

## MESSPROTOKOLL PRAKTIKUM 02

### 1. Einführung

Im Rahmen dieses Protokolls wird die Implementierung einer Web-Schnittstelle für die Banken-Komponenten beschrieben. Das Ziel besteht darin, den Mitarbeiter:innen der Banken eine benutzerfreundliche Oberfläche zur Verfügung zu stellen. Die bestehende UDP-Schnittstelle bleibt weiterhin aktiv.

Im diesem Praktikum haben wir zwei Komponenten für die Web-Schnittstelle entwickelt. Die erste Komponente ist das Frontend, das für die Darstellung der HTML-Seite verantwortlich ist. Um die Kommunikation zwischen dem Frontend und der Bank zu ermöglichen, wurde das Backend implementiert.

### 2. Funktionalitäten der Web-Schnittstelle

Zur Realisierung der Web-Schnittstelle wurden sowohl das Frontend als auch das Backend für die Web-Schnittstelle entwickelt. Das Frontend besteht aus der Browser.html-Datei und der Browser.js-Datei, die für die Darstellung der HTML-Seite im Browser verantwortlich sind. Das Backend umfasst die WebClient.js-Datei, die mithilfe von Node.js ausgeführt wird und die Kommunikation zwischen dem Frontend und der Bank ermöglicht.

Die Web-Schnittstelle bietet folgende Hauptfunktionen:

#### 2.1 Gesamtwertabfrage per HTTP GET

Über die Web-Schnittstelle ist es möglich, den Gesamtwert der Bank abzufragen. Dies erfolgt mittels einer HTTP GET-Anfrage.

```
23 app.get('/api', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : void => {
24     startTime = Date.now();
25     console.log("Start time: " + startTime);
26     requestTotalBalanceFromBank();
27     res.json( body: {msg: "TotalBalance: " + bankBalance});
28     bankBalance = 0;
```

## 2.2 Manuelle Eingabe neuer Daten per HTTP POST

Des Weiteren ermöglicht die Web-Schnittstelle die manuelle Eingabe neuer Daten. Hierbei handelt es sich beispielsweise um Ein- und Auszahlungen, Kreditvergabe, Portfolioänderungen sowie den Kauf und Verkauf von Wertpapieren. Diese Eingaben werden per HTTP POST übermittelt.

```

32 app.post( path: '/api', bodyParser, (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<any, Record<...>> ) : void => {
33     console.log("Client message: "+req.body.msg);
34     startTime = Date.now();
35     console.log("Start time: " + startTime);
36     if(req.body.msg.includes( value: "Add")){
37         let split = req.body.msg.split(":");
38         tcpSocket.write( buffer: "POST" + " " + "Add:"+split[1]+"\\n");
39     }else if(req.body.msg.includes( value: "Subtract")){
40         let split = req.body.msg.split(":");
41         tcpSocket.write( buffer: "POST" + " " + "Subtract:"+split[1]+"\\n");
42     }else if(req.body.msg.includes( value: "TotalBalance")){
43         requestTotalBalanceFromBank();
44         res.json( body: {msg: "TotalBalance: " + bankBalance});
45         bankBalance = 0;
46     }else if(req.body.msg.includes( value: "DISCONNECT")) {
47         tcpSocket.write( buffer: "DISCONNECT" + "\\n");
48         tcpSocket.end();
49         rl.close();
50     }else{
51         res.json( body: {msg: "Unknown command"});
52     }
53 }

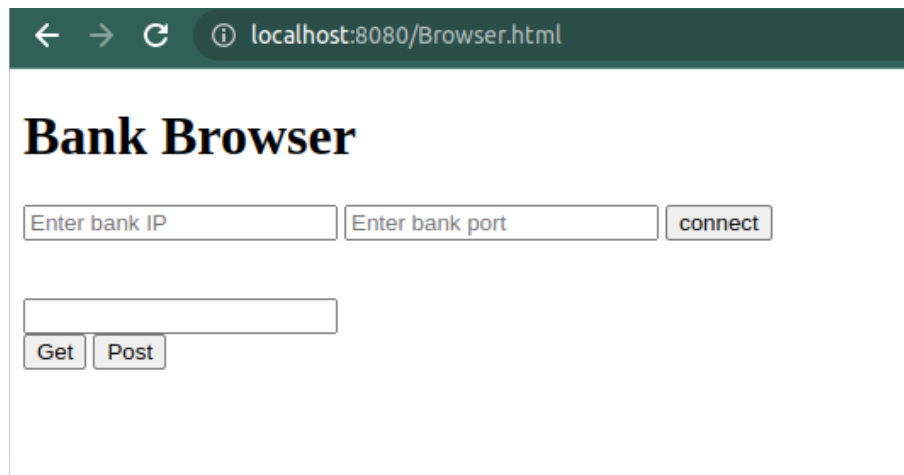
```

## 3. Auswirkungen auf den Gesamtwert der Banken

Die über die HTTP POST-Anfragen eingegebenen Daten haben Auswirkungen auf den Gesamtwert der Banken. Dies bedeutet, dass die Daten, die über die Web-Schnittstelle eingegeben werden, den Gesamtwert der Bank beeinflussen. Diese Ereignisse haben Konsequenzen sowohl für die HTTP GET-Schnittstelle als auch für die UDP-Schnittstelle.

#### 4. Vorführung der HTTP-Schnittstelle im Browser

Die HTTP-Schnittstelle soll in einem gängigen Browser wie Firefox oder Google Chrome vorgeführt werden. Hierbei können die Mitarbeiter:innen der Banken die Funktionalitäten der Web-Schnittstelle interaktiv nutzen und testen.



← → ↻ ⓘ localhost:8080/Browser.html

## Bank Browser

#### 5. Performanz-Messungen

Zur Bewertung der Leistungsfähigkeit der Web-Schnittstelle wurden Metriken wie die Round Trip Time (RTT) und der Throughput für jedes Paket gemessen, das von WebClient an die Bank gesendet wird. Die Ergebnisse dieser Messungen werden in der Konsole angezeigt.

```
WebClient      | RTT: 6 ms | Throughput: 0.234666666666666666 Mbps
WebClient      | Message from Bank: TotalBalance:2105357.2800123068
WebClient      |
```

#### 6. Bonus

Im Rahmen des Bonus können zusätzliche Funktionen und Verbesserungen für die Web-Schnittstelle implementiert werden. Beispielsweise könnten eine verbesserte Kommunikation zur Erleichterung der Arbeit der Mitarbeiter:innen sowie ein eigener Importer- oder Converter-Klient entwickelt werden, um Daten automatisch zu importieren.

## **7. Zusammenfassung**

Zusammenfassend beschreibt das Protokoll die Implementierung einer Web-Schnittstelle für Banken. Es umfasst ein Frontend zur Darstellung der HTML-Seite im Browser und ein Backend für die Kommunikation mit der Bank.

Die Hauptfunktionen sind die Abfrage des Bankwerts über HTTP GET und die manuelle Eingabe von Daten über HTTP POST.

Die Performanz wird durch Messung von RTT und Throughput bewertet.