REST Api

סטודנט: GET

**GET** List of courses Student is enrolled in

Student/CourseList

Example output:

[

    {

        "id": "89112\_2021",

        "number": 89112,

        "name": "מבוא לרשתות תקשורת",

        "year": 2021,

        "semester": 0

    }

]

**GET** List of due exercises

Student/ExerciseList

[

    {

        "id": "89111\_2021\_ex2",

        "name": "ex2",

        "date": "2021-08-22T00:00:00",

        "courseName": "תכנות מתקדם 1",

        "courseNumber": 89111,

        "courseID": "89111\_2021",

        "teacherName": **null**

    },

**GET** List of graded exercises

Student/GradesList

Example output:

[

    {

        "exID": "89111\_2021\_ex1",

        "exName": "ex1",

        "grade": 100,

        "courseName": "תכנות מתקדם 1",

        "courseNumber": 89111,

        "courseID": "89111\_2021"

    }

]

**GET** Labels of exercises of the course, in descending order by creation

Student/ExerciseLabels?coursed=(string)

Example output:

[

    {

        "id": "89111\_2021\_ex3",

        "name": "ex3"

    }

]

**GET** Exercise information:

Student/SubmissionDetails?exerciseId=(string)

"totalGrade": -1,

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

    "extensionChat": {

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

        "type": 0,

        "isClosed": **false**,

        "state": 0

    },

    "maxSubmitters": 1,

    "submitters": [

        {

            "id": "576888433",

            "name": "יוסי יוסי"

        }

    ],

    "exID": "89111\_2021\_ex1",

    "exName": "ex1",

    "dateSubmitted": "2021-08-09T05:22:19.443",

    "dates": [

        {

            "date": "2021-08-22T00:00:00",

            "id": 0,

            "exerciseID": **null**,

            "group": 0,

            "reduction": 0

        }

    ],

    "isMultipleSubmission": **true**,

    "manualCheckInfo": "",

    "filenames": [

        "ex1.py"

    ],

    "state": 1,

    "isMainSubmitter": **true**,

    "styleGrade": 100,

    "autoGrade": -1,

    "manualGrade": -1,

    "styleWeight": 30,

    "autoWeight": 70,

    "hasCopied": **false**,

    "reductions": [ 15, 20 ],

    "maxLateDays": 2,

    "timeSubmitted": "0001-01-01T00:00:00"

**GET** Message list for chat

Student/MessageList?token=(string)&chatId=(string) | Return: List<Message<

[

    {

        "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** Message attached file

Student/MessageFile?messageId=(int)

**GET** result of automatic check for submission

Student/RunResult?submitId=(string) | Return: String

{

    "text": "<u><h3>Automatic Test:</h3></u><br><b>Total grade: 100 | Total run time: 65.3978ms</b><br><hr><br>Test: Grade: 100/100 | Run Time: 65.3978ms<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

}

**GET** file download

Download submission as ZIP

Student/Download?submissionId=(string)

Return: ZIP

**GET** file from Submission:

Student/GetFile?token=(string)&submissionId=(string)

Return: File

סטודנט: POST

**POST** submit an appeal

Teacher/Appeal?submissionId=(string)

Example input:

{

    "Text": "אני רוצה לערער",

    "AttachedFile": {

        "Name": "a.txt",

        "Content": "aGVsbG8="

    }

}

**POST** submit an extension request

Teacher/ExensionRequest?submissionId=(string)

Example input:

{

    "Text": "אני רוצה לערער",

    "AttachedFile": {

        "Name": "a.txt",

        "Content": "aGVsbG8="

    }

}

**POST** new message

Student/NewMessage?chatId=(string)

Example input:

{

    "Text": "Hello there",

    "AttachedFile": {

        "Name": "a.txt",

        "Content": "aGVsbG8="

    }

}

OR

“hello there”

Returns message ID

Submit an exercise

Student/SubmitExercise?exerciseId=(string)&final=(bool)

Input example:

[

    {

        "Name" : "ex1.py",

        "Content" : "aW1wb3J0IHN5c…”

    }

]

Output example:

{

    "message": "<u><h3>Automatic Test:</h3></u><br><b>Total grade: 100 | Total run time: 123.1582ms</b><br><hr><br>Test: Grade: 100/100 | Run Time: 123.1582ms<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>",

    "files": [

        "ex1.py"

    ],

    "autoGrade": 100,

    "styleGrade": 100

}

Checks IDs of submitting students. If all IDs are valid, adds the students to the submission

Student/ValidateSubmitters?exerciseId=(string)

Input example:

[

    "132123132",

    "432543654"

]

Return: Dictionary<(Student ID), (Bool. True if student ID is valid)>

Detach from Submission

Student/DetachSubmission?submissionId=(string)

Cancel submission

Student/CancelSubmission?submissionId=(string)