

Currency Converter API Documentation

Overview

The **Currency Converter API** is a Spring Boot application that provides real-time currency exchange rates and conversion functionalities using a public API.

Base URL

<http://localhost:8080/api>

1. Fetch Exchange Rates

GET /api/rates

Description

Fetches exchange rates for a given base currency.

Request Parameters

Parameter	Type	Description	Default
base	String	Base currency code (ISO 4217)	USD

Example Request

GET /api/rates?base=USD

Example Response

```
{
  "base": "USD",
  "rates": {
    "EUR": 0.94,
    "INR": 82.50,
    "GBP": 0.82
  }
}
```

Error Responses

HTTP Code	Message
400	Invalid base currency code
500	Failed to fetch exchange rates

2. Convert Currency

POST /api/convert

Description

Converts an amount from one currency to another using real-time exchange rates.

Request Body

```
{  
  "from": "USD",  
  "to": "EUR",  
  "amount": 100  
}
```

Example Request

POST /api/convert

Content-Type: application/json

Example Response

```
{  
  "from": "USD",  
  "to": "EUR",  
  "amount": 100,  
  "convertedAmount": 94.5  
}
```

Error Responses

HTTP Code	Message
400	Invalid currency code
500	Error fetching exchange rates

3. Error Handling

The API returns structured error messages when an exception occurs.

Example Error Response

```
{  
  "error": "Invalid currency code: XYZ"  
}
```

4. Running the Application

Requirements

- Java 17+
- Maven

Steps to Run

i. Clone the repository

git clone <https://github.com/your-repo/CurrencyConverter.git>

cd CurrencyConverter

ii. Build the application

mvn clean install

iii. Run the application

mvn spring-boot:run

5. Testing with Postman

1. Fetch Exchange Rates

- Method: GET
- URL: `http://localhost:8080/api/rates?base=USD`

2. Convert Currency

- Method: POST
- URL: `http://localhost:8080/api/convert`
- Body:

```
{  
  "from": "USD",  
  "to": "EUR",  
  "amount": 100  
}
```

6. Future Improvements

- Add authentication and authorization for secure access.
- Enhance error handling with more detailed messages.
- Implement a database-backed solution for storing rates and transactions.

7. Contact

For issues or feature requests, please create an issue in the [GitHub repository](#).