

Bitte beachten Sie, dass *Präsenzaufgaben* zwar nicht schriftlich bearbeitet werden müssen und nicht bewertet werden, jedoch als Vorbereitung für den Übungstermin und für die Klausur notwendig sind.

Die Abgabe der *Pflichtaufgaben* erfolgt in Teams von 3–5 Studierenden online als exakt eine PDF-Datei spätestens bis zum oben genannten Termin. Eine spätere Abgabe ist nicht möglich. Bitte verwenden Sie dabei den Zugriffscode einer vorherigen Abgabe, wenn die Zusammensetzung Ihrer Kleingruppe unverändert geblieben ist. Achten Sie in jedem Fall auf die gleiche Schreibweise der Namen!

<https://svs.informatik.uni-hamburg.de/submission/for/vis17-8>

Die Übungen zu diesem Blatt finden vom 29.01.2018 bis 01.02.2018 statt.

Aufgabe 1: Service-orientierte Architekturen (3 Punkte)

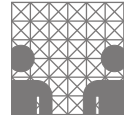
- a) (Pflichtaufgabe) Eine service-orientierte Architektur (SOA) wird oft im Kontext von Geschäftsprozessen verwendet. Erläutern Sie den Zusammenhang zwischen der SOA-Idee und Geschäftsprozessen in eigenen Worten. Stellen Sie auch dar, inwiefern die Wahl der Granularität von Diensten hier eine Rolle spielt. (3 Punkte)
- b) (Präsenzaufgabe) Bei einer Dienstkomposition gibt es die Begriffe der *Orchestrierung* und der *Choreographie*. Grenzen Sie die beiden Begriffe voneinander ab.
- c) (Präsenzaufgabe) UDDI galt als die wichtigste Technologie, um das organisationsübergreifenden Auffinden von Dienstleistungen zu ermöglichen. Allerdings existieren mittlerweile fast keine globalen UDDI-Verzeichnisse mehr. Überlegen und recherchieren Sie, warum viele große Unternehmen ihre Verzeichnisdienste eingestellt haben. Nennen Sie diese Gründe.

Aufgabe 2: Microservices (3 Punkte)

- a) (Pflichtaufgabe) Anwendungen, aber auch einzelne Services, können als größerer Monolith realisiert werden. Durch welche Nachteile von Monolithen sind die Microservices motiviert? Nennen Sie 3 Nachteile! (3 Punkte)
- b) (Präsenzaufgabe) Das Präfix „Micro“ deutet auf eine feine Granularität von Microservices hin. Aber wie klein oder umfangreich sollte ein Microservice gestaltet werden?

Aufgabe 3: REST-Service (14 Punkte)

In dieser Aufgabe sollen Sie einen eigenen REST-Service entwickeln. Hierfür nehmen wir das Beispiel einer einfachen Notizzettelverwaltung. Für die Entwicklung des Services folgen wir dem aus der Vorlesung bekannten Vorgehen (vgl. Folien VIS-SOA 48ff):



- a) Modellierung des Datenmodells
- b) Auswählen der extern sichtbaren Ressourcen (top-level URIs)
- c) Umsetzung der Methoden über REST
- d) Implementation

Die funktionalen Anforderungen an den Service lauten wie folgt:

- Anlegen einer neuen Notiz
- Auslesen aller angelegten Notizen
- Auslesen einer bestimmten Notiz
- Löschen einer bestimmten Notiz

Das Datenmodell für Notizen sei wie folgt definiert:

```
String text
String author
long timestamp
```

- a) (Pflichtaufgabe) Geben Sie zu jeder funktionalen Anforderung an die Notizzettelverwaltung einen beispielhaften HTTP-Request (in Kurzform) an. Folgen Sie dabei den Konventionen für REST-Services (4 Punkte)

Beispiel: GET /path/to/ressource/:id

Beispiel: POST /path/to/ressource?param1=value1

- b) (Pflichtaufgabe) Implementieren Sie den Service zum Verwalten. Als Ergebnis geben Sie bitte den Quelltext der Service-Klasse ab. (10 Punkte)

Hinweise:

- Die Implementierung kann einfach gehalten werden und muss nicht robust gegenüber falschen Anfragen sein.
- Ein einfacher Einstieg gelingt mittels des Spark-Frameworks für Java. Download, Dokumentation und Tutorials finden sich unter <http://sparkjava.com/>.
- Um Ihnen den Einstieg noch weiter zu vereinfachen, stellen wir Ihnen über STiNE ein kleines Eclipse-Projekt zur Verfügung. Dieses enthält alle benötigten Bibliotheken, das Datenmodell sowie eine Template-Klasse für den Spark-Service¹.
- Testen Sie Ihren Service ²! Im Browser bieten sich hierfür Plugins an, die auch POST-Anfragen verschicken können, z.B. *Postman* ³ für Chrome oder *Rested* ⁴ für Firefox.

¹Importieren Sie dieses Projekt in Eclipse über File -> Import -> General -> Existing Projects -> Archive File

²Bei Verwendung von Spark erreichen Sie Ihren Server unter der URL <http://localhost:4567/>

³<https://chrome.google.com/webstore/detail/postman-rest-client/fhbjgbiflinjbdggehcdcbncdddomop>

⁴<https://addons.mozilla.org/de/firefox/addon/rested/>