

Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Frage: Passt die Vorlesung zu allen WebServices?

Antwort: Generell ermöglicht es Java recht tief auf die Protokollebene zuzugreifen, so dass Java zum Testen genutzt werden kann, aber sicherlich nicht muss. Dadurch ist Java auch für alle Arten von typischen WebServices zum Erstellen und Testen einsetzbar.

Frage: Könnte ich damit auch WebServices testen, die nicht in Java geschrieben sind?

Antwort: Außer der Information über das genutzte Protokoll muss bei der Nutzung von WebServices nichts über den Anbieter bekannt sein. Der Service in der Vorlesung und im Praktikum ist z. B. in Go geschrieben.

Frage: Nutzen wir eigentlich REST-Services?

Antwort, Achtung, REST ist kein Protokoll, sondern ein Architekturstil für die Gestaltung von webdienst-basierten Programmen. REST verwendet standardmäßig HTTP-Methoden und ist stark auf Performanz und Skalierbarkeit ausgelegt. Im Gegensatz zu SOAP, das hauptsächlich XML nutzt, kann REST mit verschiedenen Datenformaten umgesetzt werden, am häufigsten jedoch mit JSON, oft auch mit XML.

Werden bestimmte weitere Regeln befolgt, wird von RESTful Services gesprochen. Oft werden nicht alle Regeln berücksichtigt, so dass von RESTbasiert gesprochen wird. Genaueres dazu in anderen Vorlesungen (verteilte Systeme, Software Architektur).

Frage: Woher kommen eigentlich Fehlermeldungen wie

WARNUNG: Exception thrown by a TimerTask
`java.lang.reflect.InaccessibleObjectException: Unable to make field int java.util.TimerTask.state accessible: module java.base does not "opens java.util" to unnamed module @548e7350`

Antwort: In Java 9 wurde das Module-System eingeführt, das fordert, dass alle Klassen in Paketen liegen. Mit einer Datei module-info.java wird festgelegt, welche selbstgeschriebenen Pakete von anderer Software genutzt werden können. Weiterhin wird festgelegt, welche Software-Pakete, typischerweise in Jar-Dateien, genutzt werden können. Dies kann als Erhöhung der Sicherheit angesehen werden, da nicht alle Klassen dadurch in anderen Projekten genutzt werden können (daher die Fehlermeldung). Da weiterhin jedes Paket nur in einem Module vorkommen darf, werden Konflikte zwischen mehrfach im Projekt aus Bibliotheken geladenen Klassen erkannt. Dies Zugriffeinschränkungen haben auch viele Konsequenzen für Bibliotheken, die den Byte-Code von Klassen verändern wollen, wie es bei Mockito und vielen anderen Projekten der Fall ist. Da nicht alle Projekte umgestellt werden können, wird teilweise wieder auf das Module-System verzichtet. Eine etwas detaillierte Einführung gibt es am Anfang von https://youtu.be/mr_WBWhS-wk. Der Kompromiss in der Vorlesung ist es, mit den Meldungen „zu leben“, solange es keine Einschränkungen der gewünschten Funktionalität gibt.