REST Api

(Token is sent with a cookie now. Browser should send it automatically)

עמוד ראשי

**GET** List of teacher’s courses:

Teacher/CourseList **| Return**: List<Course>

[

    {

        "id": "89115\_2021",

        "number": 89115,

        "name": "מבנה מחשב",

        "year": 2021,

        "semester": 0

    }

]

**GET** exercises by order in which they were created

Teacher/ExerciseList

Returns:

[

    {

        "exID": "89111\_2021\_ex2",

        "exName": "ex2",

        "courseName": "תכנות מתקדם 1",

        "courseNumber": 89111,

        "courseID": "89111\_2021"

    },

    {

        "exID": "89111\_2021\_ex1",

        "exName": "ex1",

        "courseName": "תכנות מתקדם 1",

        "courseNumber": 89111,

        "courseID": "89111\_2021"

    }

]

**GET** Extension requests for exercises in the teachers courses:

Teacher/GetExtensionRequestsMainPage

Example:

[

    {

        "courseID": "89111\_2021",

        "courseName": "תכנות מתקדם 1",

        "courseNumber": 89111,

        "exerciseID": "89111\_2021\_ex1",

        "exerciseName": "ex1",

        "chatID": "A576888433\_89111\_2021\_ex1",

        "submissionID": "576888433\_89111\_2021\_ex1",

        "studentName": "יוסי יוסי",

        "type": 0

    }

]

**GET** Extension requests for exercises in the teachers courses:

Teacher/GetAppealsMainPage

[

    {

        "courseID": "89111\_2021",

        "courseName": "תכנות מתקדם 1",

        "courseNumber": 89111,

        "exerciseID": "89111\_2021\_ex1",

        "exerciseName": "ex1",

        "chatID": "B576888433\_89111\_2021\_ex1",

        "submissionID": "576888433\_89111\_2021\_ex1",

        "studentName": "יוסי יוסי",

        "type": 1

    }

]

**Course**:

**GET** all exercises from all years:

Teacher/AllExercises?course=(string) | Return:

Dictionary: Key – year (as string) , Value - List<ExerciseLabel> Descending order by year

Example result:

{

    "2021": [

        {

            "id": "89111\_2021\_ex2",

            "name": "ex2"

        },

        {

            "id": "89111\_2021\_ex1",

            "name": "ex1"

        }

    ]

"2020": [

        {

            "id": "89111\_2020\_ex2",

            "name": "ex2"

        },

        {

            "id": "89111\_2020\_ex1",

            "name": "ex1"

        }

    ]

}

**GET** all teachers for the course

Teacher/GetTeachers?course=(string) |

Return:

[

    {

        "id": "576888433",

        "name": "יוסי יוסי"

    }

]

**GET** all checkers for the course

Teacher/GetCheckers?course=(string) |

Return:

[

    {

        "id": "149675111",

        "name": "עמית עמית"

    }

]

**POST**: Add Checker to course:

Teacher/AddCourseChecker?course=(string)&checkerId=(string)

Returns: Checker name

**POST**: Add Teacher to course:

Teacher/AddTeacher?course=(string)&teacherId=(string)

Returns: Teacher name

**DELETE** Checker from course:

Teacher/RemoveCourseChecker?course=(string)&checkerId=(string)

**Exercise:**

**GET** information of exercise

Teacher/ExerciseDetails?exerciseId=(string) | Return Exercise

Example result:

{

    "id": "89111\_2021\_ex1",

    "name": "ex1",

    "maxSubmitters": 1,

    "reductions": [

        10,

        20

    ],

    "programmingLanguage": "python",

    "autoTestGradeWeight": 60,

    "styleTestGradeWeight": 20,

    "isActive": 1,

    "multipleSubmission": **true**

    "mossShownMatches": 250,

# number of times a piece of code in Moss until it no longer counts.

    "mossMaxTimesMatch": 10,

    "mossLink": "",

# help files names

    "filenames": [

        "ex2.zip"

    ]

}

**GET**: Downloads a Help file

Teacher/GetHelpFile?exerciseId=(string)&file=(string)

**POST**: Create exercise  
Teacher/CreateExercise?courseId=(string)

{

# Optional. Main date can still be added separately.

"MainDate": "2021-11-11T00:00:00",

"Exercise": {

        "name": "ex12",

        "maxSubmitters": 1,

        "reductions": [

            10,

            20

        ],

        "programmingLanguage": "python",

        "autoTestGradeWeight": 60,

        "styleTestGradeWeight": 20,

        "isActive": 1,

        "multipleSubmission": **true**

    },

# Files to help solve the exercise. Optional.

"HelpFiles": [

     {

        "Name" : "ex2.zip",

        "Content" : "UEsDBBQAAAAIA…"

    }

]

}

**PUT:** Update an exercise (cannot change exercise name):

Teacher/UpdateExercise.

Input:

{

        "id": "89111\_2021\_ex1",

        "name": "ex1",

        "maxSubmitters": 1,

        "reductions": [

            15,

            20

        ],

        "programmingLanguage": "python",

        "autoTestGradeWeight": 70,

        "styleTestGradeWeight": 30,

        "isActive": 1,

        "multipleSubmission": **true**

}

Copy an exercise. All the exercises properties and all files and tests are copied:

Teacher/CopyExercise?courseId=(string)&oldExerciseId=(string

Body: name for the new exercise

Returns: ID for new exercise (string)

**POST**: Add Checker to exercise:

Teacher/AddExChecker?exercise=(string)&checkerId=(string)

**POST**: Add Help Files

Teacher/AddHelpFiles?exerciseId=(string)

Body example:

[

     {

        "Name" : "ex2.zip",

        "Content" : "UEsDBBQAAAAIA…"

    }

]

**DELETE** exercise (exercise can’t be deleted if students have already submitted):

Teacher/DeleteExercise?exerciseId=(string)

**DELETE** Checker from exercise:

Teacher/RemoveExChecker?exercise=(string)&checkerId=(string)

**DELETE** HelpFiles:

Teacher/DeleteHelpFile?exercise=(string)&file=(string)

Dates:

**GET** exercise Dates:

Teacher/GetDates?exerciseId=(string)

Example output:

[

    {

        "id": 2,

        "date": "2021-08-20T00:00:00",

        "group": 0,

        "reduction": 0,

        "submitters": **null**

    },

    {

        "id": 1,

        "date": "2021-10-28T00:00:00",

        "group": -1,

        "reduction": 10,

        "submitters": [

            {

                "id": "576888433",

                "name": "יוסי יוסי"

            }

        ]

    }

]

**POST:** Add Date. Cannot add a date if there’s already a date with the same group number

Also cannot add a reduction to the default date (group number 0)

Teacher/AddDate?exerciseId=(string)

Example input:

    {

        "date": "2021-08-20T00:00:00",

        "group": 0,

        "reduction": 0

    }

**PUT:** Update Date. cannot change a group number or add a reduction to the default date (group number 0)

Teacher/UpdateDate

Example input:

    {

        "id": 1014,

        "date": "2021-09-28T00:00:00",

        "reduction": 0

    }

**DELETE** Date. cannot delete default date (group number 0)

Teacher/DelteDate?dateId=(int)

Copy check:

**POST:** perform a copying check using Moss.

Teacher/CopyCheck

Input:

{

    "ExerciseID": "89111\_2021\_ex1",

# number of matches shown in the moss check

    "MaxFound": 250,

# number of times a code snippet can be found until it no longer counts as copying

    "MaxMatchs": 10

}

Returns link to results. Example: <http://moss.stanford.edu/results/2/846219715635/>

Marks two students for copying. The IDs of the students are the folder names in the moss results.

/Teacher/SetCopied

{

    "ExerciseID": "89111\_2021\_ex1",

    "User1": "576888444",

    "User2": "576888433"

}

Requests:

Gets all deadline extension requests for the exercise:

Teacher/GetExtensions?exerciseId=(string)

Example output:

[

    {

        "chatID": "A576888433\_89111\_2021\_ex1",

        "studentID": "576888433",

        "studentName": "יוסי יוסי",

        "message": "Hello there",

        "type": 0,

        "state": 1,

        "isClosed": **true**

    }

]

Gets all deadline extension requests for the exercise:

Teacher/GetAppeals?exerciseId=(string)

Example output:

[

    {

        "chatID": "A576888433\_89111\_2021\_ex1",

        "studentID": "576888433",

        "studentName": "יוסי יוסי",

        "message": "Hello there",

        "type": 1,

        "state": 0,

        "isClosed": **false**

    }

]

**GET** Messages of chat:

Teacher/MessageList?chatId=(int)

Example output:

[

    {

        "id": 1,

        "senderID": "576888433",

        "senderName": "יוסי יוסי",

        "date": "2021-08-08T23:26:25.243",

        "body": "Hello there",

        "chatID": "B576888433\_89111\_2021\_ex1",

        "status": 0,

        "filename": "a.txt",

        "isTeacher": **false**

    }

]

**GET:** download attached file of message:

/Teacher/MessageFile?messageId=1

**POST:** Add message to chat:

/Teacher/NewMessage?chatId={{ch1}}

Example input:

{

    "Text": "Hello there",

    "AttachedFile": {

        "Name": "a.txt",

        "Content": "aGVsbG8="

    }

}

OR

“Hello there”

Reject the Extension Request or Appeal:

Teacher/RejectRequest?chatId=(int)

Accept The Extension Request. Needs to be sent with a new date and point reduction (optional):

Teacher/AcceptExtension?chatId=(int)

Body:

{

    "Date": "2021-10-28T00:00:00",

    "Reduction": 0

}

Accept The Appeal:

Teacher/AcceptAppeal?chatId=(int)

Tests:

**GET** tests:

Teacher/ExerciseTests?exerciseId=(string)

Example output:

{

        "id": 4,

        "type": 0,

        "exerciseID": "89111\_2021\_ex1",

        "weight": 100,

        "input": "",

        "expectedOutput": "hello there\n",

        "outputFileName": "stdout",

        "argumentsString": "50",

        "timeoutInSeconds": 60,

        "mainSourseFile": "ex1.py",

        "additionFileNames": [

            "a.txt"

        ],

        "has\_Adittional\_Files": **true**

}

**POST:** Add a test:

Teacher/ExerciseTests

{"Test": {

        "ExerciseID": "*89111\_2021\_ex1*",

        "Weight": 100,

        "Input": "",

        "ExpectedOutput": "hello there\n",

        "OutputFileName": "stdout",

        "ArgumentsString": "50",

        "TimeoutInSeconds": 60,

        "MainSourseFile": "ex1.py"

    },

    "AdditionalFiles": [

        {

            "Name": "a.txt",

            "content": "Z3JlZXRpbmdz"

        }

    ]

}

**DELETE** test:

Teacher/DeleteTest?testId=(int)

**PUT:** update test

Teacher/UpdateTest

{

        "ID": 3,

        "ExerciseID": “89111\_2021\_ex1",

        "Weight": 100,

        "Input": "",

        "ExpectedOutput": "hello there\n",

        "OutputFileName": "stdout",

        "ArgumentsString": "50",

        "TimeoutInSeconds": 70,

        "MainSourseFile": "ex1.py"

}

**POST:** Add files for testing submissions. Zip files get unzipped automatically.

Teacher/AddRunFiles?testId=(int)

Example input:

[

     {

        "Name" : "ex2.zip",

        "Content" : "UEsDBBQAAAAIA…"

    }

]

**DELETE:** delete Runfile

Teacher/DeleteRunFile?testId=int&file=(string)

**GET** Runfile:

Teacher/GetRunFile?testId=(int)&file=(string)

**GET:** Download all runfiels as zip file

Teacher/DownloadRunFiles?testId=(int)

**GET** submission labels. Sorted by CheckState.

Teacher/SubmissionLabels?exerciseId

{

    "Unchecked": [

        {

            "id": "576888433\_89111\_2021\_ex1",

            "state": 1,

            "submitters": [

                {

                    "id": "576888433",

                    "name": "יוסי יוסי"

                }

            ],

# 0 – you are currently checking this exercise

# 1 – this submission is not being checked

# 2 – this exercise is being checked by someone else

            "checkState": 1

        }

    ],

    "Checked": [],

    "Appeal": []

}

**GET student** submission.

Teacher/SubmissionDetails?submissionId=(string)

Example outout:

{

    "id": "576888433\_89111\_2021\_ex1",

    "filenames": [

        "ex1.py"

    ],

    "autoGrade": -1,

    "styleGrade": 100,

    "manualFinalGrade": -1,

    "manualCheckData": "",

    "submissionStatus": 1,

    "timeSubmitted": "2021-08-08T23:59:17.067",

    "hasCopied": **false**,

    "checkState": 1,

    "totalGrade": -1

}

**GET** automatic check results for submission. Also marks you as the current checker of this exercise.

Teacher/RunResult?submissionId=(string)

Example output:

{

# a summary of the results in html format

    "text": "<u><h3>Automatic Test:</h3></u><br><b>Total grade: 100 | Total run time: 59.2758ms</b><br><hr><br>Test: Grade: 100/100 | Run Time: 59.2758ms<br><u><h5>Input:</h5></u><br><br><u><h5>Expected output:</h5></u><br>The number is 50\*50=2500<br><u><h5>Your output:</h5></u><br>The number is 50\*50=2500<br><hr><br><u><h3>Coding Style Test:</h3></u><br>Grade: 100<br>",

    "autoGrade": 100,

    "styleGrade": 100

}

**POST:** Finishes checking exercise.

Teacher/EndChecking

Example input:

{

    "ID": "576888433\_89111\_2021\_ex1",

    "AutoGrade": 100,

    "ManualGrade": 100,

    "ManualCheckData": "?"

}