

Programmentwurf zur Vorlesung Web Engineering Backend

DHBW Stuttgart

Prof. Dr. Raymond Bimazubute

1. Implementierung einer einfachen REST API

Als Prüfungsleistung erstellen Sie ein REST API für das Backend einer Anwendung zur Vorlesungsplanung. Es soll beispielsweise möglich sein, Vorlesungstermine zu erstellen, zu modifizieren oder zu löschen. Es soll ebenfalls möglich sein, Dozenten zu erstellen oder Daten bestehender Dozenten zu modifizieren.

1.1 Einzuhaltende Konventionen und Regeln

- Das Projekt benötigt Java 11 (oder höher) zur Umsetzung
- Evtl. bereits vorhandene Klassen und Interfaces dürfen **nicht** verändert oder angepasst werden (Dies gilt zum Beispiel für die Exceptions im Package „de.digitra.uni-planer.exceptions“).
- Sämtliche Implementierungen dürfen nur im Package „de.digitra.uniplaner.controller“ gemäß den Vorgaben erstellt werden. In diesem Package sind bereits die zu implementierenden Klassen angelegt und der Code ist auskommentiert.
- Eventuell notwendige Hilfsklassen dürfen **nur** im Package „de.digitra.uniplaner.controller“ abgelegt werden
- Es dürfen **keine** Dritt-Libraries verwendet werden. Nur die Standard-Bibliotheken sind erlaubt (bzw. all die, welche bereits im pom.xml eingebunden wurden).
- Der Code soll kompilierbar sein und stabil laufen.
- Vorgaben zu Bezeichnungen, Parameter und Rückgabewerte sind genau einzuhalten.
- Die Abgabe erfolgt in Form eines Zip-Files. Der Name des Zip-Files besteht **nur** aus der Matrikelnummer und dem Suffix „.zip“

1.2 Programmierung

Für das Projekt wird ein Java JDK 11 (oder höher) benötigt. Als IDE können IntelliJ oder Eclipse verwenden.

1. Öffnen Sie das Start-Projekt.

2. Machen Sie sich mit der Struktur des Projektes vertraut.

- Sie werden womöglich Compiler-Fehler im Code finden.
- Diese verschwinden, wenn sie die Klassen sauber angelegt und implementiert haben.
- Die vorgegebenen Interfaces sind kommentiert. Aus diesen sogenannten Javadoc-Kommentaren können Sie erkennen, welche Methode zu implementieren sind.

3. Implementieren Sie notwendigen Klassen

4. Prüfen Sie, ob Ihre Implementierungen richtig sind (Optionaler Schritt). Sie können dazu zum Beispiel eine Startklasse (Name: App.java) mit einer Main-Methode im Package impl erstellen und Ihre Implementierung damit testen.

5. Nach Fertigstellung exportieren Sie das gesamte Projekt in ein zip-File und laden in Moodle hoch. Achten Sie bitte darauf, dass der Name ihres zip-Files **nur** aus ihrer Matrikelnummer und dem Suffix `.zip` besteht. Wenn ihre Matrikelnummer 123456789 ist, so lautet der Name des zip-Files: **123456789.zip**. Achten sie auch darauf, dass nur relevante Dateien im zip-file liegen und nicht z.B. Binaries aus dem target-Ordner oder z.B. bin-Ordner.

1.3 Interfaces

Folgende Interfaces sind im Start-Projekt bereits vorbereitet. Die Interfaces dürfen nicht modifiziert werden. **Diese Interfaces enthalten die Vorgaben für die zu implementierenden Klassen im Package `„de.digitra.uniplaner.controller“`**

Interface	Beschreibung
ILectureController	Das Interface beschreibt einen REST Controller für die Handhabung von Ressourcen des Typs Lecture.
ILectureDateController	Das Interface beschreibt einen REST Controller für die Handhabung von Ressourcen des Typs LectureDate.
ILecturerController	Das Interface beschreibt einen REST Controller für die Handhabung von Ressourcen des Typs Lecturer
ISemesterController	Das Interface beschreibt einen REST Controller für die Handhabung von Ressourcen des Typs Semester.
IStudyClassController	Das Interface beschreibt einen REST Controller für die Handhabung von Ressourcen des Typs StudyClass.
IStudyProgramController	Das Interface beschreibt einen REST Controller für die Handhabung von Ressourcen des Typs StudyProgramController.

1.4 Implementierungsklassen

Folgende Implementierungsklassen sollen programmiert werden:

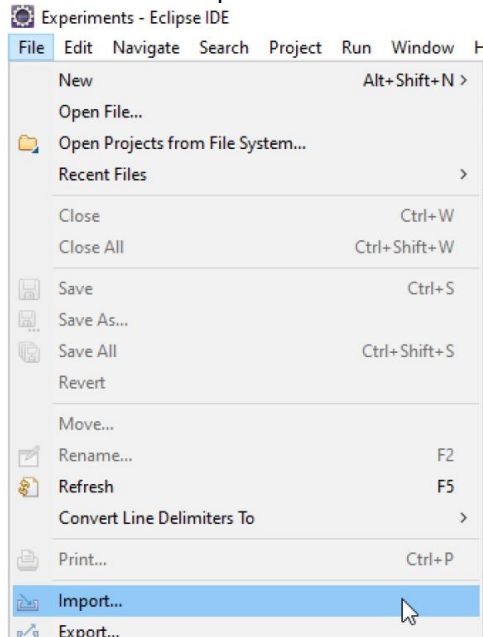
- LectureController
- LectureDateController
- LecturerController
- SemesterController
- StudyClassController
- StudyProgramController

Hinweise:

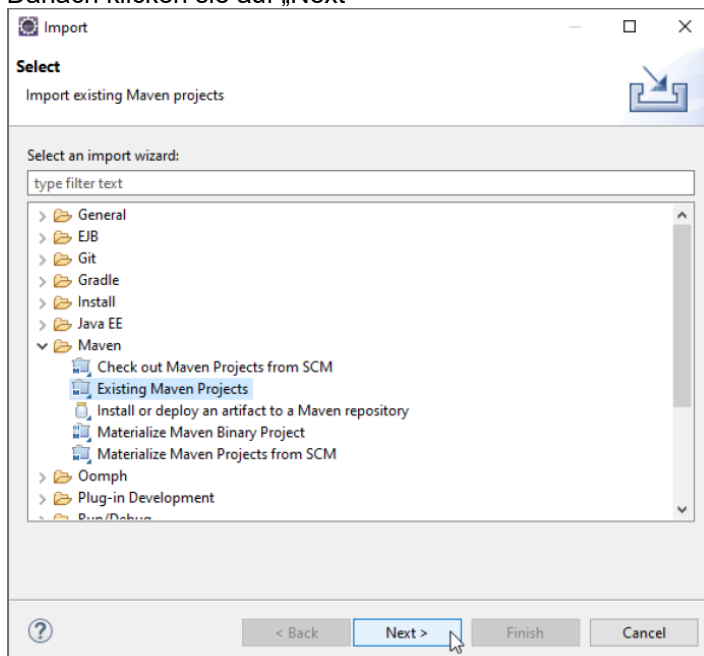
- Lesen sie die Javadoc Kommentare an den entsprechenden Interfaces genau durch
- Nutzen sie die Javadoc für Java-Klassen: <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>

2 Projekt öffnen mit Eclipse

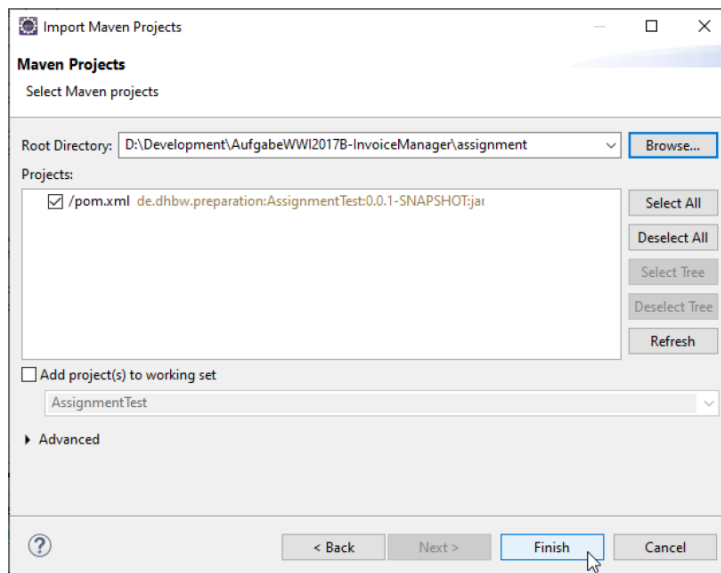
1. Entpacken sie Project-ZIP File in ein beliebiges leeres Verzeichnis
2. Starten sie ihre Eclipse IDE
3. Öffnen sie den Import-Assistenten in ihrer Eclipse IDE



4. Im Wizard öffnen sie nun den Unterpunkt „Maven“ und selektieren „Existing Maven Projects“. Danach klicken sie auf „Next >“



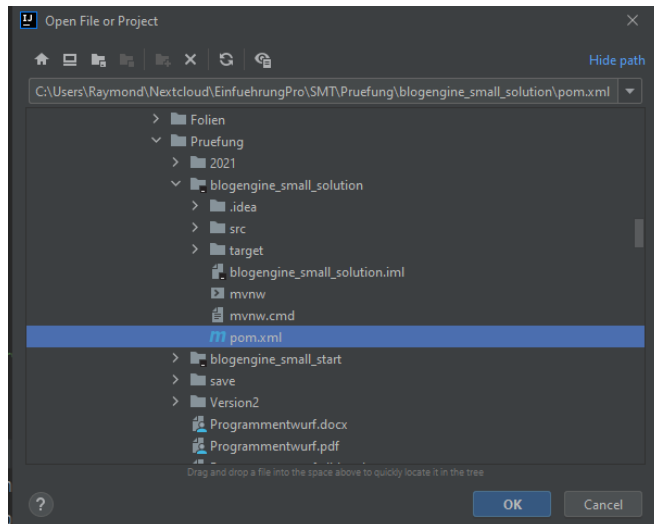
5. Wählen sie nun das Root-Verzeichnis (Verzeichnis, in das sie das ZIP entpackt haben), selektieren sie das gefundene Projekt und klicken sie auf „Finish“.



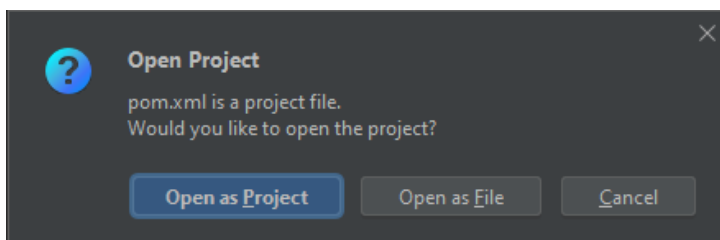
6. Das importierte Projekt wird nun im Projektextplorer angezeigt.

3 Projekt öffnen mit IntelliJ

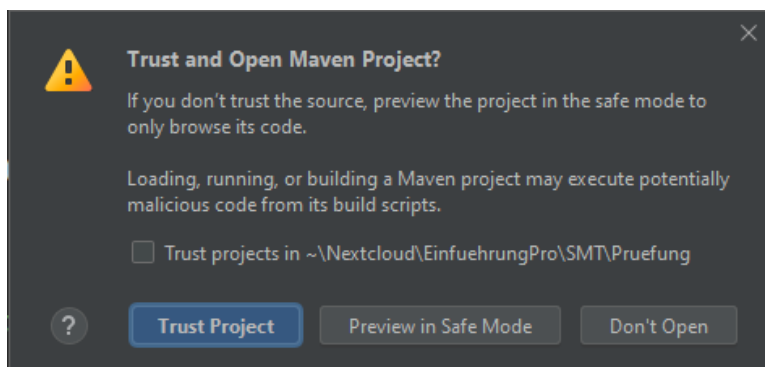
1. Entpacken sie Project-ZIP File in ein beliebiges leeres Verzeichnis
2. Starten sie IntelliJ
3. Wählen Sie im IntelliJ-Menü den Punkt **File->Open**
4. Selektieren Sie das Verzeichnis, in dem Sie die Zip-Datei des Start-Projektes entpackt haben.
5. Wählen Sie die Datei „pom.xml“ und klicken Sie auf OK.



6. Wählen Sie in dem folgenden Dialog die Option „**Open as Project**“.

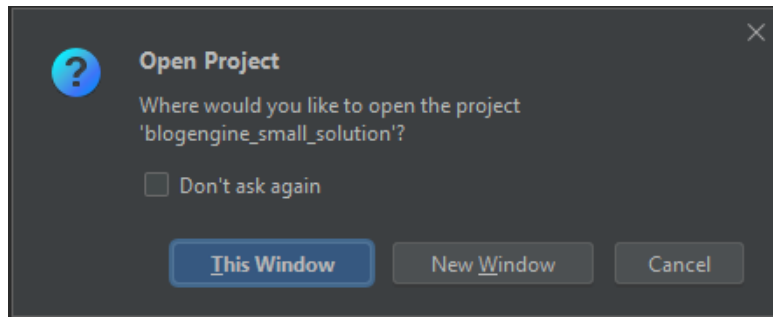


7. Wählen Sie in dem folgenden Dialog die Option „**Trust Project**“.



8.

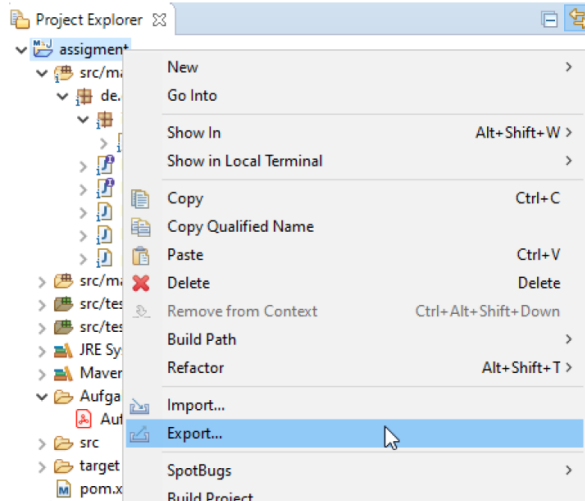
9. Wählen Sie in dem folgenden Dialog die Option „**New Window**“.



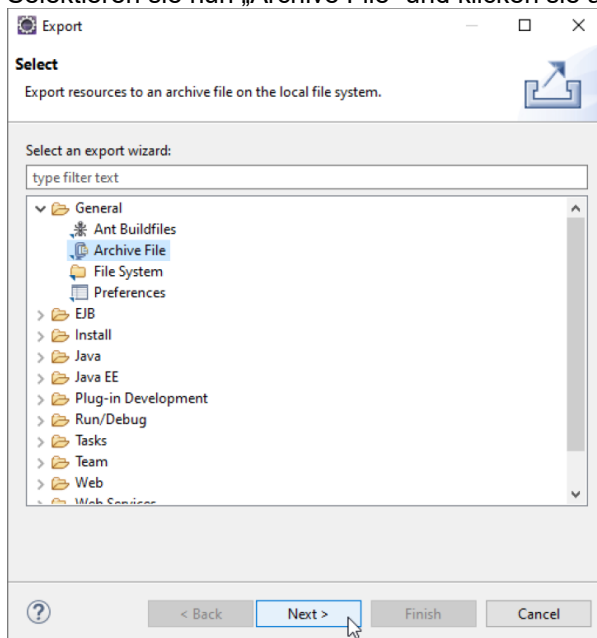
4 Eclipse: Projekt exportieren und als Zip-Datei speichern

Sobald Sie mit dem Programm fertig sind, müssen sie es für die Abgabe vorbereiten. Die Abgabe erfolgt in Form eines ZIP-Files, welches in Moodle hochgeladen werden muss.

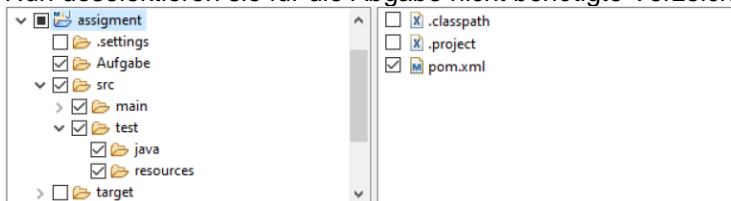
1. Klicken sie mit der rechten Maustaste auf das Projekt im Project Explorer.
2. Im sich öffnenden Kontextmenu klicken sie auf den Menüpunkt „Export...“.



3. Im sich öffnenden Wizard klappen sie den Punkt „General“ auf.
4. Selektieren sie nun „Archive File“ und klicken sie auf „Next >“



5. Im nächsten Fenster wählen sie das zu exportierende Projekt aus.
6. Nun deselektieren sie für die Abgabe nicht benötigte Verzeichnisse und Dateien.



Zu den nicht benötigten Verzeichnissen gehören: „.settings“, „target“ und Verzeichnisse unterhalb „src/test/java“.

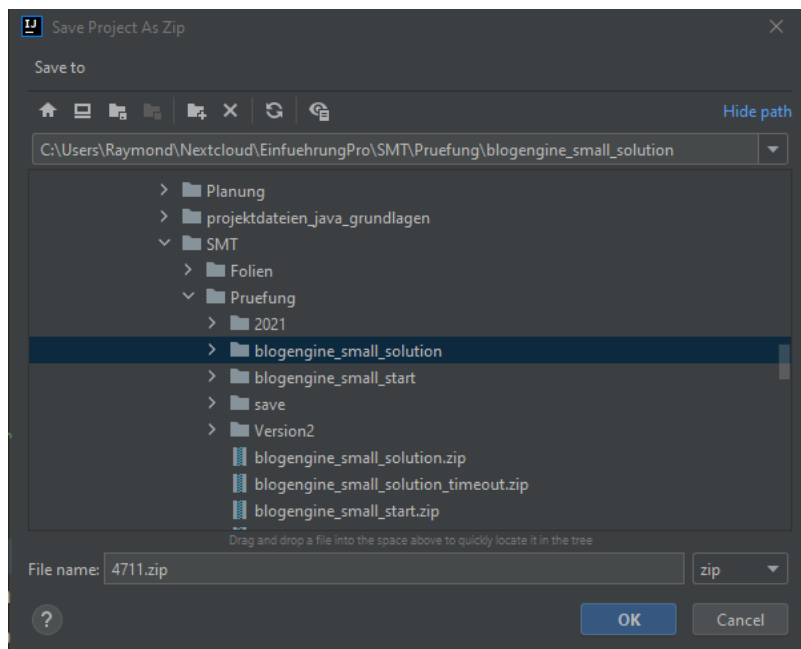
Nicht benötigte Files sind: „.classpath“ und „.project“.

7. Nun legen sie den Export-Pfad fest. Unter „Options“ selektieren sie „Dave in zip format“ und „Create directory structure for files“
8. Nun klicken sie auf „Finish“.
9. Prüfen sie noch einmal das entstandene ZIP
 - Richtige Bezeichnung (<Matrikelnummer>.zip)
 - Sind alle notwendigen Files enthalten?

5 IntelliJ: Projekt exportieren und als Zip-Datei speichern

Sobald Sie mit dem Programm fertig sind, müssen sie es für die Abgabe vorbereiten. Die Abgabe erfolgt in Form eines ZIP-Files, welches in Moodle hochgeladen werden muss

1. Wählen Sie den in der Menü-Leiste von IntelliJ den Menü-Punkt **File -> Export -> Project to Zip File**
2. Wählen Sie das Verzeichnis aus, wohin Sie das exportierte Projekt speichern wollen.
3. Geben Sie den Dateinamen an (Format: Matrikelnummer.zip, z.B. 4711.zip).



4. Nun klicken sie auf „OK“.
5. Prüfen sie noch einmal die entstandene Zip-Datei.
 - Richtige Bezeichnung (<Matrikelnummer>.zip)
 - Sind alle notwendigen Dateien enthalten?