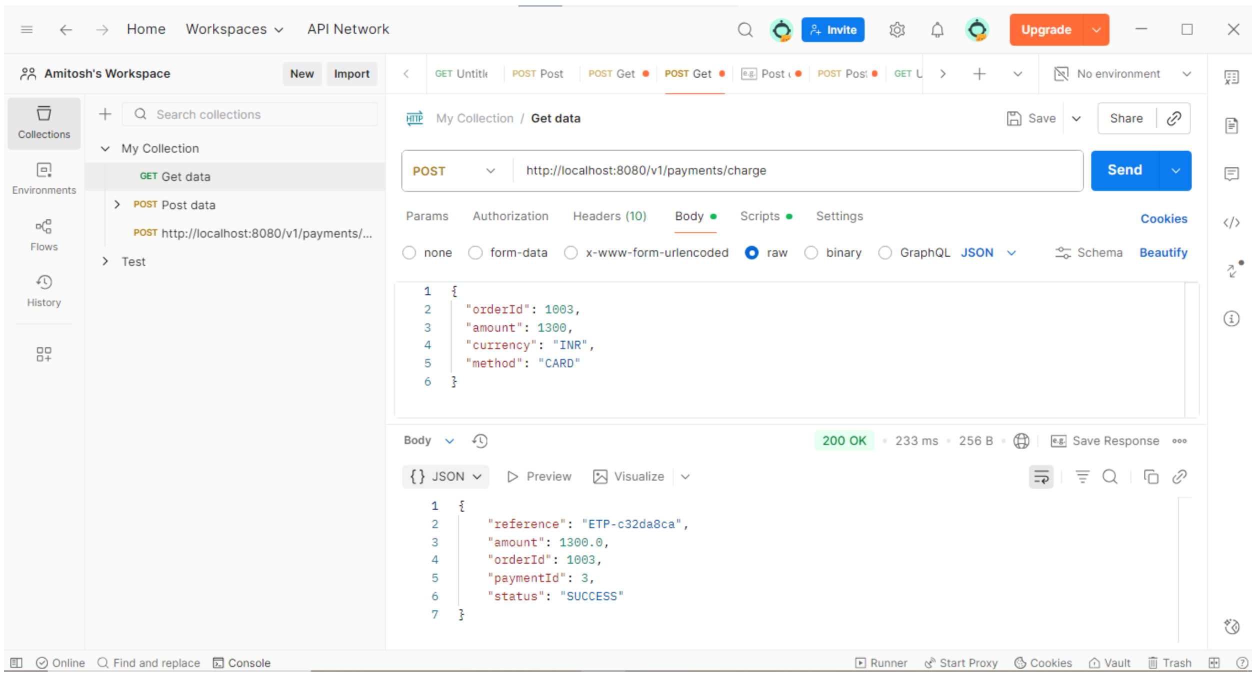
POST http://localhost:8080/v1/payments/charge

Headers (10)

	Key	Value	Description
✓	Idempotency-Key	5678-1234	
✓	Content-Type	application/json	
	Key	Value	Description

Body — 200 OK · 233 ms · 256 B

```
{
    "reference": "ETP-c32da8ca",
    "amount": 1300.0,
    "orderId": 1003,
    "paymentId": 3,
    "status": "SUCCESS"
}
```


Amitosh's Workspace

My Collection / Get data

POST http://localhost:8080/v1/payments/charge

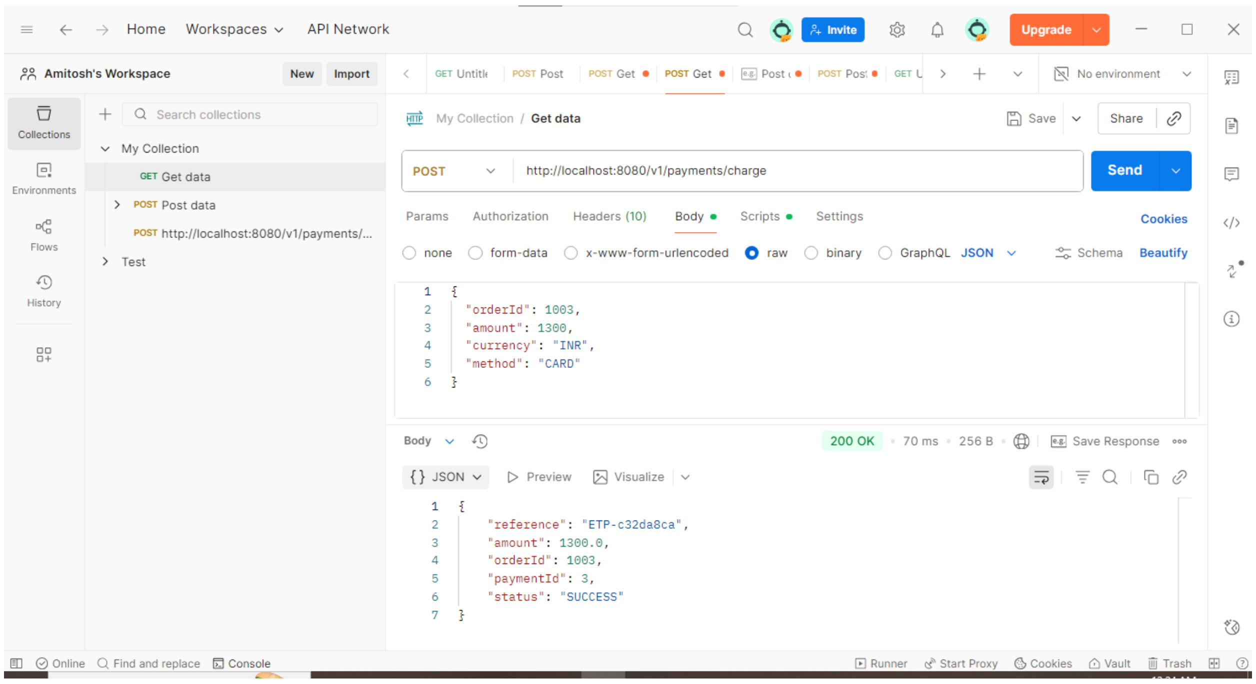
Body: raw JSON

```
{
  "orderId": 1003,
  "amount": 1300,
  "currency": "INR",
  "method": "CARD"
}
```

200 OK · 233 ms · 256 B

```
{
    "reference": "ETP-c32da8ca",
    "amount": 1300.0,
    "orderId": 1003,
    "paymentId": 3,
    "status": "SUCCESS"
}
```

__Retry and same result –__


Amitosh's Workspace

My Collection / Get data

POST http://localhost:8080/v1/payments/charge

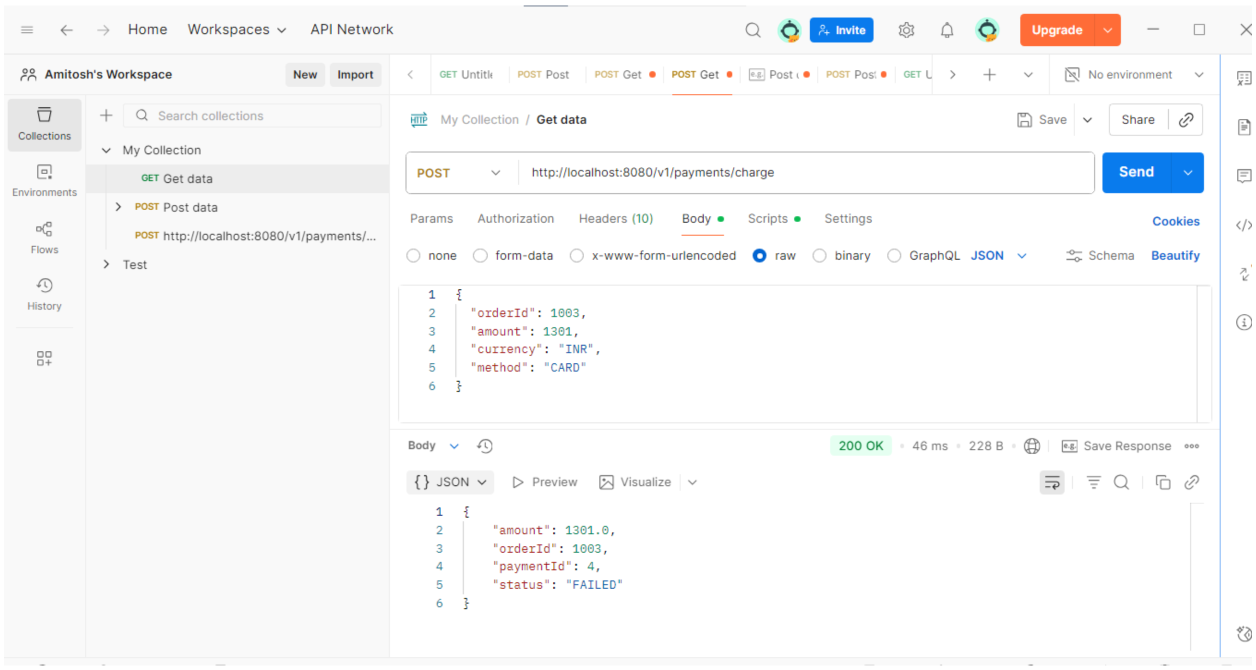
Body: raw JSON

```
{
  "orderId": 1003,
  "amount": 1300,
  "currency": "INR",
  "method": "CARD"
}
```

200 OK · 70 ms · 256 B

```
{
    "reference": "ETP-c32da8ca",
    "amount": 1300.0,
    "orderId": 1003,
    "paymentId": 3,
    "status": "SUCCESS"
}
```

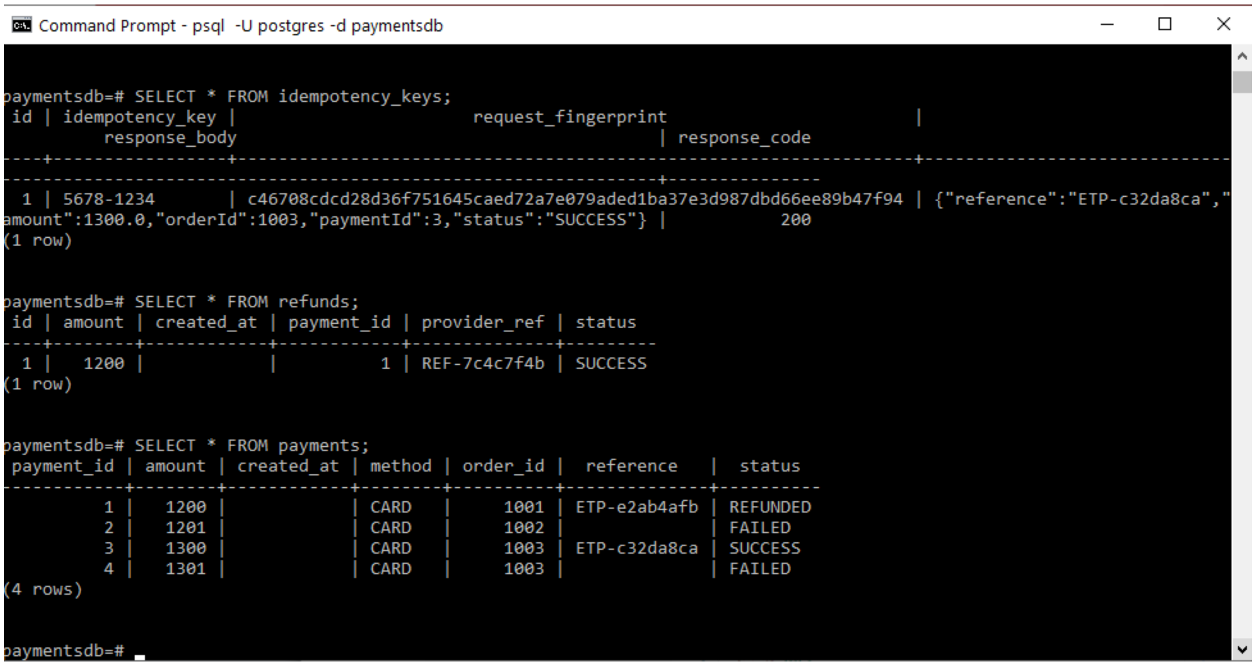
Failed scenario when amount odd –



POST http://localhost:8080/v1/payments/charge

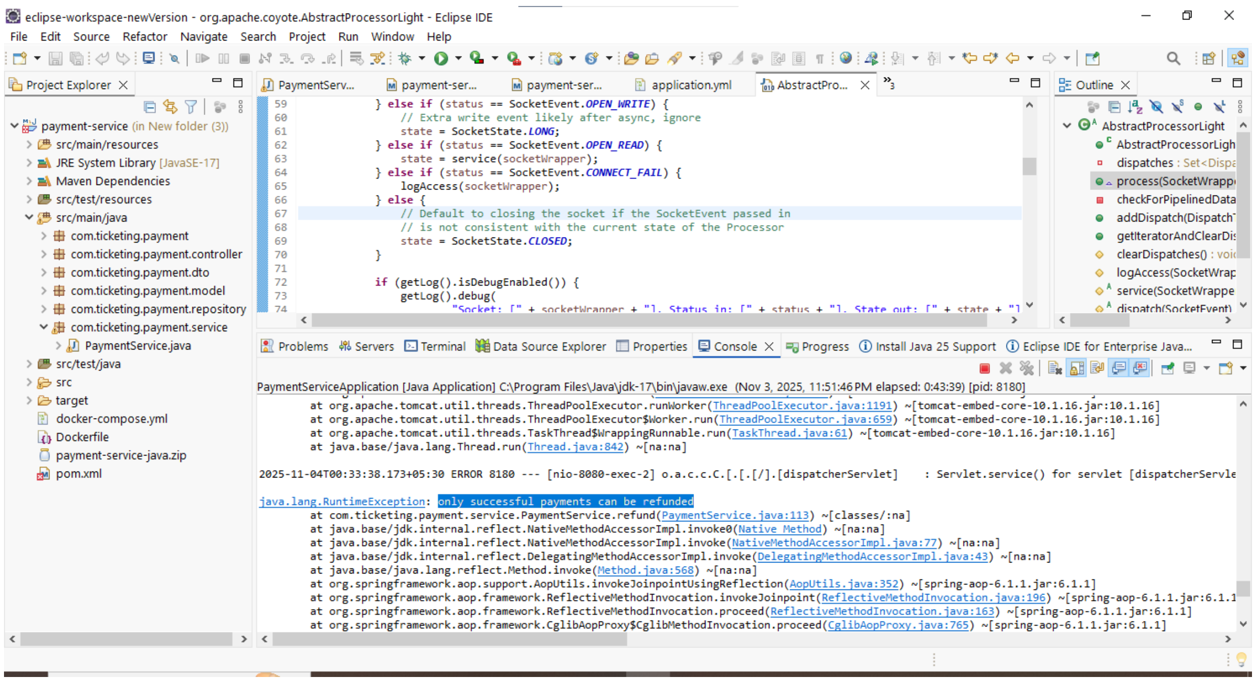
```
{
  "orderId": 1003,
  "amount": 1301,
  "currency": "INR",
  "method": "CARD"
}
```

```
{
    "amount": 1301.0,
    "orderId": 1003,
    "paymentId": 4,
    "status": "FAILED"
}
```


Command Prompt - psql -U postgres -d paymentsdb

```
paymentsdb=# SELECT * FROM idempotency_keys;
 id | idempotency_key |                       request_fingerprint                        |
        response_body                                          | response_code
----+-----------------+------------------------------------------------------------------+--------------------------
--------------------------------------------------------------+---------------
  1 | 5678-1234       | c46708cdcd28d36f751645caed72a7e079aded1ba37e3d987dbd66ee89b47f94 | {"reference":"ETP-c32da8ca","
amount":1300.0,"orderId":1003,"paymentId":3,"status":"SUCCESS"} |           200
(1 row)


paymentsdb=# SELECT * FROM refunds;
 id | amount | created_at | payment_id | provider_ref | status
----+--------+------------+------------+--------------+---------
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
(1 row)


paymentsdb=# SELECT * FROM payments;
 payment_id | amount | created_at | method | order_id |  reference   |  status
------------+--------+------------+--------+----------+--------------+----------
          1 |   1200 |            | CARD   |     1001 | ETP-e2ab4afb | REFUNDED
          2 |   1201 |            | CARD   |     1002 |              | FAILED
          3 |   1300 |            | CARD   |     1003 | ETP-c32da8ca | SUCCESS
          4 |   1301 |            | CARD   |     1003 |              | FAILED
(4 rows)


paymentsdb=# _
```

Failed status should not refund –

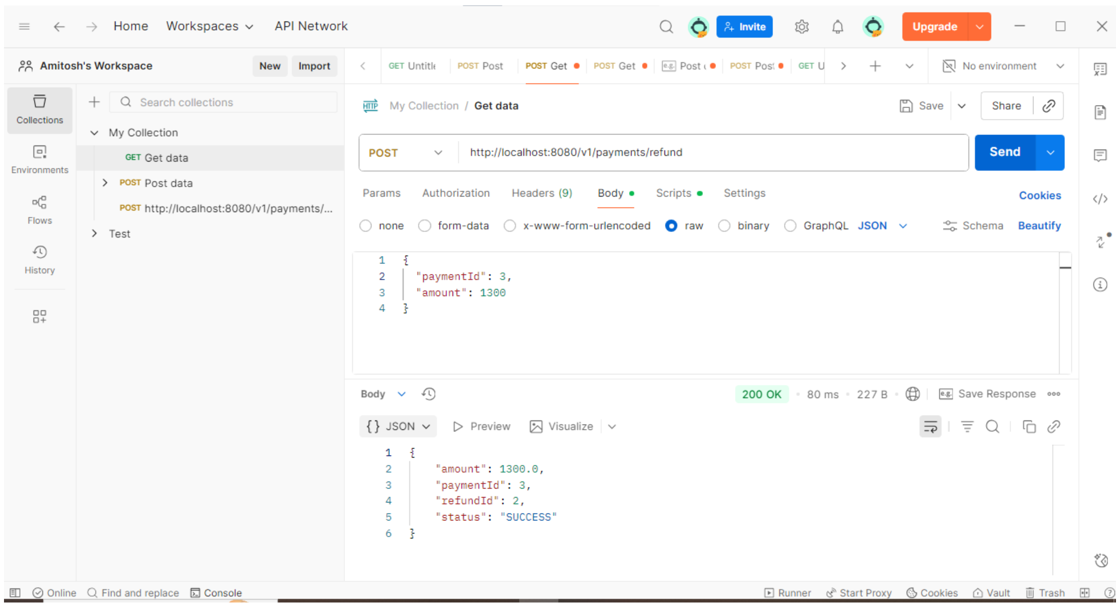

eclipse-workspace-newVersion - org.apache.coyote.AbstractProcessorLight - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
payment-service (in New folder (3))
src/main/resources
JRE System Library [JavaSE-17]
Maven Dependencies
src/test/resources
src/main/java
com.ticketing.payment
com.ticketing.payment.controller
com.ticketing.payment.dto
com.ticketing.payment.model
com.ticketing.payment.repository
com.ticketing.payment.service
PaymentService.java
src/test/java
src
target
docker-compose.yml
Dockerfile
payment-service-java.zip
pom.xml
PaymentServ... payment-ser... payment-ser... application.yml AbstractPro...
Outline
AbstractProcessorLight
AbstractProcessorLigh
dispatches : Set<Dispa
process(SocketWrapp
checkForPipelinedData
addDispatch(Dispatch
getIteratorAndClearDis
clearDispatches() : voi
logAccess(SocketWrap
service(SocketWrappe
dispatch(SocketEvent)
Problems Servers Terminal Data Source Explorer Properties Console Progress Install Java 25 Support Eclipse IDE for Enterprise Java...

```
            } else if (status == SocketEvent.OPEN_WRITE) {
                // Extra write event likely after async, ignore
                state = SocketState.LONG;
            } else if (status == SocketEvent.OPEN_READ) {
                state = service(socketWrapper);
            } else if (status == SocketEvent.CONNECT_FAIL) {
                logAccess(socketWrapper);
            } else {
                // Default to closing the socket if the SocketEvent passed in
                // is not consistent with the current state of the Processor
                state = SocketState.CLOSED;
            }

            if (getLog().isDebugEnabled()) {
                getLog().debug(
                        "Socket: [" + socketWrapper + "], Status in: [" + status + "], State out: [" + state + "]
```

PaymentServiceApplication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 3, 2025, 11:51:46 PM elapsed: 0:43:39) [pid: 8180]

```
	at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61) ~[tomcat-embed-core-10.1.16.jar:10.1.16]
	at java.base/java.lang.Thread.run(Thread.java:842) ~[na:na]

2025-11-04T00:33:38.173+05:30 ERROR 8180 --- [nio-8080-exec-2] o.a.c.c.C.[.[.[/].[dispatcherServlet]    : Servlet.service() for servlet [dispatcherServle

java.lang.RuntimeException: only successful payments can be refunded
	at com.ticketing.payment.service.PaymentService.refund(PaymentService.java:113) ~[classes/:na]
	at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[na:na]
	at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77) ~[na:na]
	at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:na]
	at java.base/java.lang.reflect.Method.invoke(Method.java:568) ~[na:na]
	at org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.java:352) ~[spring-aop-6.1.1.jar:6.1.1]
	at org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint(ReflectiveMethodInvocation.java:196) ~[spring-aop-6.1.1.jar:6.1.1
	at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163) ~[spring-aop-6.1.1.jar:6.1.1]
	at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:765) ~[spring-aop-6.1.1.jar:6.1.1]
```

Refund - http://localhost:8080/v1/payments/refund -

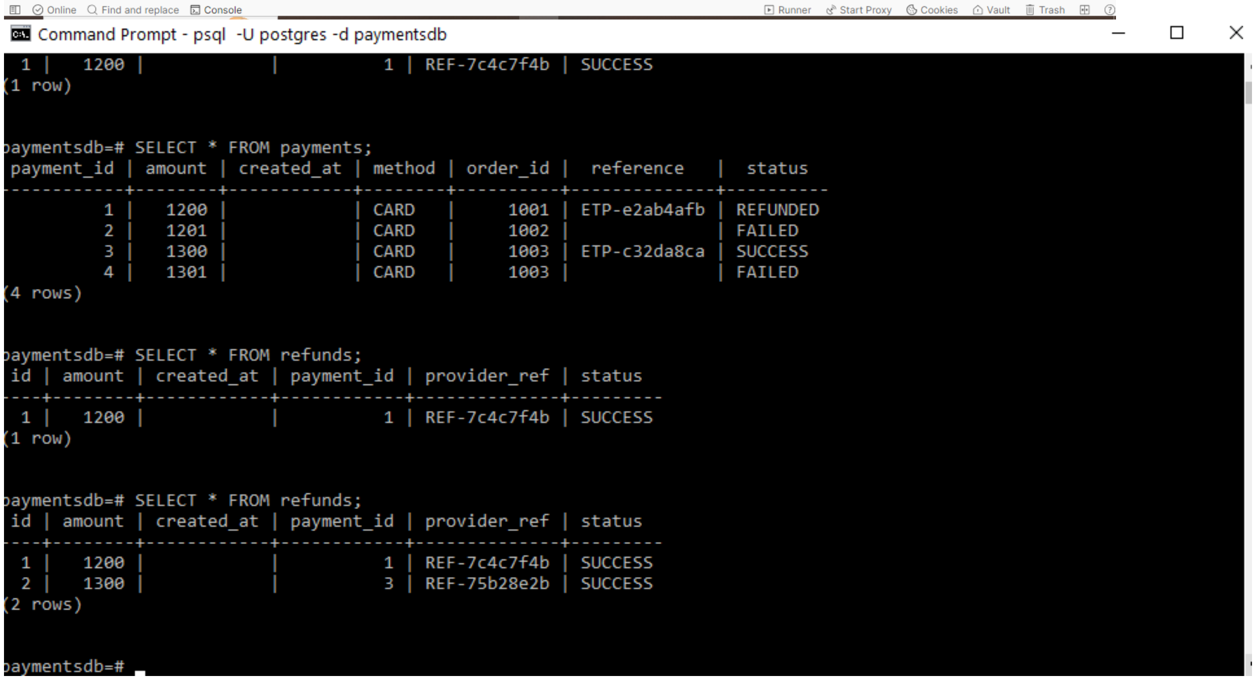


POST http://localhost:8080/v1/payments/refund

```
{
  "paymentId": 3,
  "amount": 1300
}
```

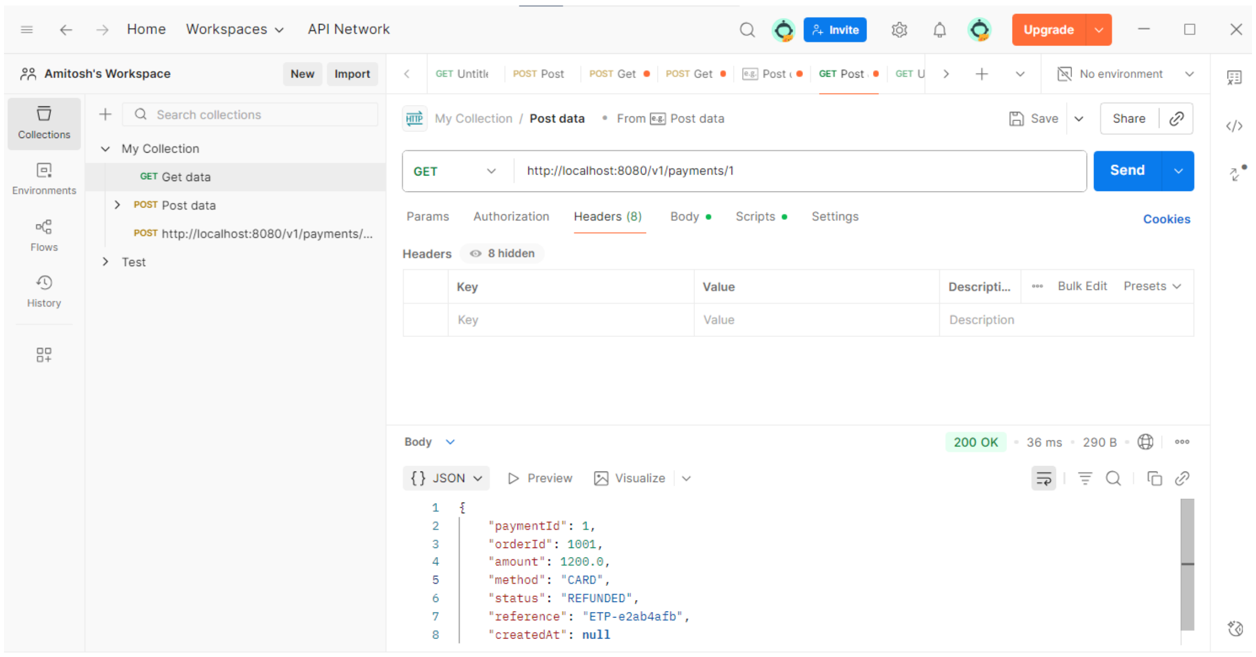
Response:

```
{
    "amount": 1300.0,
    "paymentId": 3,
    "refundId": 2,
    "status": "SUCCESS"
}
```



```
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
(1 row)


paymentsdb=# SELECT * FROM payments;
 payment_id | amount | created_at | method | order_id |  reference   |  status
------------+--------+------------+--------+----------+--------------+----------
          1 |   1200 |            | CARD   |     1001 | ETP-e2ab4afb | REFUNDED
          2 |   1201 |            | CARD   |     1002 |              | FAILED
          3 |   1300 |            | CARD   |     1003 | ETP-c32da8ca | SUCCESS
          4 |   1301 |            | CARD   |     1003 |              | FAILED
(4 rows)


paymentsdb=# SELECT * FROM refunds;
 id | amount | created_at | payment_id | provider_ref | status
----+--------+------------+------------+--------------+---------
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
(1 row)


paymentsdb=# SELECT * FROM refunds;
 id | amount | created_at | payment_id | provider_ref | status
----+--------+------------+------------+--------------+---------
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
  2 |   1300 |            |          3 | REF-75b28e2b | SUCCESS
(2 rows)


paymentsdb=# _
```

Payement ID – 1 ---



GET http://localhost:8080/v1/payments/1

```
{
    "paymentId": 1,
    "orderId": 1001,
    "amount": 1200.0,
    "method": "CARD",
    "status": "REFUNDED",
    "reference": "ETP-e2ab4afb",
    "createdAt": null
```

Testing and Validation -

Testing was performed using Postman and Swagger UI.
Verified endpoints:

- Charge request with and without Idempotency-Key.
- Refund of successful transactions.
- Retrieval of payment details.
- Health and Prometheus monitoring endpoints.

Confirmed:

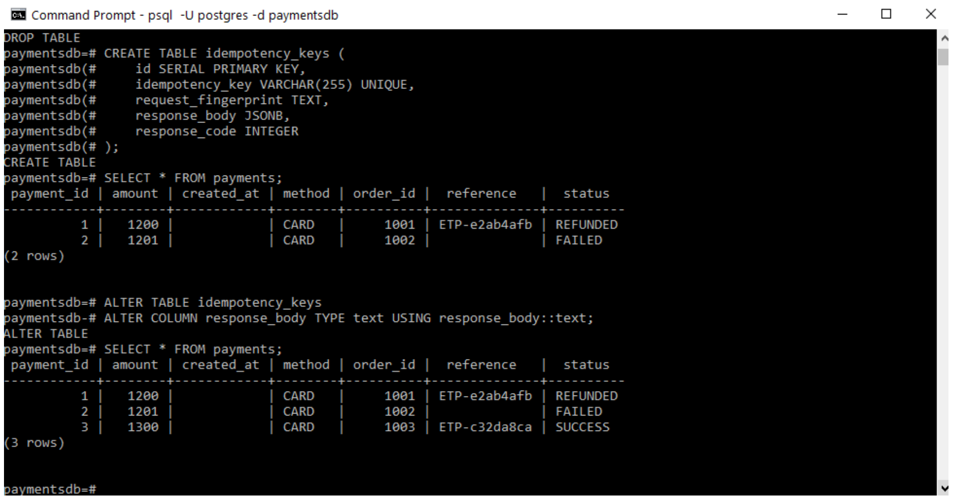
- Database persistence in PostgreSQL.
- Accurate status transitions (PENDING → SUCCESS/FAILED → REFUNDED).
- Consistent JSON responses.
- Duplicate prevention via Idempotency-Key logic.

Database – PostgreSQL -

Tables

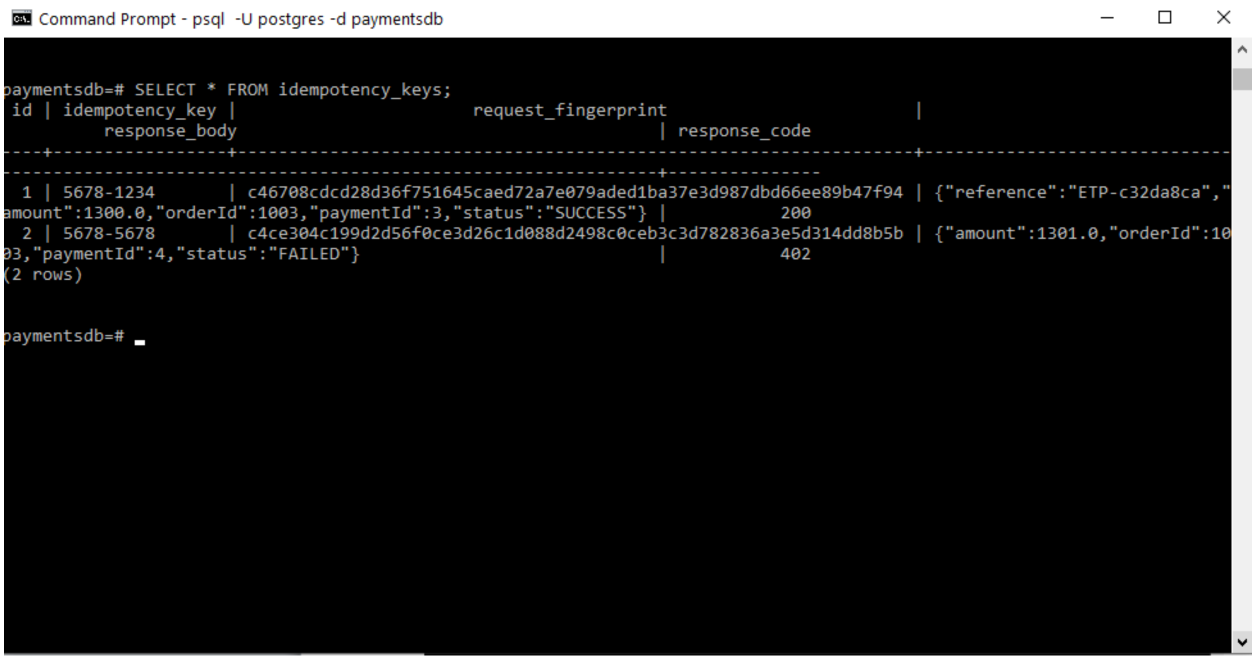
- ***payments***: Stores charge details and transaction statuses.
- ***refunds***: Tracks refund operations.
- ***idempotency_keys***: Manages request fingerprints and cached responses.

SELECT * FROM payments;



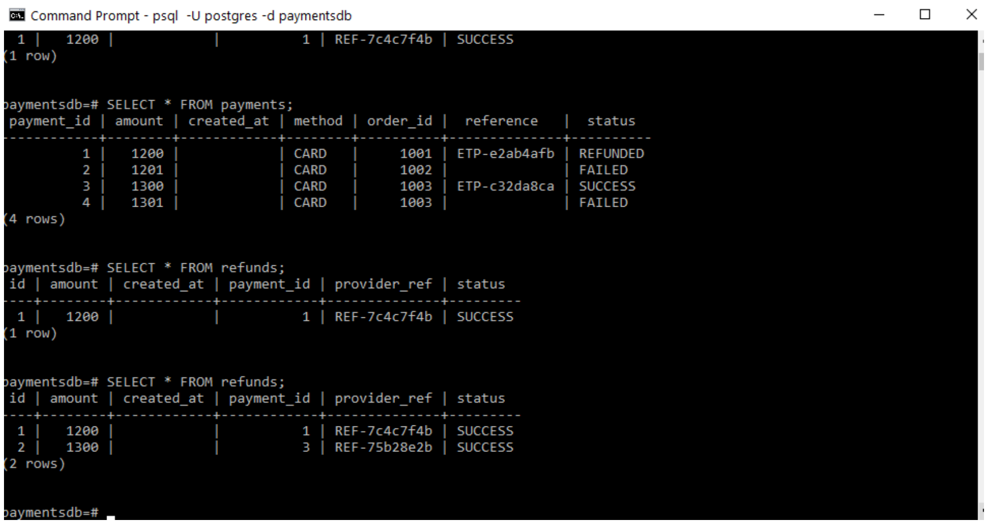
```
Command Prompt - psql  -U postgres -d paymentsdb
DROP TABLE
paymentsdb=# CREATE TABLE idempotency_keys (
paymentsdb(#     id SERIAL PRIMARY KEY,
paymentsdb(#     idempotency_key VARCHAR(255) UNIQUE,
paymentsdb(#     request_fingerprint TEXT,
paymentsdb(#     response_body JSONB,
paymentsdb(#     response_code INTEGER
paymentsdb(# );
CREATE TABLE
paymentsdb=# SELECT * FROM payments;
 payment_id | amount | created_at | method | order_id |  reference   |  status
------------+--------+------------+--------+----------+--------------+----------
          1 |   1200 |            | CARD   |     1001 | ETP-e2ab4afb | REFUNDED
          2 |   1201 |            | CARD   |     1002 |              | FAILED
(2 rows)


paymentsdb=# ALTER TABLE idempotency_keys
paymentsdb-# ALTER COLUMN response_body TYPE text USING response_body::text;
ALTER TABLE
paymentsdb=# SELECT * FROM payments;
 payment_id | amount | created_at | method | order_id |  reference   |  status
------------+--------+------------+--------+----------+--------------+----------
          1 |   1200 |            | CARD   |     1001 | ETP-e2ab4afb | REFUNDED
          2 |   1201 |            | CARD   |     1002 |              | FAILED
          3 |   1300 |            | CARD   |     1003 | ETP-c32da8ca | SUCCESS
(3 rows)


paymentsdb=#
```

SELECT * FROM idempotency_keys;



```
Command Prompt - psql  -U postgres -d paymentsdb

paymentsdb=# SELECT * FROM idempotency_keys;
 id | idempotency_key |                       request_fingerprint                        |
          response_body                                                   | response_code
----+-----------------+------------------------------------------------------------------+-----------------------------
-------------------------------------------------------------------+---------------
  1 | 5678-1234       | c46708cdcd28d36f751645caed72a7e079aded1ba37e3d987dbd66ee89b47f94 | {"reference":"ETP-c32da8ca","
amount":1300.0,"orderId":1003,"paymentId":3,"status":"SUCCESS"} |           200
  2 | 5678-5678       | c4ce304c199d2d56f0ce3d26c1d088d2498c0ceb3c3d782836a3e5d314dd8b5b | {"amount":1301.0,"orderId":10
03,"paymentId":4,"status":"FAILED"}                               |           402
(2 rows)


paymentsdb=# _
```

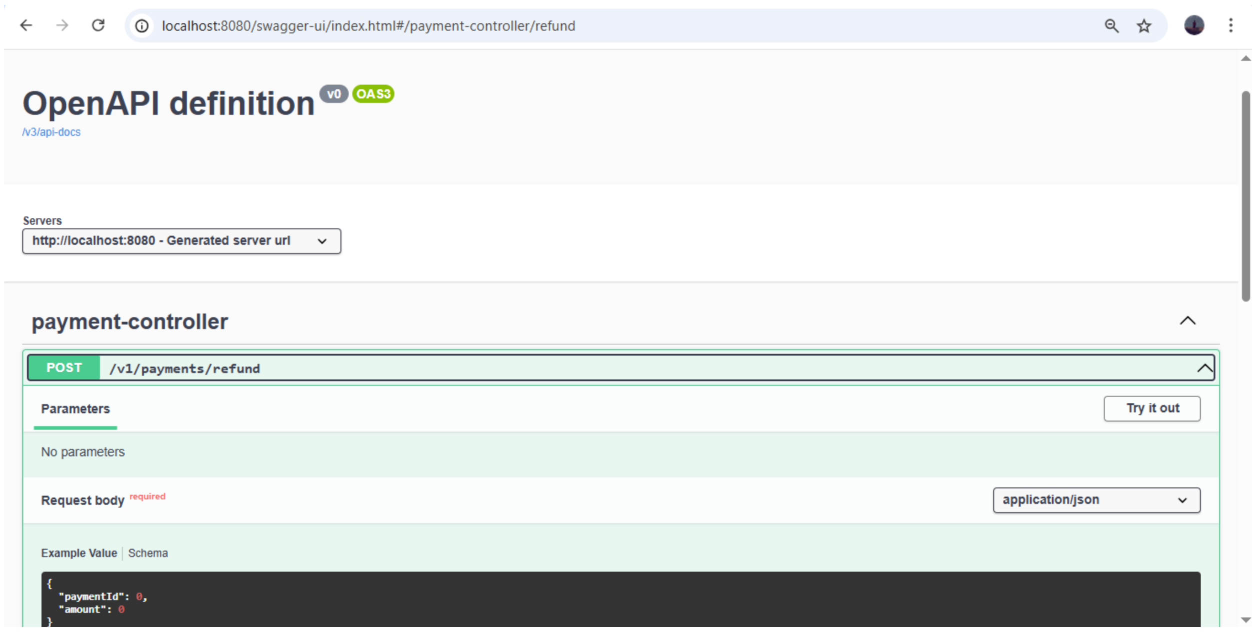
SELECT * FROM refunds;



Command Prompt - psql -U postgres -d paymentsdb

```
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
(1 row)


paymentsdb=# SELECT * FROM payments;
 payment_id | amount | created_at | method | order_id |  reference   |  status
------------+--------+------------+--------+----------+--------------+----------
          1 |   1200 |            | CARD   |     1001 | ETP-e2ab4afb | REFUNDED
          2 |   1201 |            | CARD   |     1002 |              | FAILED
          3 |   1300 |            | CARD   |     1003 | ETP-c32da8ca | SUCCESS
          4 |   1301 |            | CARD   |     1003 |              | FAILED
(4 rows)


paymentsdb=# SELECT * FROM refunds;
 id | amount | created_at | payment_id | provider_ref | status
----+--------+------------+------------+--------------+---------
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
(1 row)


paymentsdb=# SELECT * FROM refunds;
 id | amount | created_at | payment_id | provider_ref | status
----+--------+------------+------------+--------------+---------
  1 |   1200 |            |          1 | REF-7c4c7f4b | SUCCESS
  2 |   1300 |            |          3 | REF-75b28e2b | SUCCESS
(2 rows)


paymentsdb=# _
```

<u>Swagger UI –</u>



localhost:8080/swagger-ui/index.html#/payment-controller/refund

OpenAPI definition v0 OAS3

/v3/api-docs

Servers

http://localhost:8080 - Generated server url

payment-controller

POST /v1/payments/refund

Parameters

Try it out

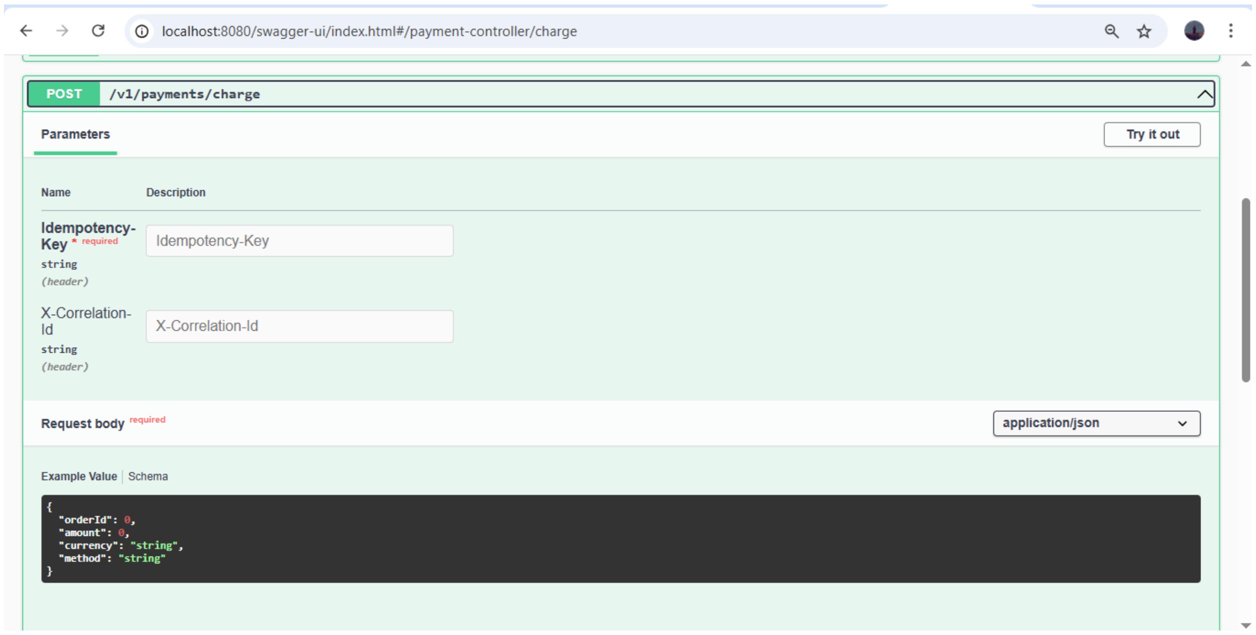
No parameters

Request body required

application/json

Example Value | Schema

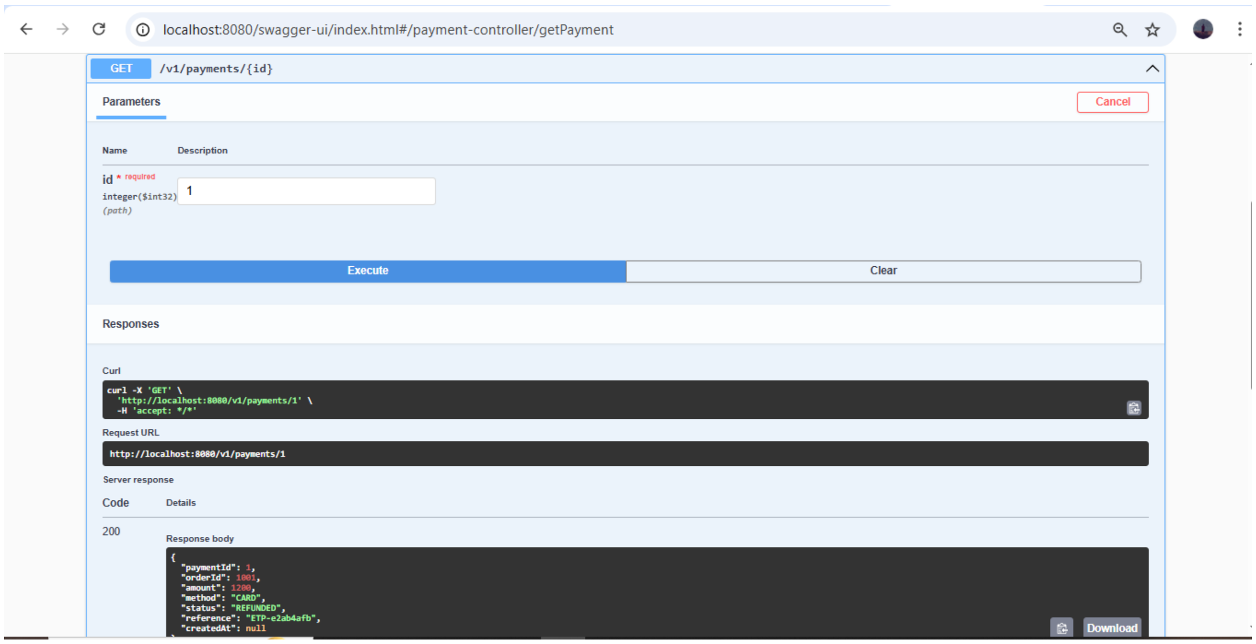
```
{
  "paymentId": 0,
  "amount": 0
}
```



POST /v1/payments/charge

Name	Description
Idempotency-Key * required string *(header)*	Idempotency-Key
X-Correlation-Id string *(header)*	X-Correlation-Id

```
{
  "orderId": 0,
  "amount": 0,
  "currency": "string",
  "method": "string"
}
```


localhost:8080/swagger-ui/index.html#/payment-controller/getPayment
GET
Parameters
Cancel
Execute
Clear
Responses
Download

GET /v1/payments/{id}

Name	Description
id * required integer($int32) *(path)*	1

Curl

```
curl -X 'GET' \
  'http://localhost:8080/v1/payments/1' \
  -H 'accept: */*'
```

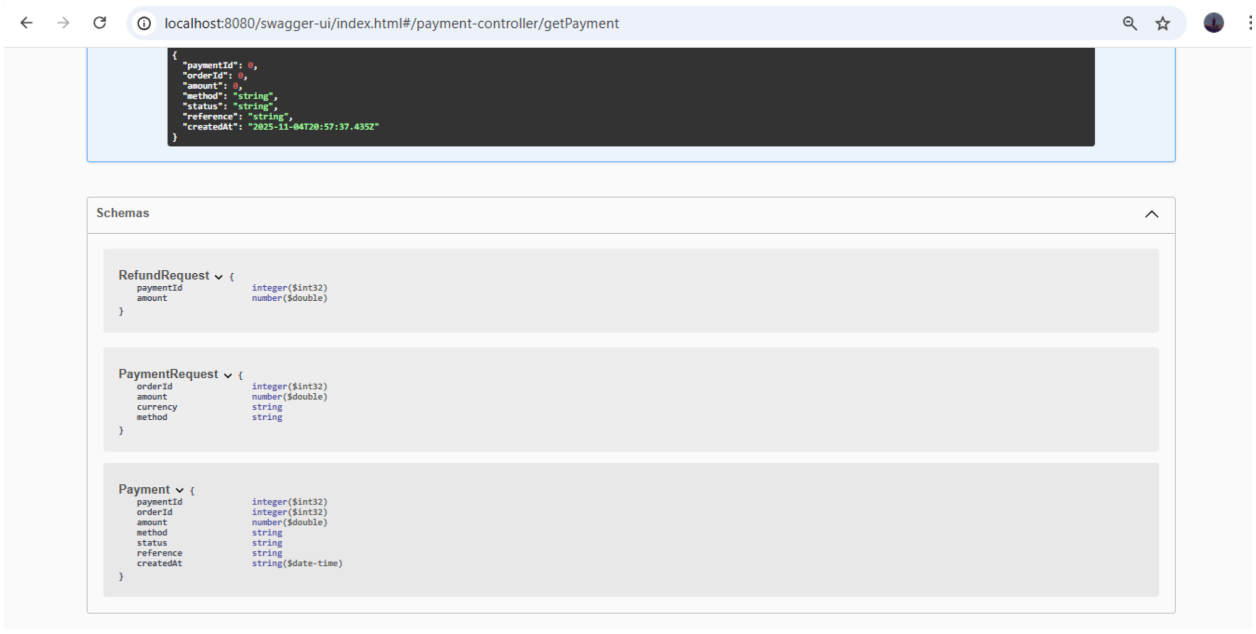
Request URL

```
http://localhost:8080/v1/payments/1
```

Server response

Code	Details
200	Response body

```
{
  "paymentId": 1,
  "orderId": 1001,
  "amount": 1200,
  "method": "CARD",
  "status": "REFUNDED",
  "reference": "ETP-e2ab4afb",
  "createdAt": null
```



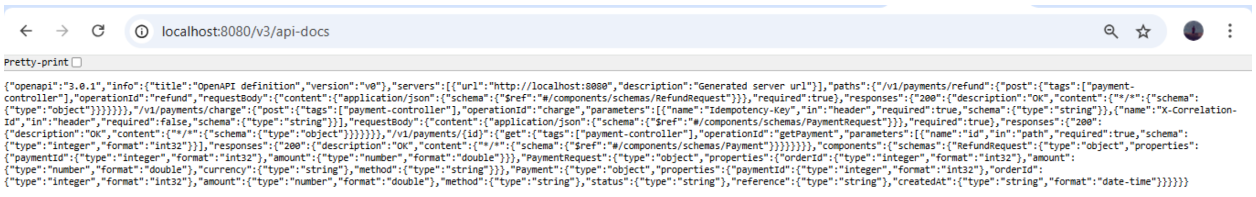
localhost:8080/swagger-ui/index.html#/payment-controller/getPayment

```
{
  "paymentId": 0,
  "orderId": 0,
  "amount": 0,
  "method": "string",
  "status": "string",
  "reference": "string",
  "createdAt": "2025-11-04T20:57:37.435Z"
}
```

Schemas

```
RefundRequest ⌄ {
   paymentId           integer($int32)
   amount              number($double)
}

PaymentRequest ⌄ {
   orderId             integer($int32)
   amount              number($double)
   currency            string
   method              string
}

Payment ⌄ {
   paymentId           integer($int32)
   orderId             integer($int32)
   amount              number($double)
   method              string
   status              string
   reference           string
   createdAt           string($date-time)
}
```

API docs -



localhost:8080/v3/api-docs

Pretty-print ☐

```
{"openapi":"3.0.1","info":{"title":"OpenAPI definition","version":"v0"},"servers":[{"url":"http://localhost:8080","description":"Generated server url"}],"paths":{"/v1/payments/refund":{"post":{"tags":["payment-
controller"],"operationId":"refund","requestBody":{"content":{"application/json":{"schema":{"$ref":"#/components/schemas/RefundRequest"}}},"required":true},"responses":{"200":{"description":"OK","content":{"*/*":{"schema":
{"type":"object"}}}}}}},"/v1/payments/charge":{"post":{"tags":["payment-controller"],"operationId":"charge","parameters":[{"name":"Idempotency-Key","in":"header","required":true,"schema":{"type":"string"}},{"name":"X-Correlation-
Id","in":"header","required":false,"schema":{"type":"string"}}],"requestBody":{"content":{"application/json":{"schema":{"$ref":"#/components/schemas/PaymentRequest"}}},"required":true},"responses":{"200":
{"description":"OK","content":{"*/*":{"schema":{"type":"object"}}}}}}},"/v1/payments/{id}":{"get":{"tags":["payment-controller"],"operationId":"getPayment","parameters":[{"name":"id","in":"path","required":true,"schema":
{"type":"integer","format":"int32"}}],"responses":{"200":{"description":"OK","content":{"*/*":{"schema":{"$ref":"#/components/schemas/Payment"}}}}}}}},"components":{"schemas":{"RefundRequest":{"type":"object","properties":
{"paymentId":{"type":"integer","format":"int32"},"amount":{"type":"number","format":"double"}}},"PaymentRequest":{"type":"object","properties":{"orderId":{"type":"integer","format":"int32"},"amount":
{"type":"number","format":"double"},"currency":{"type":"string"},"method":{"type":"string"}}},"Payment":{"type":"object","properties":{"paymentId":{"type":"integer","format":"int32"},"orderId":
{"type":"integer","format":"int32"},"amount":{"type":"number","format":"double"},"method":{"type":"string"},"status":{"type":"string"},"reference":{"type":"string"},"createdAt":{"type":"string","format":"date-time"}}}}}}
```

Key Learnings –

- Gained hands-on experience with Spring Boot 3, PostgreSQL, and Docker.
- Implemented idempotency for safe and reliable payment operations.
- Understood microservice design with clear service boundaries.
- Achieved successful deployment on Docker and Kubernetes.
- Strengthened understanding of RESTful APIs, error handling, and monitoring.

Conclusion –

The Payment Service successfully fulfills its role in the Event Ticketing System by ensuring secure, consistent, and idempotent transaction management.
Its modular design, containerized deployment, and clear API structure make it a reliable and scalable component in the distributed architecture.