Gesamt überschrift

Teil 1: Projekt mit einer neuen Dependency (Hier LOG4J)

|  |  |
| --- | --- |
| **Explanation** | **Screenshot** |
| 1. Rechtsklick in den Project Explorer   ⭢ New  ⭢ Other… |  |
| 1. Auswahl im Wizzard: Maven Project   ⭢ Next |  |
| 1. Im nächten Schritt „Create a simple project (skip archetype selection)“ anwählen   ⭢ Next |  |
| 1. Group ID setzen: “edu.thi.demo”   Artifact ID und Namen auf den selben Wert setzen  Parent Project Sektion leer lassen.  Optional eine Beschreibung Einfügen.  ⭢ Finish |  |
| 1. Doppelklick auf pom.xml   Bei den Tabs im unteren Bereich des Fensters auf Dependencies wechseln  **Anmerkung**: Dieser Abschnitt tendiert dazu visuelle glitches zu verursachen. Wenn nötig kurz zurück auf Overview wechseln, dann zurück auf Dependencies |  |
| 1. Im linken Dependencies Abschnitt auf „Add“ klicken, es öffnet sich folgender Dialog |  |
| 1. In dem Suchfenster „log4j“ eingeben, nicht mit Enter bestätigen sondern einfach kurz warten |  |
| 1. log4j mit einfachem klick auswählen und nachdem die Text Felder automatisch befüllt wurden auf ok drücken |  |
| 1. Möglicherweise tritt jetzt wieder ein visueller Glitch auf, welcher den im nächsten Schritt benötigten „Add“ Button verdeckt: Kurz auf Overview, dann zurück auf Dependencies wechseln |  |
| 1. Nun rechts unter Dependency Management auf Add drücken, es öffnet sich wieder ein ähnlicher Dialog |  |
| 1. Wie oben log4j über die Suche auswählen   ⭢ ok |  |
| 1. In den Package Explorer wechseln und Rechtsklick auf src/main/java |  |
| 1. ⭢ New   ⭢ Class |  |
| 1. Als Namen LoggingProgram eingeben   ⭢ Finish |  |
| 1. Code mit dem Inhalt von LoggingProgram.java aus Moodle ersetzen   **Anmerkung**: Nicht in Panik geraten wenn IntelliJ die Imports noch nicht kennt. |  |
| 1. Strg + Shift + S drücken um alle Änderungen zu speichern!   Rechtsklick auf das Projekt ⭢ Run As ⭢ Maven install |  |
| 1. Im Konsolenfenster sollte nun BUILD SUCCESS erscheinen |  |
| 1. Rechtsklick auf LoggingProgram.java   ⭢ Run As  ⭢ Java Application  **Anmerkung**: Falls ein Dialog erscheint ob ungespeicherte Änderungen gespeichert werden sollen, auf Ok drücken und dann nochmal Maven install ausführen, da die ungespeicherten Änderungen nicht mitkompiliert wurden |  |
| 1. Ausgabe sollte folgendermaßen aussehen: |  |

Teil 2: Maven automatisiertes deployen

|  |  |
| --- | --- |
| **Explanation** | **Screenshot** |
| 1. Im Projekt orderhandling-wildfly-process die pom.xml öffnen und in die XML Ansicht wechseln |  |
| 1. Unter dem Tag <build> den <plugins> tag suchen und dort den Inhalt der Datei deploy\_plugin.txt einfügen |  |
| 1. Überprüfen ob die Syntax korrekt ist und der Inhalt der Datei (beginnend und endend mit dem <plugin> tag) richtig im eingefügt wurde.   **Anmerkung**: Da in unserem Projekt alles auf einer Maschiene läuft sind keine Uploads nötig. Für das simple Kopieren gibt es mehrere Möglichkeiten und verschiedene Plugins. In diesem StackOverflow Post sind einige davon aufgelistet:  <https://stackoverflow.com/questions/586202/best-practices-for-copying-files-with-maven>  Wer also stattdessen AntRun nutzen möchte, kann den Schritten dort folgen.  Eine AntRun dependency sollte bereits in der pom vorhanden sein |  |
| 1. Alles speichern mit Strg + Shift + S   Maven install durch Rechtsklick auf das Projekt  ⭢ Run As ⭢ Maven install ausführen  **Erklärung**:  Obwohl es um Deployment geht, nutzen wir nicht die Phase mvn deploy. Der Grund dafür ist, dass es sich bei deploy um das Hochladen von Dateien auf Remotes dreht. Unser Deployment ist lediglich das Kopieren der fertigen Datei in einen anderen Ordner |  |
| 1. Im Ordner   /home/lars/camunda/server/wildfly-10.1.0.Final/standalone/deployments  auf die Datei orderhandling-wildfly-process.war rechtsklicken > Eigenschaften und das Änderungsdatum beachten. Wenn das Änderungsdatum identisch mit der jetzigen Zeit ist, wurde die Datei kopiert. |  |

Teil 3: Unit Tests für das JPA Projekt

|  |  |
| --- | --- |
| **Explanation** | **Screenshot** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |