Backend Documentation WIP

Ngrok handles HTTP/TCP tunnels on random URLs/ports so I don't need to mess with settings on my home router to open ports.
Local Url = http://0.0.0.0:23567
Ngrok Url = https://63d0e20e8300.ngrok.io/
The ngrok Url will change every time I launch a demo session (limitation of free trial), need to be able to edit our programs to quickly change this Url

I used the Kotlin Ktor framework to handle routing calls in my API micro service and I used the Kotlin Exposed ORM (Object Relational Mapping) framework to handle DB transactions with our postgreSQL

We have 3 different DB tables, these are:

studentuser

lecture

module

The Student User Table contains the following columns:

uuid:
- Type: UUID
- Desc: A unique ID for the entry row
- Example: "3b6440a2-7ee1-48aa-8d47-bc3af59a08cd"
- Primary Key

first_name:
- Type: String
- Example: "Todd"

last_name:
- Type: String
- Example: "Test"

student_id:
- Type: String
- Desc: The student number
- Example: "12345678"
- Unique Index

course _code:
- Type: String
- Desc: The student's course code
- Example: "CASE4"

module_codes:
- Type: String
- Desc: The modules the student is enrolled in (this is a hack solution, discussed below)
- Example: “CA4004, CA4007, CA4009”

Comments:
module_codes is a list that has been transformed into a string, while postgreSQL supports storing arrays, Kotlins Exposed frameWork does not. Storing an array as a string and converting it back and forward is a hack solution and may have scaling issues. Further changes may come to this now that the module table has created.

Student numbers are unique, however there is currently no limit on what can make up the string, eg “abc” is a valid student number

The Lecture Table contains the following columns:

uuid:
- Type: UUID
- Desc: A unique ID for the entry row
- Example: "3b6440a2-7ee1-48aa-8d47-bc3af59a08cd"
- Primary Key

course_code:
- Type: String
- Example: “CASE4”

module_code:
- Type: String
- Example: “CA4004”

date:
- Type: DateTime
- Desc: This is the start time for the lecture
- Example: “2021-09-11T09:00:00.000Z”

end_time:
- Type: DateTime
- Desc: This is the end time for the lecture
- Example: “2021-09-11T11:00:00.000Z”

location:
- Type: String
- Desc: Room were the lecture will take place
- Example: “GLA.L104”

type:
- Type: String
- Desc: the lecture type
- Example: "lecture", "lab", "tutorial"

expectedAttendanceNumber:

Type: Integer
Desc: The number of students enrolled in the module
Example: 100

actualAttendanceNumber:
Type: Integer
Desc: The number of students who arrived at the lecture
Example: 67

expectedAttendance:
Type: String
Desc: The student ids as string in a list,of students enrolled in the module
Example: “12345678, 87654321, ...”

expectedAttendance:
Type: String
Desc: The student ids as string in a list, of students who arrived at the lecture
Example: “12345678, ...”

Comments:
As with moduleCodes in the student user table, both expectedAttendance and expectedAttendance as strings of a complete kotlin list that gets transform back into a List<String> in the code, but can’t be stored like that

The Module Table contains the following columns:

uuid:
Type: UUID
Desc: A unique ID for the entry row
Example: "3b6440a2-7ee1-48aa-8d47-bc3af59a08cd"
Primary Key

module_code:
Type: String
Example: “CA4004”
UNIQUE INDEX

module_title:
Type: String
Example: “Soft. Eng.:Process,Principles & Methods”

reg_student_ids:
Type: String
Desc: the student numbers of all students enrolled, in an array list string like Lecture.moduleCodes
Example: “12345678, 87654321, ...”

expected_attendance:
Type: Integer
Desc: number of students enrolled in the module, eg reg_student_ids.size()
Example: 10

Comments:

User routes:

url = http://0.0.0.0:23567/users

Route: Get {/users}

Desc: Returns all student users in DB
Body: n/a
Returns:


```
{
	"users": [
		{
			"uuid": "92089a4d-a36a-480a-93b5-1053fa555ea0",
			"firstName": "Ronn",
			"lastName": "Test",
			"studentId": "11247913",
			"courseCode": "CASE4",
			"moduleCodes": [
				"CA4004",
				"CA4006",
				"CA4009"
			],
		},
		{
		"uuid": "043b69ca-.....
		}
	]
}
```


Route: Post {/users}

Desc: Adds a new student user to the DB
Body:


```
	{
		"firstName": "John",
		"lastName": "Test",
		"studentId": "17348912",
		"courseCode": "CASE4",
```

```
        "moduleCodes": [
            "CA4004",
            "CA4006",
            "CA4007"
        ],
    }
```
Returns: 200

Route: Put {/users}

Desc: Updates a student user to the DB, matches on UUID
Body:
```
    {
        “uuid”: “3b6440a2-7ee1-48aa-8d47-bc3af59a08cd”,
        "firstName": "John",
        "lastName": "Test",
        "studentId": "17348912",
        "courseCode": "CASE4",
        "moduleCodes": [
            "CA4004",
            "CA4006",
            "CA4007"
        ],
    }
```
Returns: 200

Route: Delete {/users/[uuid]}

Desc: Deletes a student user to the DB, matches on UUID
Body: n/a
Returns: 200 or 404 if there is no uuid match

Route: Get{/users/[uuid]}

Desc: Gets a student user to the DB, matches on UUID
Body: n/a
Returns:
```
    {
        “uuid”: “3b6440a2-7ee1-48aa-8d47-bc3af59a08cd”,
        "firstName": "John",
        "lastName": "Test",
        "studentId": "17348912",
        "courseCode": "CASE4",
        "moduleCodes": [
            "CA4004",
            "CA4006",
```

```
                "CA4007"
            ],
        }
```

Route: Get{/users/student/[studentId]}

Desc: Gets a student user to the DB, matches on Student ID/Number
Body:

```
        {
            “uuid”: “3b6440a2-7ee1-48aa-8d47-bc3af59a08cd”,
            "firstName": "John",
            "lastName": "Test",
            "studentId": "17348912",
            "courseCode": "CASE4",
            "moduleCodes": [
                "CA4004",
                "CA4006",
                "CA4007"
            ],
        }
```

Returns: 200 or error

Route: Get{/users/module/[moduleCode]}

Desc: Gets all student users reg to module
Body: n/a
Returns:

```
{
    "users": [
        {
            "uuid": "92089a4d-a36a-480a-93b5-1053fa555ea0",
            "firstName": "Ronn",
            "lastName": "Test",
            "studentId": "11247913",
            "courseCode": "CASE4",
            "moduleCodes": [
                "CA4004",
                "CA4006",
                "CA4009"
            ],
        },
        {
        "uuid": "043b69ca-.....
        }
```

```
        ]
}
```

Route: Get{/users/week-timetable/[studentId]}

Desc: Gets a student user’s lectures that occur in the current week
Body: n/a
Returns:

```
        {
                [
                        {
                        "uuid": "7f5583ab-546d-46a9-8fc6-da9898c28026",
                        "courseCode": "CASE4",
                        "moduleCode": "CA4004",
                        "date": {
                                "year": 2021,
                                "dayOfMonth": 28,
                                "dayOfWeek": 3,
                                "dayOfYear": 118,
                                …,
                                …,
                                "millis": 1619600400000,
                                "beforeNow": false,
                                "afterNow": true,
                                "equalNow": false
                        },
                        "endTime": {
                                "year": 2021,
                                "dayOfMonth": 28,
                                "dayOfWeek": 3,
                                "dayOfYear": 118,
                                …,
                                ...,
                                "millis": 1619604000000,
                                "beforeNow": false,
                                "afterNow": true,
                                "equalNow": false
                                },
                        "location": "L2.02",
                        "type": "lab",
                        "expectedAttendance": 4,
                        "actualAttendance": 0
                        },
                        {…},
                        {...}
                ]
        }
```

Route: Get{/users/complete-timetable/[studentId]}

Desc: Gets all of one student's lectures, past and future
Body: n/a
Returns:


```
{
    [
        {
        "uuid": "7f5583ab-546d-46a9-8fc6-da9898c28026",
        "courseCode": "CASE4",
        "moduleCode": "CA4004",
        "date": {
            "year": 2021,
            "dayOfMonth": 28,
            "dayOfWeek": 3,
            "dayOfYear": 118,
            …,
            …,
            "millis": 1619600400000,
            "beforeNow": false,
            "afterNow": true,
            "equalNow": false
        },
        "endTime": {
            "year": 2021,
            "dayOfMonth": 28,
            "dayOfWeek": 3,
            "dayOfYear": 118,
            …,
            ...,
            "millis": 1619604000000,
            "beforeNow": false,
            "afterNow": true,
            "equalNow": false
            },
        "location": "L2.02",
        "type": "lab",
        "expectedAttendance": 4,
        "actualAttendance": 0
        },
        {…},
        {...}
    ]
}
```

Route: Get{/users/week-timetable/[studentId]}

Desc: Gets a student user's lectures that occur in the current week
Body: n/a
Returns:


```
{
    [
        {
        "uuid": "7f5583ab-546d-46a9-8fc6-da9898c28026",
        "courseCode": "CASE4",
        "moduleCode": "CA4004",
        "date": {
            "year": 2021,
            "dayOfMonth": 28,
            "dayOfWeek": 3,
            "dayOfYear": 118,
            …,
            …,
            "millis": 1619600400000,
            "beforeNow": false,
            "afterNow": true,
            "equalNow": false
        },
        "endTime": {
            "year": 2021,
            "dayOfMonth": 28,
            "dayOfWeek": 3,
            "dayOfYear": 118,
            …,
            ...,
            "millis": 1619604000000,
            "beforeNow": false,
            "afterNow": true,
            "equalNow": false
            },
        "location": "L2.02",
        "type": "lab",
        "expectedAttendance": 4,
        "actualAttendance": 0
        },
        {…},
        {...}
    ]
}
```

Lecture Routes:

url = http://0.0.0.0:23567/lecture

Route: Get {/lecture}

Desc: Returns all lectures in DB
Body: n/a
Returns:


```
{
	"lecture": [
	{
		"uuid": "7f5583ab-546d-46a9-8fc6-da9898c28026",
		"courseCode": "CASE4",
		"moduleCode": "CA4004",
		"date": {
			"year": 2021,
			"dayOfMonth": 28,
			"dayOfWeek": 3,
			"dayOfYear": 118,
			…,
			"millis": 1619600400000,
			"beforeNow": false,
			"afterNow": true,
			"equalNow": false
		},
		"endTime": {
			"year": 2021,
			"dayOfMonth": 28,
			"dayOfWeek": 3,
			"dayOfYear": 118,
			...
			"millis": 1619604000000,
			"beforeNow": false,
			"afterNow": true,
			"equalNow": false
		},
		"location": "L2.02",
		"type": "lab",
		"expectedAttendance": 4,
		"actualAttendance": 0
	},
	{
		"uuid": "d1006493-c007-4ba3-a316-ca6ef831e3b8",
		"courseCode": "CASE4",
```

```
"moduleCode": "CA4004",
"date": {
    "year": 2021,
    ...,
```


Route: Post{/lecture}

Desc: Adds a new lecture to the DB, date must not specify a timezone
Body:


```
{
    "courseCode": "CASE4",
    "moduleCode": "CA4004",
    "date": "2021-04-28T09:00:00.000",
    "endTime": "2021-04-28T10:00:00.000",
    "location": "L2.02",
    "type": "lab"
}
```


Returns: 200 or Bad Request with clashing lectures booked in the same room for the same time eg below


```
{
    "lecture": [
    {
        "uuid": "7f5583ab-546d-46a9-8fc6-da9898c28026",
        "courseCode": "CASE4",
        "moduleCode": "CA4004",
        "date": {
            "year": 2021,
            "dayOfMonth": 28,
            "dayOfWeek": 3,
    etc
```


Route:Put{/lecture}

Desc: Updates a lecture in the DB, matches on UUID
Body:


```
{
    “uuid”: “7f5583ab-546d-46a9-8fc6-da9898c28026”
    "courseCode": "CASE4",
    "moduleCode": "CA4004",
    "date": "2021-04-28T09:00:00.000",
    "endTime": "2021-04-28T10:00:00.000",
    "location": "L2.02",
    "type": "lab"
}
```


Returns: 200 or Bad Request with clashing lectures booked in the same room for the same time eg below


```
{
```

```
"lecture": [
{
        "uuid": "7f5583ab-546d-46a9-8fc6-da9898c28026",
        "courseCode": "CASE4",
        "moduleCode": "CA4004",
        "date": {
                "year": 2021,
                "dayOfMonth": 28,
                "dayOfWeek": 3,
Etc
```


Route: Delete{/lecture/[uuid]}

Desc: Deletes a lecture in the DB, matches on UUID.
Body: n/a
Returns: 200 or errors

Route:get{/lecture/[uuid]}

Desc: Gets a lecture in the DB, matches on UUID.
Body: n/a
Returns: Lecture object

Route:Get{/lecture/module/[module_code]}

Desc: gets all lectures for a certain module, matches on module_code.
Body: n/a
Returns: Array of Lecture objects

Route:Get{/lecture/module/[module_code]/[date]}

Desc: gets all lectures for a certain module in a certain week. Week depends on date provided, matches on module_code.
Body: n/a
Returns: Array of Lecture objects

Route:Get{/lecture/course/[course_code]}

Desc: gets all lectures for a certain course, matches on course_code.
Body: n/a
Returns: Array of Lecture objects

Route:Get{/lecture/course/[course_code]/[date]}

Desc: gets all lectures for a certain course in a certain week. Week depends on date provided, matches on course_code.
Body: n/a
Returns: Array of Lecture objects

Route:Get{/lecture/next-module/[module_code]}

Desc:Gets the next soonest lecture for a given module, matches on module_code.
Body: n/a
Returns: Lecture object

Route:Get{/lecture/next-course/[course_code]}

Desc:Gets the next soonest lecture for a given course, matches on course_code.
Body: n/a
Returns: Lecture object

Route:Get{/lecture/location/[location]/}

Desc: Gets all lectures at a location, including past and future, matches on location.
Body: n/a
Returns: Array of Lecture objects

Route:Get{/lecture/location/[location]/[date]}

Desc: Gets all lectures at a location, during the week of date given, matches on location.
Body: n/a
Returns: Array of Lecture objects

Route:Get{/lecture/reg-device/[location]}

Desc: Gets all lectures at a given location, that have a start time that is either within 20 minutes in the future, or 30 minutes in the past based on current time. The point of this is to register a nfc reader device with the current lecture.
Body: n/a
Returns: Array of Lecture objects

Route:Put{/lecture/reg-user/[uuid]/[student-id]}

Desc: Updates a lecture entry to add that a student with [student-id] has attended, used my NFC scanner.
Body: n/a

Returns: 200 or error

Modules Routes:

url = http://0.0.0.0:23567/modules

Route: Get {/modules}

Desc: Returns all modules in DB
Body: n/a
Returns: Array of Module Objects


```
{
	"modules": [
		{
		"uuid": "09c342f0-bcd5-4ca5-9804-f3e92860bba4",
		"moduleCode": "CA4004",
		"moduleTitle": "Soft. Eng.:Process,Principles & Methods",
		"regStudentIds": [
			"12345678",
			"22345678",
			"32345678"
		],
		"expectedAttendance": 3
		}
	]
}
```


Route: Post{/modules}

Desc: Adds a new module to the DB
Body:


```
	{
		"moduleCode": "CA4004",
		"moduleTitle": "Soft. Eng.:Process,Principles & Methods",
		"regStudentIds": [
			"12345678",
			"22345678",
			"32345678"
		]
	}
```


Returns: 200 or error

Route:Put{/modules}

Desc: Adds a new module to the DB
Body:

```
{
    “uuid”: “09c342f0-bcd5-4ca5-9804-f3e92860bba4”
    "moduleCode": "CA4004",
    "moduleTitle": "Soft. Eng.:Process,Principles & Methods",
    "regStudentIds": [
        "12345678",
        "22345678",
        "32345678"
    ]
    “expectedAttendance”: 3
}
```

Returns: 200 or error

Route:Delete{/modules/[uuid]}

Desc: Dels a module in the db, matches on uuid
Body: n/a
Returns: 200 or error

Route:Get{/modules/[uuid]}

Desc: Gets a module in the db, matches on uuid
Body: n/a
Returns: Module Object

Route:Get{/modules/moduleCode/[moduleCode]}

Desc: Gets a module in the db, matches on moduleCode
Body: n/a
Returns: Module Object

Route:Get{/modules/register/[moduleCode]/[studentId]}

Desc: Registers a student for a module, Adds the student Id to the module’s reg_student_ids column in the db
Body: n/a
Returns: 200 or error