

Assignments/Quizzes

Assignments

id : Long
title: String
description: String (TEXT in db)
dueDate: Date
course : Course

Course/Assignment Relation

Course
List<Assignment>

Assignment
Course

Student/Assignment Relation

```
@Entity
public class Student

    @OneToMany(mappedBy = "student", cascade = CascadeType.ALL)
    private Set<AssignmentSubmission> assignments;

    // ...
}

@Entity
public class Assignment {

    @OneToMany(mappedBy = "assignment", cascade = CascadeType.ALL)
    private Set<AssignmentSubmission> students ;
```

```

        // ...
    }

    public class AssignmentSubmission{

        @ManyToOne
        @MapsId("studentId")
        private Student student;

        @ManyToOne
        @MapsId("assignmentId")
        private Assignment assignment;

        private Date submittedDate;

        private String url ; // url of the submitted file
        // ...

        private Integer grade ; // null if not graded

    }

```

Endpoints

```

Post 'localhost:8080/courses/{courseId}/assignments/
    // For instructor to create an assignment (title, content,
Put 'localhost:8080/courses/{courseId}/assignments/{aId}
    // For instructor to update an assignment (title, content,
Get 'localhost:8080/courses/{courseId}/assignments/{aId}
    // For students to get an assignment
    // (title, content, uploadDate, dueDate)
POST 'localhost:8080/courses/{courseId}/assignments/{aId}/submit
    // for students to submit an assignment
    // (upload a file)
Get 'localhost:8080/courses/{courseId}/assignments/{aId}/submit

```

```
// for instructor to get all students submission

Post 'localhost:8080/courses/{courseId}/assignments/{aId}/grade,
    // for instructor to grade quiz
Get 'localhost:8080/courses/{courseId}/assignments/{aId}/grade/.
    // for student to get his grade
```

Quizzes

NOTE

1. QUIZZES ARE GENERATED RANDOMLY FOR EACH STUDENT FROM THE QUESTION BANK OF THE COURSE
2. QUIZZES are graded automatically (Questions have Choices).

id : Long
title: String
numOfQues : Integer
Date startDate
Date endDate

Course/Quiz Relation

Course
List<Quiz>

Quiz
Course

Student/Quiz Relation

```
@Entity
public class Student

    @OneToMany(mappedBy = "student", cascade = CascadeType.ALL)
    private Set<QuizSubmission> quizzes;
```

```

        // ...
    }

@Entity
public class Quiz{

    @OneToMany(mappedBy = "quiz", cascade = CascadeType.ALL)
    private Set<QuizSubmisson> students ;

    // ...
}

public class QuizSubmisson{

    @ManyToOne
    @MapsId("studentId")
    private Student student;

    @ManyToOne
    @MapsId("quizId")
    private Quiz quiz;

    Set<Question> questions;

    private int grade ;

}

```

Endpoints

```

Post 'localhost:8080/courses/{courseId}/quizzes/
    // For instructor to create an assignment (title, numOfQue:
Get 'localhost:8080/courses/{courseId}/quizzes/{qId}
    // For students to get an assignment

```

```
// (his questions)
// questions are randomized the first time
// if he enters again he gets the same questions (check his
POST 'localhost:8080/courses/{courseId}/quizzes/{qId}/submit
// for students to submit quizzes answers
// { {questionId,choiceId},{questionId,choiceId}, ... }
Get 'localhost:8080/courses/{courseId}/quizzes/{qId}/submit
// for instructor to get all students submissions
Get 'localhost:8080/courses/{courseId}/assignments/{aId}/grade/
// for student to get his grade
```