



Netzwerke I – Praktische Übungen

Übung 4: Webservices

Aufgabe 4.1: JSON (2 Punkte)

Neben dem in der Vorlesung vorgestellten Format XML wird häufig das Format JavaScript Object Notation (JSON) von Webservices verwendet. Es wird im RFC 4627 definiert.

- Machen Sie sich mit dem JSON Format vertraut. Welche Vor- und welche Nachteile sehen Sie verglichen mit XML?
- Formulieren Sie die Inhalte folgenden XML-Dokumentes in JSON:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<course>
  <name lang="English">Computer Networks I</name>
  <name lang="German">Netzwerke I</name>
  <typ>L</typ>
  <member>
    <id>1400</id>
    <firstname>Markus</firstname>
    <lastname>Mustermann</lastname>
  </member>
  <member>
    <id>1401</id>
    <firstname>Bart</firstname>
    <lastname>Simpson</lastname>
    <visiting />
  </member>
  <location room="R0.006" street="Lothstr. 64" city="Munich"/>
</course>
```

Aufgabe 4.2 – BreakfastToTheLimit – lange frühstücken und trotzdem pünktlich (8 Punkte)

Sie leben in einer WG in der Lothstraße 64 mit einer Mitbewohnerin namens Paula Pünktlich und einem Mitbewohner mit Namen Lothar Late. Sie alle frühstücken gerne, lange und ausführlich. Alle drei WG-Bewohner arbeiten an unterschiedlichen Orten und müssen daher in der Regel zu unterschiedlichen Zeiten aufbrechen um pünktlich bei der Arbeit zu sein. Wegen der stets unterschiedlichen Verkehrslage in München schwankt die Zeit, die Sie zur Fahrt bis zur Arbeitsstelle benötigen, stark. Da Sie nicht unnötig früh aufbrechen wollen, wenn mal weniger Verkehr ist, wollen Sie nun ein Programm entwickeln, welches über drei Hue-Lampen oberhalb des Frühstückstisches anzeigt, wer aufbrechen muss. Jedem WG-Bewohner/-in ist dabei genau eine Lampe zugeordnet, die durch unterschiedliche Farben/Blinken anzeigt, wie viel Zeit noch zum Frühstück verbleibt und wann aufgebrochen werden muss.

Daher sollen Sie im Folgenden eine kleine Anwendung entwickeln, die

- als Eingabe für jeden Mitbewohner einzeln den Arbeitsort, den Zeitpunkt des Arbeitsbeginns sowie das genutzte Verkehrsmittel (Auto, zu Fuß, Rad, ÖPNV) entgegennimmt



- mit Hilfe eines Webservice alle 5 Sekunden die aktuelle Reisezeit sowie den notwendigen Abfahrtszeitpunkt für alle Mitbewohner berechnet und mit der aktuellen Uhrzeit vergleicht
- 2 Minuten vor der notwendigen Abfahrzeit mit Hilfe der Philips Hue API (ein Webservice zur Lampensteuerung, weitere Hinweise dazu unten) die Lampe des Mitbewohners vom weißen Frühstückslight auf Orange umschaltet und dann 1 Minute vor der notwendigen Abfahrzeit auf Rot wechselt. (Der Mitbewohner beendet daraufhin hoffentlich das Frühstück und fährt los.)
- Falls bis zur Abfahrzeit keine Bestätigung erfolgt, dass der Mitbewohner losgefahren ist, die drei Lampen in der Küche rot blinken lässt – sodass die anderen Mitbewohner ihn zum Losfahren animieren, falls er wieder eingeschlafen ist.

Hinweise zur Umsetzung:

- Die Applikation kann als Standalone-Applikation mit Ausgabe über Konsole/AWT/Swing/JavaFX oder auch als Webapplikation/Servlet unter einem Servlet-Container wie Tomcat, o.ä. entwickelt werden. Im Falle einer solchen Webapplikation erfolgt die Ein-/Ausgabe über den Browser. Auch die Entwicklung einer Smartphone-Applikation (Android oder iOS) ist möglich. Die Wahl der Programmier-/Skriptsprache ist Ihnen für diese Aufgabe ebenfalls freigestellt.
- Die Art der Netzwerkkommunikation in dieser Aufgabe hängt davon ab, für welchen Weg der Implementierung Sie sich entscheiden:
 - **Standalone (Java-)Anwendung:** Direkte Bedienung der Socket-Schnittstelle. Dies ist der Ihnen bereits bekannte Weg über die Verwendung von Java Socket und URL/URLConnection.
 - **Servlet/Webanwendung:** Ein Servlet-Container nimmt Ihnen die Implementierung der Socket-Kommunikation ab und erlaubt die komfortable Verwaltung Ihrer Webapplikation. Im Internet findet sich eine Vielzahl an Tutorials, Sie finden sie über die Begriffe „Servlet Tutorial“ über die Suchmaschine Ihrer Wahl.
- Zur Lösung der Aufgabe ist die virtuelle Maschine nicht notwendig – Sie können die Aufgabe direkt auf einen privaten oder Labor-PC unter Windows/Linux entwickeln.
- Als Datenformat zur Abfrage der Informationen über die Webservices nutzen Sie bitte entweder JSON oder XML. Zur Verarbeitung beider Datenformate gibt es passende Bibliotheken/Toolkits, Beispiele für passende Bibliotheken finden Sie auf der Moodle-Seite zur Veranstaltung. **Eine Nutzung von Client-Bibliotheken, welche direkt die Verarbeitung des in Form von JSON oder XML zurückgelieferten Ergebnisses übernehmen (z.B. Google Clientbibliothek), ist für diese Aufgabe nicht zulässig.**

Hinweise zu Webservices

- Als Webservice zur **Ermittlung der aktuellen Reisezeit** können Sie die Google Distance Matrix API nutzen <https://developers.google.com/maps/documentation/distance-matrix/start?hl=de> oder einen anderen Webservice für Reisezeiten/Navigation der im Internet verfügbar ist. Eine Verwendung einer lokalen Datenbank ist nicht zulässig.
 - Zur Nutzung der Google-Dienste benötigen Sie einen API-Key, diesen können Sie kostenlos über die Google Cloud Console erzeugen (<https://console.cloud.google.com/>).
 - Falls Sie die Google-Webservices nicht nutzen wollen, finden Sie eine Reihe von Alternativen unter dem Verzeichnis von <http://www.programmableweb.com/>.
- Als **Webservice zur Ansteuerung der Lampen** nutzen Sie bitte die Philips Hue API (www.developers.meethue.com). Mit dieser können Sie auf einen Webservice zugreifen, der von der



Hue Bridge zur Verfügung gestellt wird. Die Bridge steuert dann wiederum die Lampen über eine drahtlose Zigbee-Verbindung, ein typischer Ansatz aus dem Bereich Internet-of-Things (IoT).

- Über jeder Tischgruppe im Praktikumsraum finden Sie daher einen Leuchter mit drei Hue-Lampen, welche von jeweils einer Bridge gesteuert werden (Farbe setzen, Helligkeit setzen, Color-Loop, blinken). Die Steuerung kann individuell/pro Lampe aber auch für eine ganze Lampengruppe erfolgen.
- Die Absicherung des Zugriffs auf eine Bridge erfolgt über einen sogenannten Authorized User. Dies ist eine (geheime) ID, welche hinter dem prefix „/api“ beim Aufruf der API angehängt wird, beispielsweise liest folgender GET-Request den Zustand der ersten Lampe der Bridge 1:
GET /api/197ea42c25303cef1a68c4042ed56887/lights/1/state
- Die hierfür notwendigen IP-Adressen und Authorized User entnehmen Sie bitte folgender Tabelle:

Bridge	IP-Adresse	Authorized User
Bridge 1	10.28.9.120	197ea42c25303cef1a68c4042ed56887
Bridge 2	10.28.9.121	3dc1d8f23e55321f3c049c03ac88dff
Bridge 3	10.28.9.122	2217334838210e7f244460f83b42026f
Bridge 3	10.28.9.123	2b2d3ff23d63751f10c1d8c0332d50ff

- Ein Zugriff auf die Lampen ist nur aus dem Labornetz möglich, d.h. falls Sie auf einem eigenen Rechner entwickeln, müssen Sie erkennen, ob die Lampen erreichbar sind oder nicht. Vor der Abnahme müssen Sie ggf. das Programm auf einen der Laborrechner übertragen, so dass die Lampen erreichbar sind. Zum Test Ihrer Lampenansteuerung außerhalb des Labores können Sie den Hue Emulator (<http://steveyo.github.io/Hue-Emulator/>) verwenden.

Bonus-Aufgabe (+2 Punkte)

Stellen Sie die Fahrtrouten der Studierenden auf einer Karte dar. Hierfür können Sie entsprechende Kartendienste (OpenStreetMap, Google Maps, etc.) nutzen.

Aufgabe 4.2 B (Alternative-Version der Aufgabe 4.2 für Bahn-Enthusiasten):

Gehen Sie von der gleichen Aufgabenstellung wie 4.2 aus, mit folgender Änderung: Alle Mitbewohner nutzen die Deutsche Bahn, um zur Arbeit zu kommen. Anstelle des Arbeitsortes fragen Sie bei Beginn die Zugnummer/Verbindung ab, statt die Reisezeit über den Google Webservice zu berechnen, nutzen Sie den Webservice der Deutschen Bahn (<https://developer.deutschebahn.com>) um Abfahrzeit und aktuelle Verspätungen zu den genutzten Zugverbindungen abzufragen und den Abfahrzeitpunkt zu berechnen. (Wie bei der Originalaufgabe 4.2 ist auch hier die Nutzung fertiger Client-Klassen zur Verarbeitung der Ergebnisse nicht gestattet.)

Notwendige Abgaben für dieses Übungsblatt:

- Erfolgreiche Bearbeitung des Online-Tests zu Blatt 4
- Dokumentierter Quellcode zur Aufgabe 4.2. Bitte kennzeichnen Sie innerhalb der implementierten Klassen, von welchem Teammitglied die Klasse implementiert worden ist und **achten Sie darauf, dass beide Teammitglieder einen in etwa gleichen Anteil an der Implementierung haben, das Gesamtprogramm verstehen und im Detail erklären können.**