

Übung 08: REST-Server

Abgabetermin: 13. 6. 2022, 8:30

Aufgabe	Punkte	Abzugeben über Moodle
Übung 08	24	Java-Programm in Zip-Datei

Übung 08: HTML-Server mit REST (24 Points)

Die Johannes Kalender Universität¹ möchte im Zuge einer Digitalisierungsoffensive auch die Korrektur von Hausübungen transparenter gestalten. In einem ersten Schritt soll dabei den Studierenden Zugang zu Korrekturplänen gewährt werden. Nachdem mittlerweile alle Institute zur Einteilung und Koordination der Tutor*innen Ihre Kalenderanwendung verwenden, hat man darum beschlossen, ein öffentliches Web-Interface zu den in der Kalenderdatenbank gespeicherten Einträgen zu entwickeln. Da der Kalender in Java implementiert ist, soll dieses Webinterface als Java Servlet implementiert werden, welches über eine REST-Schnittstelle dynamisch generierte Webseiten zur Verfügung stellt. Studierende sollen damit über ihren Webbrowser die in der Datenbank gespeicherten Kalendereinträge einsehen aber nicht ändern können.

Als bekanntermaßen fähige*r und zuverlässige*r Softwareentwickler*in werden selbstverständlich wieder Sie mit der Implementierung dieser Schnittstelle beauftragt. Ihr RMI-basierter *SharedCalendar* wurde schon bei LVA-Leiter*innen und Tutor*innen gut angenommen. Allerdings dauern die Gespräche zwischen den einzelnen Instituten und dem Rektorat über die Verantwortung für und die Finanzierung von einem gemeinsamen Server noch an. Sie erhalten darum erneut eine (dieses Mal mit einigen Bugfixes versehene) Implementierung der ursprünglichen lokalen Kalenderanwendung, auf Basis derer Sie das Webinterface implementieren sollen.

Sie erhalten erneut eine Implementierung der Kalenderanwendung aus Aufgabe 2. Diese Implementierung gleicht dem zur Verfügung gestellten Framework für Übung 7, enthält aber zusätzliche Dateien für die REST-Implementierung, wie etwa eine *web.xml*. Implementieren Sie auf Basis dieser Anwendung ein Java Servlet, welches über eine geeignete REST-Schnittstelle einem Browser Zugriff auf den als HTML aufbereiteten Kalenderinhalt bietet. Konkret soll Ihre Schnittstelle zumindest folgende Abfragen ermöglichen:

- alle Termine eines bestimmten Monats, gruppiert in Tagen
- alle Termine eines bestimmten Jahres, gruppiert zumindest in Monaten
- alle Termine eines bestimmten Tages
- Einzelansicht eines Termins (Abfrage per ID)

Das Design der Schnittstelle - also die Benennung der Ressourcen, Aufbau der URIs und die Wahl zwischen *@PathParam* und *@QueryParam* - ist Ihnen überlassen. Dokumentieren Sie allerdings Ihre Schnittstelle in Form eines Blockkommentars am Anfang Ihrer Servlet-Klasse.

Treten bei der Bearbeitung einer Anfrage Exceptions auf, so übergeben Sie auch diese als HTML, allerdings in einer *Response* mit geeignetem Statuscode. Hierfür bietet es sich an, geeignete *ExceptionHandler* zu implementieren, das ist aber nicht verpflichtend.

Sie dürfen jeden Teil des zur Verfügung gestellten Codes verändern.

¹ Disclaimer: Die *Johannes Kalender Universität* ist eine fiktive Hochschule, die in keinem Zusammenhang zur Johannes Kepler Universität steht.

Hinweise:

- Einfache Formattierung mittels HTML Tags wie **...** (hervorgehoben) oder *<i>...</i>* (kursiv) ist ausreichend. Verwendung von CSS oder JavaScript ist ausdrücklich **nicht** erforderlich.
- Eine Implementierung der Kalenderanwendung wird Ihnen zur Verfügung gestellt. Diese Implementierung enthält eine JavaFX-basierte GUI und eine Derby-basierte Datenbankbindung. Verwenden Sie diese Implementierung als Grundlage für die Implementierung dieser Hausübung. Das gleiche Framework wird auch bei Übung 9 verwendet werden.
- Das zur Verfügung gestellte Zip-Archiv enthält neben dem Source Code der Kalenderanwendung auch eine *Maven* Konfigurationsdatei. Gängige Entwicklungsumgebungen wie Eclipse oder IntelliJ, die Maven-Unterstützung bieten, sollten aus dieser Konfigurationsdatei die benötigten Frameworks (JavaFX, Derby) auslesen und die notwendigen Dateien und Projekteinstellungen automatisch herunterladen bzw. setzen können. Z.B. in IntelliJ genügt: „Open“ -> <extrahiertes Zip-Archiv auswählen> um den Importvorgang anzustoßen. Sie dürfen die Abhängigkeiten aber bei Bedarf bzw. Problemen auch manuell konfigurieren. Beachten Sie bitte, dass das Framework zumindest Java 11 als Basis erwartet. Der Code sollte auch mit Java 8 funktionieren, allerdings wäre in diesem Fall wohl eine manuelle Korrektur der Abhängigkeiten notwendig, um zu Java 8 kompatible Versionen zu verwenden.
- Sie müssen sich nicht darum kümmern, das Modell-Objekt am Server freizugeben und den Server zu terminieren. Der Server kann endlos laufen, bis er manuell terminiert wird.
- Achten Sie auf eine saubere Implementierung.