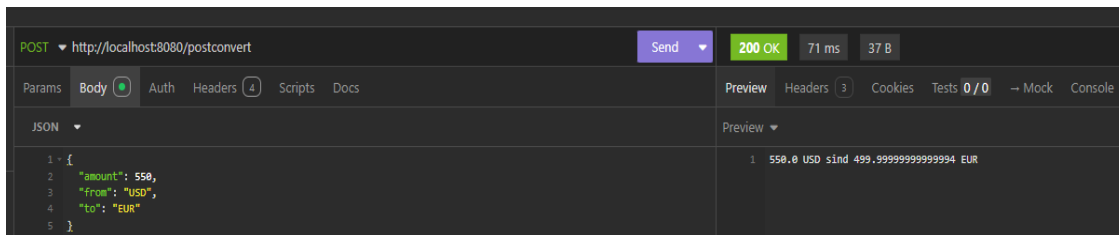


Webservice (REST) Entwicklung



1. Spring Boot-Projekt mit start.spring.io erstellen

1. Gehe zu <https://start.spring.io/>.
2. **Projekttyp:** Wähle "Maven Project".
3. **Sprache:** Wähle "Java".
4. **Spring Boot Version:** Wähle die neueste stabile Version (z.B. 3.x).
5. **Projekt-Metadaten:**
 - **Group:** `com.example`
 - **Artifact:** `currency_converter`
6. **Abhängigkeiten hinzufügen:**
 - Füge die Abhängigkeit **Spring Web** hinzu (dies stellt die nötigen Bibliotheken für den REST-Controller bereit).
7. Klicke auf **Generate** und lade das Projekt herunter.
8. Entpacke die ZIP-Datei und öffne das Projekt in **IntelliJ IDEA**.

2. Controller-Klasse erstellen

1. Im Projekt in `src/main/java/com/example/currency_converter` den Ordner `controller` erstellen.
2. Erstelle eine neue Java-Klasse namens `CurrencyConverterController` und füge den bereitgestellten Code ein.
3. Die Datei sollte in etwa so aussehen:

```
java
Code kopieren
package com.example.currency_converter;

import org.springframework.web.bind.annotation.*;

@RestController
public class CurrencyConverterController {

    private final double exchangeRateEURtoUSD = 1.1;

    @GetMapping("/convert")
    public String convertCurrency(
        @RequestParam double amount,
        @RequestParam String from,
        @RequestParam String to) {

        if (from.equalsIgnoreCase("EUR") && to.equalsIgnoreCase("USD")) {
            double convertedAmount = amount * exchangeRateEURtoUSD;
            return amount + " EUR sind " + convertedAmount + " USD";
        }
    }
}
```

```

        } else if (from.equalsIgnoreCase("USD") && to.equalsIgnoreCase("EUR")) {
            double convertedAmount = amount / exchangeRateEURtoUSD;
            return amount + " USD sind " + convertedAmount + " EUR";
        } else {
            return "Umrechnung von " + from + " zu " + to + " wird nicht
unterstützt.";
        }
    }

    @PostMapping("/postconvert")
    public String convertCurrency(@RequestBody ConversionRequest request) {
        if (request.getFrom().equalsIgnoreCase("EUR") &&
request.getTo().equalsIgnoreCase("USD")) {
            double convertedAmount = request.getAmount() * exchangeRateEURtoUSD;
            return request.getAmount() + " EUR sind " + convertedAmount + "
USD";
        } else if (request.getFrom().equalsIgnoreCase("USD") &&
request.getTo().equalsIgnoreCase("EUR")) {
            double convertedAmount = request.getAmount() / exchangeRateEURtoUSD;
            return request.getAmount() + " USD sind " + convertedAmount + "
EUR";
        } else {
            return "Umrechnung von " + request.getFrom() + " zu " +
request.getTo() + " wird nicht unterstützt.";
        }
    }

    public static class ConversionRequest {
        private double amount;
        private String from;
        private String to;

        public double getAmount() { return amount; }
        public void setAmount(double amount) { this.amount = amount; }

        public String getFrom() { return from; }
        public void setFrom(String from) { this.from = from; }

        public String getTo() { return to; }
        public void setTo(String to) { this.to = to; }
    }
}

```

3. Anwendung starten

1. Starte die Anwendung, indem du die Hauptklasse (z.B. CurrencyConverterApplication) mit der main-Methode ausführst.
2. Spring Boot startet auf dem Standard-Port **8080**.

4. Endpunkte testen

GET-Endpunkt /convert testen

Dieser Endpunkt erwartet drei Parameter (amount, from, to).

1. Mit Insomnia:

- Öffne Insomnia und erstelle eine neue GET-Anfrage.

- Setze die URL auf `http://localhost:8080/convert?amount=100&from=EUR&to=USD`.
- Klicke auf "Send", um die Anfrage zu senden.
- Die Antwort sollte etwa lauten: "100 EUR sind 110.0 USD".

2. Mit PowerShell:

```
powershell
Code kopieren
Invoke-RestMethod -Uri "http://localhost:8080/convert?amount=100&from=EUR&to=USD" -Method GET
```

POST-Endpoint `/postconvert` testen

Dieser Endpoint erwartet ein JSON-Objekt im Anfragekörper mit den Feldern `amount`, `from`, und `to`.

1. Mit Insomnia:

- Erstelle eine neue POST-Anfrage.
- Setze die URL auf `http://localhost:8080/postconvert`.
- Wähle im Reiter "Body" das Format "JSON".
- Füge folgendes JSON ein:

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

- Klicke auf "Send", um die Anfrage zu senden.
- Die Antwort sollte etwa lauten: "100 EUR sind 110.0 USD".

2. Mit PowerShell:

```
powershell
Code kopieren
$body = @{
  amount = 100
  from = "EUR"
  to = "USD"
} | ConvertTo-Json
```

```
Invoke-RestMethod -Uri "http://localhost:8080/postconvert" -Method POST -
Body $body -ContentType "application/json"
```