

## Übung 2: English-Translator als i18n Filter entwickeln

Ausgabe: 24.09.18

Besprechung: Selbststudium

Der Request auf `http://localhost:8080/flashcard-basic/questionnaires` führt zur Response in Abb. 1.

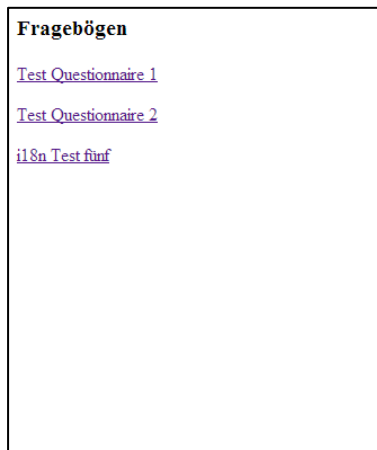


Abb. 1: Response auf `http://localhost:8080/flashcard-basic/questionnaires`

Wie in Abb.1 zu sehen ist die Sprache der Antwort nicht einheitlich. Mit einem Filter soll nun eine einfache Internationalisierung realisiert werden.

```
Fragebögen: Questionnaires  
fünf: five
```

Listing 1: Beispieleinträge in einer Sprachdatei

Ausgehend von einer Sprachdatei mit Key/Value Einträgen (siehe Listing 1) sollen sie in dieser Übung einen Filter entwickeln, der folgende Logik umsetzt:

- Dateiname eruieren:  
Dateiname der Sprachdatei aus dem ServletContext lesen
- Messages laden:  
i18n-Messages aus der Sprachdatei laden und in einer Key/Value Liste ablegen
- Deutsche Wörter übersetzen:  
Servlet Response im Filter abfangen, Response auf Messages untersuchen und falls Key vorhanden, diesen mit dem Value ersetzen.
- Definitive Response generieren:  
Übersetzte Response an Browser weiterleiten

Tipps:

1. Um die Servlet-Response bearbeiten zu können, muss die Klasse `HttpServletResponseWrapper` eingesetzt werden.  
Suchen sie im Internet (z.B. auf StackOverflow) nach einem entsprechenden Beispiel.
2. Der Name der Sprachdatei soll als init-Parameter im File `web.xml` konfiguriert werden können. Beachten sie das Error-Handling, falls dieser Wert nicht gesetzt ist. Setzen sie dann einen Default-Filename wie z.B. `messages.properties`.
3. Legen sie die Sprachdatei auf den Classpath z.B. in `src/main/resources`. Dann kann der Zugriff auf die Datei über  
`Thread.currentThread().getContextClassLoader().getResourceAsStream(filename)`  
erfolgen.
4. Eine mögliche Lösung des Filters finden sie auf dem AD. Verstehen sie den Vorschlag?