

Übung zu Python und Qt

TGM – Schule der Technik

Autor: Walter Rafeiner-Magor, Clemens Koza

Version 4

Inhaltsverzeichnis

Übung zu Python und Qt	1
Inhaltsverzeichnis.....	2
1. Einführung	3
2. Hinweise zur Erfüllung der Anforderungen.....	3
3. Hinweise zu Unterlagen.....	3
4. Grundinhalt	3
4.1. Model	3
4.2. GUI	4
4.3. Weitere Vorgaben	4
5. Erweiterter Inhalt.....	5
5.1. Variante 1	5
5.1.1. Mindestanforderungen	5

1. Einführung

Representational State Transfer (REST) ist ein Architektur-Stil für verteilte Systeme, welcher auf eine möglichst einfache Architektur abzielt. RESTful Webservices erlauben den Zugriff auf Daten und verwenden üblicherweise weit verbreitete Web-Standards wie HTTP, URLs und JSON. Ein simples RESTful Webservice ist meist über eine bestimmte URL erreichbar und liefert je nach übergebenem GET-Parameter unterschiedliche Daten im JSON-Format zurück.

Sie wurden beauftragt, eine schlanke grafische Oberfläche für einen Währungsumwandler zu schreiben, der es erlaubt, einen Betrag aus einer Ursprungs-Währung in eine oder mehrere Zielwährungen umzuwandeln. Dabei soll es möglich sein, sowohl Live-Kurse (RESTful Webservice) als auch (fest hinterlegte) Offline-Daten zu verwenden. Sie müssen kein eigenes RESTful Webservice implementieren! Verwenden Sie ein kostenlos verfügbares Webservice, wie zum Beispiel folgendes Service von apilayer.com¹. Bitte den „Free Plan“ verwenden; mittels Subscribe registrieren und anmelden.

Die Verwendung kann mittels „Live Demo“ in vielen Programmiersprachen ausprobiert werden.

2. Hinweise zur Erfüllung der Anforderungen

Das Programm sollte die Anforderungen erfüllen, kann aber darüber hinaus auch weitere Funktionalität enthalten. Falls es bei der Programmierung Alternativen gibt, reicht die Umsetzung einer Variante aus.

3. Hinweise zu Unterlagen

Die Unterlagen sind als Unterstützung bei der Erarbeitung der Anforderungen gedacht. Diese decken jedoch nicht alle Anforderungen zu 100% ab.

4. Grundinhalt

Es werden viele Informationen via Webservice angeboten. Meist sind dazu eine Anmeldung bzw. manchmal auch eine Bezahlung notwendig. Bei unserem Beispiel ist ebenso eine Anmeldung notwendig, um einen API Access Key zu erhalten. Für unsere schulischen Übungszwecke werden wir den kostenlosen „Free plan“ verwenden, welcher für die Übung vollkommen ausreichend ist.

4.1. Model

Für die Kommunikation wird ein REST-Client erstellt, welcher Daten für den Aufruf des Webservices bereitstellt, und den Response aufbereitet.

¹ https://apilayer.com/marketplace/exchangerates_data-api

4.2. GUI

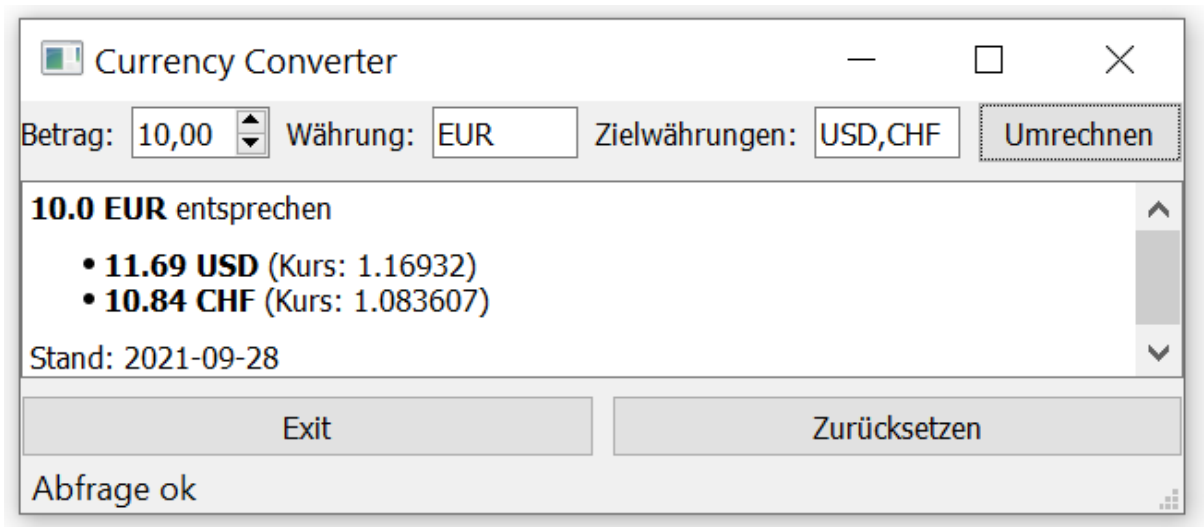


Abbildung 1: Einfacher GUI mit Basisfunktionalität

Folgende Informationen werden dargestellt:

- Sinnvoller Window-Titel (GKÜ)
- Eingabe eines Betrages via DoubleSpinner (GKÜ)
- Eingabe der Ausgangswährung via LineEdit (GKÜ)
- Eingabe der Zielwährungen via LineEdit (GKÜ)
- GUI zurücksetzen: Alle Eingabe-Widgets löschen, bzw. Betrag auf 0,00 setzen (GKÜ)
- GUI und Anwendung schließen (GKÜ)
- Neue Berechnung starten (GKÜ)
- Die Ausgabe wird userfreundlich angezeigt (GKV)
- Basisinformationen zur Berechnung via Statusbar (GKV)
- Fehler der Eingabe werden abgefangen und in der Statusbar zurückgemeldet (GKV)
- Fehler bei der Verwendung des Webservices werden abgefangen und in der Statusbar zurückgemeldet (GKV)

4.3. Weitere Vorgaben

Abgaben werden nach dem Softwaredesignmuster MVC erstellt und abgegeben.

5. Erweiterter Inhalt

Auf Basis des Grundinhaltes können noch Vertiefungen erstellt werden.

5.1. Variante 1

Ausgangs- und Zielwährungen können direkt im GUI ausgewählt werden. Es können neben den Live-Daten auch Daten aus anderen (lokalen) Quellen herangezogen und angezeigt werden.

Ausgangswährungen:

```
liste = ["EUR", "USD", "CHF"]
```

Zielwährungen:

```
liste = ["CAD", "HKD", "ISK", "PHP", "DKK", "HUF",  
        "CZK", "AUD", "RON", "SEK", "IDR",  
        "INR", "BRL", "RUB", "HRK", "JPY",  
        "THB", "CHF", "SGD", "PLN", "BGN", "TRY",  
        "CNY", "NOK", "NZD", "ZAR", "USD", "EUR",  
        "MXN", "ILS", "GBP", "KRW", "MYR"]
```

5.1.1. Mindestanforderungen

- Es kann eine vorgegebene Ausgangswährung gewählt werden via ComboBox (EKÜ)
- Es können mehrere vorgegebene Zielwährungen gewählt werden via ListWidget (EKÜ)
- Es können neben dem Live-Daten auch vorgefertigte Daten zur Verfügung und dargestellt werden (Hardcoded, File, ...) (EKÜ)
- Überlegen Sie eine sinnvolle Strategie, um beide Anforderungen im Model leicht austauschbar zu halten und für weitere Möglichkeiten offen zu bleiben. In einer abstrakten Basisklasse werden dabei die Gemeinsamkeiten beider Strategien abstrahiert. (EKV)
- Die Zielwährungen werden **beim Hinzufügen** aufsteigend sortiert. (EKV)
- Beim responsiven GUI wird sichergestellt, dass sich eine horizontale Veränderung nur auf den Abstand zwischen Ausgangswährung und Live-Daten auswirkt. (EKV)

Currency Converter

Betrag: 10,00 Ausgangswährung: EUR ☒ Live-Daten

Auswahl der Zielwährungen

- AUD
- BGN
- BRL
- CAD
- CHF
- CNY
- CZK
- DKK

Ausgabe

10.0 EUR entsprechen

- **16.06 AUD** (Kurs: 1.60645)
- **14.79 CAD** (Kurs: 1.479377)
- **254.44 CZK** (Kurs: 25.44408)

Stand: 2021-09-27

Exit Umrechnen Zurücksetzen

Abfrage ok