

Notas BlackRock-FinancialWellbeing

Esta es una API construida con Java y el framework Spring Boot.

Se eligió este framework. Ya que permite crear de una manera fácil y rápida una API y es robusto y escalable.

Al mismo tiempo permite que en un futuro si se requiere se puede configurar todos los componentes necesarios.

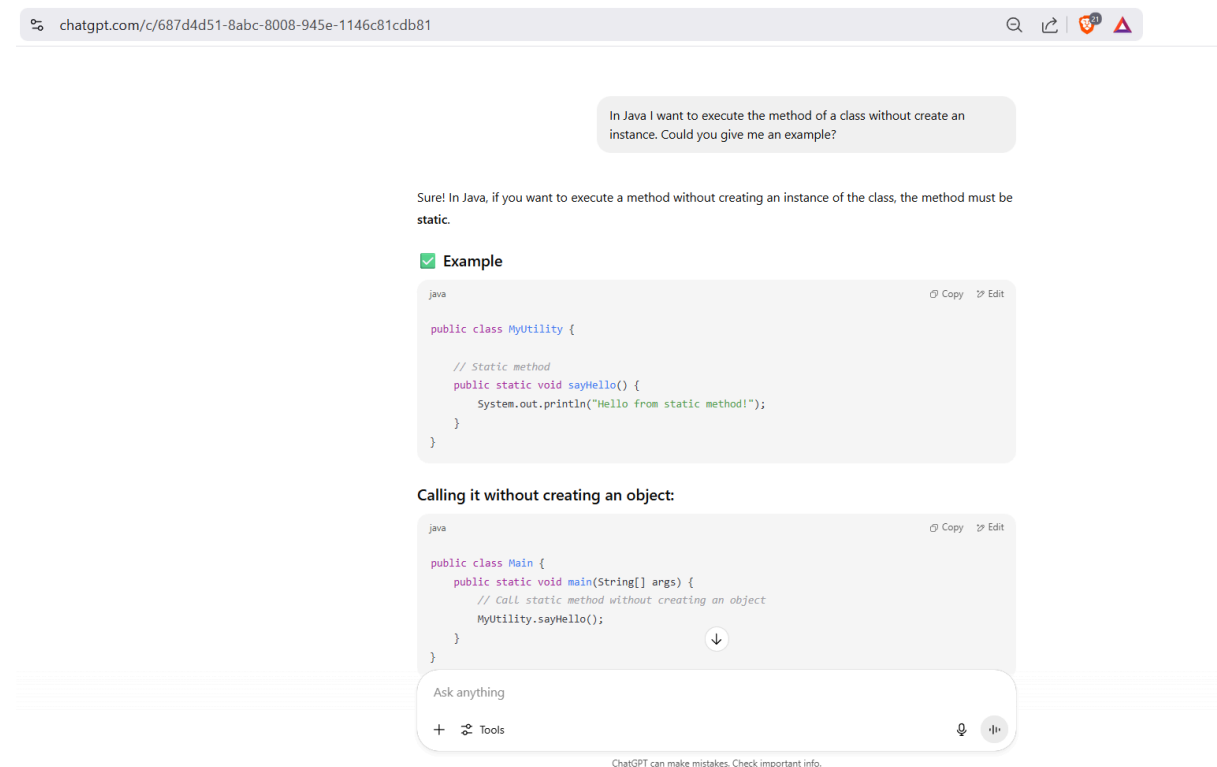
El endpoint que no alcance a completar fue el del transactions:filter

Mi idea era hacer una Lista con los arreglos de p, q, k y recorrer las diferentes transacciones para primero identificar si es una transacción válida y está en los rangos de k. En caso de que no esté en los rangos de k. Guardar la transacción en la lista de invalidas y agregar el mensaje de “La transacción no es válida porque no cumple con los rangos de k”.

En caso de que esté en los rangos de k. Se procederá a hacer las operación de q y p. y después de realizar las operación se vería si son válidas o no.

Se ocupó de la IA, en específico Chat GPT. En general fue un apoyo para consultar varias dudas que tenía.

Sin embargo yo tomaba su respuesta y lo implementa a mi proyecto. El único endpoint que me ayudó completamente fue el de performance. A continuación se comparten algunas de las consultas realizadas.



in java, i want to transform a string date to a Date in Java. This is the string:
"2023-10-12 20:15"
Could you give me some ideas?

Yes! To convert a string like "2023-10-12 20:15" into a `java.util.Date` or `java.time.LocalDateTime` in Java, you can use the modern `java.time` API (recommended) or the legacy `SimpleDateFormat` (not recommended for new code).

✓ Recommended (Java 8+) – Using `LocalDateTime` and `DateTimeFormatter`

```
java                                                                    Copy Edit

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class DateParser {
    public static void main(String[] args) {
        String dateString = "2023-10-12 20:15";

        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");
        LocalDateTime dateTime = LocalDateTime.parse(dateString, formatter);

        System.out.println(dateTime); // Output: 2023-10-12T20:15
    }
}
```

could you give me an example of the reverse method. I mean from `localdatetime` to string?

Absolutely! To convert a `LocalDateTime` back to a `String` in Java, you use the `DateTimeFormatter` with the `format()` method.

✓ Example: Convert `LocalDateTime` → `String`

```
java                                                                    Copy Edit

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class FormatDateExample {
    public static void main(String[] args) {
        // Create or get a LocalDateTime
        LocalDateTime dateTime = LocalDateTime.of(2023, 10, 12, 20, 15);

        // Define the format
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");

        // Convert to String
        String formattedDate = dateTime.format(formatter);

        System.out.println(formattedDate); // Output: 2023-10-12 20:15
    }
}
```

Ask anything

+ Tools

ChatGPT can make mistakes. Check important info.

what will be a good way to validate if the string can not be transformed to a localdatetime?

Great question! A good and safe way to validate whether a `String` can be converted to a `LocalDateTime` is by using a `try-catch` block. If the string is not in the expected format, a `DateTimeParseException` will be thrown.

✓ Example: Validate String to `LocalDateTime`

```
java
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeParseException;

public class DateValidator {
    public static void main(String[] args) {
        String input = "2023-10-12 20:15";
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm");

        LocalDateTime result = parseDateTime(input, formatter);

        if (result != null) {
            System.out.println("Valid date: " + result);
        } else {
            System.out.println("Invalid date f↓t.");
        }
    }
}
```

Ask anything

+ Tools

ChatGPT can make mistakes. Check important info.

if i create a method that throws an exception. How can i obligate the user to catch the exception?

Great question! In Java, you can force the caller of your method to handle an exception by throwing a **checked exception**.

✓ How to "Obligate" the Caller to Catch the Exception

1. Use a checked exception

Any class that extends `Exception` (but not `RuntimeException`) is a **checked exception**.

```
java
// Define a custom checked exception
public class InvalidDateFormatException extends Exception {
    public InvalidDateFormatException(String message) {
        super(message);
    }
}
```

2. Declare your method with `throws`

java



Copy Edit

to understand when you have an API and a endpoint as:
/hello/v1/transactions:validator

what means validator?

Great question!

In the endpoint:

```
bash
/hello/v1/transactions:validator
```

🔗 Copy 🗑️ Edit

The part after the colon — `:validator` — is **not a path segment** like the others (`/hello`, `/v1`, `/transactions`). Instead, it's typically used in one of two ways:

🌱 1. Colon as an operation or action

The `:validator` suffix is commonly used to indicate a **special operation** on a resource.

In this case, `transactions:validator` likely means:

“Perform a **validation operation** on a transaction.”

So the endpoint is **not fetching a transaction**, but instead calling a **validator function** or **triggering validation logic** on a transaction or transaction list.

This is especially common in:

- gRPC-REST style APIs



Ask anything

+ 🔗 Tools

🔍 🗑️

In my api with Spring boot. I want that the initial url show a message:
`http://localhost:8080/`
and when the user tries to another url that is no defined appears an error message. could you give me an example?

Certainly! Here's how you can do **two things** in your Spring Boot API:

✅ 1. Show a Custom Welcome Message at `/`

Return a simple string or JSON when the root path (`/`) is hit.

✅ 2. Show a Custom Error for Undefined URLs

Catch 404 errors and return a custom message or JSON response.

📦 Example Implementation

HomeController.java — Handles `/`

```
java
package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
```

🔗 Copy 🗑️ Edit

Ask anything




+ 🔗 Tools

🔍 🗑️

ChatGPT can make mistakes. Check important info.

In my java api. I want to create an endpoint that shows the performance. I need the metrics of the execution as time of response, memory use and threads. Could you help me?

Absolutely! You can create a **performance endpoint** in your Spring Boot API to return metrics like:

-  **Execution time** (response time)
-  **Memory usage** (heap, non-heap)
-  **Thread count**

This is useful for **monitoring, debugging, or performance testing**.

Step-by-Step: Create `/performance` Endpoint

PerformanceController.java

```
java
package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.lang.management.ManagementFactory;
import java.lang.management.MemoryMXBean;
import java.lang.management.MemoryUsage;
```

Ask anything

+ Tools

ChatGPT can make mistakes. Check important info.

could you only leave 3 variables:
"time" (execution duration in the format "2025-01-01 00:00:01.56").
"memory" memory use in Kb,
"threads"

Certainly! Here's a **clean and minimal** version of your Spring Boot performance endpoint, returning exactly the 3 variables you requested:

Output Format

```
json
{
  "time": "2025-01-01 00:00:01.56",
  "memory": 41256,
  "threads": 23
}
```

Updated PerformanceController.java

```
java
package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;
```

Ask anything

+ Tools

ChatGPT can make mistakes. Check important info.

could you only update the time. I want that if the api started "2025-07-20 13:00" and i check the performance api at "2025-07-20 14:00". I want that show the difference. I mean the output will be: "2025-07-20 01:00:00.00"

If i check the performance at "2025-07-21 13:00". The answer will be: "2025-07-20 24:00:00.00"

Got it! You want the `/performance` endpoint to show the **uptime duration** since the API started — formatted like:

arduino

```
"yyyy-MM-dd HH:mm:ss.SS"
```

Where:

- `yyyy-MM-dd` is the **startup date**
- `HH:mm:ss.SS` is the **duration** since startup (not the current time)

So if the app started at `2025-07-20 13:00`, and you're checking one hour later, the output should be:

arduino

```
"2025-07-20 01:00:00.00"
```

✓ Updated Controller with Uptime Tracking

Ask anything

+  Tools

ChatGPT can make mistakes. Check important info.

I have a list of objects in Java. start, end that are `LocalDateTime` and a `x` value. I want to sort in a optimize way from the oldest to the newest. Considering the start attribute. Could you help me?

Absolutely! To sort a list of Java objects by the `start` attribute (which is a `LocalDateTime`) from the **oldest to the newest**, you can use Java's `Comparator` with `List.sort()`.

✓ Example

Assume your class looks like this:

java

```
import java.time.LocalDateTime;

public class MyObject {
    private LocalDateTime start;
    private LocalDateTime end;
    private int x;

    // Constructor, getters and setters

    public MyObject(LocalDateTime start, LocalDateTime end, int x) {
        this.start = start;
        this.end = end;
        this.x = x;
    }

    public LocalDateTime getStart() {
        return start;
    }
}
```

Ask anything

+  Tools

ChatGPT can make mistakes. Check important info.

give me an example of how to deploy a java api in docker

Here's a full **step-by-step** example of how to **deploy a Java Spring Boot API in Docker**.

1. Sample Java Spring Boot API

Let's assume you already have a Spring Boot application (e.g., a simple REST controller):

```
java
// src/main/java/com/example/demo/HelloController.java
package com.example.demo;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {
    @GetMapping("/hello")
    public String sayHello() {
        return "Hello from Docker!";
    }
}
```

Make sure your `pom.xml` includes the necessary Spring Boot dependencies and the `spring-boot-maven-plugin`.

Ask anything

+  Tools

Send your updates. Or, simply select a new question.

Absolutely. Here's how to update the **Dockerfile** by explicitly choosing a Linux distribution (for example, **Debian**-based or **Alpine**-based image). I'll show you both:

Option 1: **Debian-based (recommended for compatibility)**

```
Dockerfile
# Base image: Debian with OpenJDK 17
FROM openjdk:17-jdk-slim-bullseye

# Set working directory inside container
WORKDIR /app

# Copy the Spring Boot fat JAR
COPY target/demo-0.0.1-SNAPSHOT.jar app.jar

# Expose the application's port
EXPOSE 8080

# Run the JAR
ENTRYPOINT ["java", "-jar", "app.jar"]
```

Advantages:

- Good compatibility.
- Smaller than full Debian.
- Stable and well-supported.

Ask anything

+  Tools