Practical work 3

https://github.com/heig-vd-dai-course

Web · PDF

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

Objectives

- Acquire a virtual machine from a cloud provider
- Access the virtual machine (SSH)
- Install Docker and Docker Compose
- Develop a simple CRUD API
- Deploy the applications (reverse proxy + CRUD API)
- Access the applications from a (free) domain name

- The CRUD API manages resources. You can choose what the CRUD API does/manages:
  - Music
  - Books
  - Video games
  - A todo/groceries list
  - ...

The CRUD API must be accessible from the internet.

Demo

The API for the demonstration is accessible at https://heig-vd-dai-course.duckdns.org.

Locally - Compile the project:

```
./mvnw clean package
```

Locally - Build the Docker image with Docker Compose:

```
docker build -t <docker tag> .
```

Locally - Publish the Docker image to the container registry:

```
docker push <docker tag>
```

On the server - Pull the Docker image from the container registry:

```
docker compose pull
```

On the server - Start Traefik (the reverse proxy):

```
docker compose -f traefik/docker-compose.yaml up -d
```

On the server - Start the CRUD API:

```
docker compose -f api/docker-compose.yaml up -d
```

Create a few drinks:

```
# Hot wine
curl -X POST \
  -H "Content-Type: application/json" \
  -d '{"name":"Hot wine","description":"Hot wine with spices","price":3.0}' \
  https://heig-vd-dai-course.duckdns.org/drinks

# Christmas tea
curl -X POST \
  -H "Content-Type: application/json" \
  -d '{"name":"Christmas tea","description":"Warm tea","price":2.0}' \
  https://heig-vd-dai-course.duckdns.org/drinks
```

Get the list of drinks:

```
curl https://heig-vd-dai-course.duckdns.org/drinks
```

Output:

```
[
        {
                "id": 1,
                "name": "Hot wine",
                "description": "Hot wine with spices",
                "price": 3.0
        }
        // All the other drinks
]
```

Filter the drinks with a price equal to 2.0 CHF:

```
curl https://heig-vd-dai-course.duckdns.org/drinks?price=2.0
```

Output:


```
[
        {
                "id": 2,
                "name": "Christmas tea",
                "description": "Warm tea",
                "price": 2.0
        }
]
```

Get a specific drink:

```
curl https://heig-vd-dai-course.duckdns.org/drinks/1
```

Output:

```
{
        "id": 1,
        "name": "Hot wine",
        "description": "Hot wine with spices",
        "price": 3.0
}
```

Update a drink:

```
curl -X PUT \
  -H "Content-Type: application/json" \
  -d '{"name":"Hot wine","description":"Nice hot wine","price":3.0}' \
  https://heig-vd-dai-course.duckdns.org/drinks/1
```

Output:

```
{
        "id": 1,
        "name": "Hot wine",
        "description": "Nice hot wine",
        "price": 3.0
}
```

Delete a drink:

```
curl -X DELETE -i https://heig-vd-dai-course.duckdns.org/drinks/1
```

Output:

```
HTTP/2 204
content-type: text/plain
date: Sat, 16 Dec 2023 13:31:56 GMT
```

No content as we return a `204` (No Content) status code!

Adding another drink with the same name:

```
curl -X POST \
  -H "Content-Type: application/json" \
  -d '{"name":"Christmas tea","description":"Another tea","price":2.0}' \
  https://heig-vd-dai-course.duckdns.org/drinks
```

Output:

```
Conflict
```

Leads to a `409` (Conflict) status code as we want to keep the names unique.

Get the schedules of the Baleinev hot wine week:

```
curl https://heig-vd-dai-course.duckdns.org/schedules
```

Output:

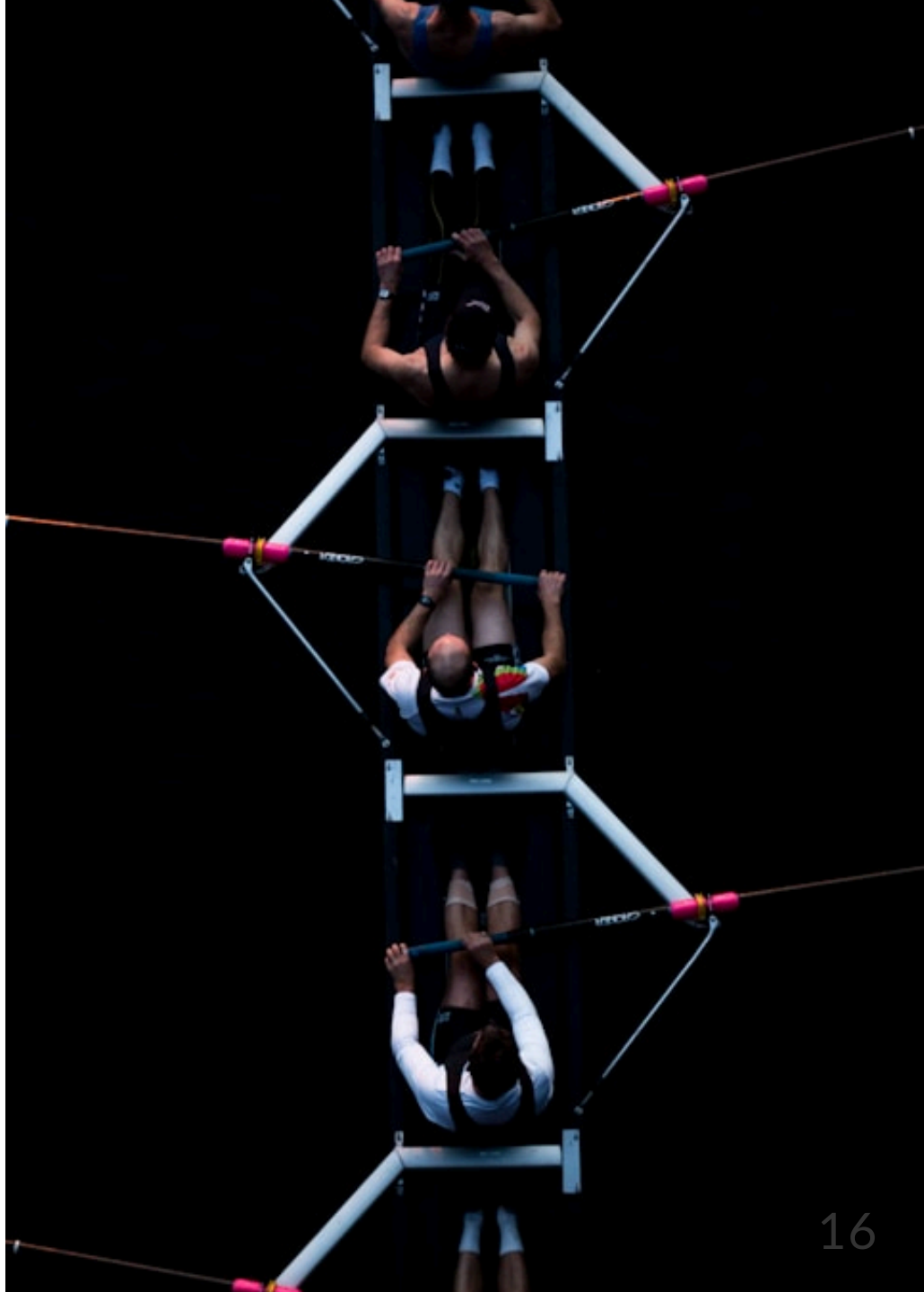
```
[
        { "id": 0, "day": "MONDAY", "start": [10, 0], "end": [18, 0] },
        { "id": 1, "day": "TUESDAY", "start": [10, 0], "end": [18, 0] },
        { "id": 2, "day": "WEDNESDAY", "start": [10, 0], "end": [18, 0] },
        { "id": 3, "day": "THURSDAY", "start": [10, 0], "end": [22, 0] },
        { "id": 4, "day": "FRIDAY", "start": [10, 0], "end": [18, 0] }
]
```

Group composition

More details for this section in the course material.

Group composition

- 2 or 3 students per group
- Create a GitHub Discussion to:
  - Announce your group members
  - Announce your idea (even a draft is fine)
- **Do it as soon as possible before next week!**
  - This helps us to plan the presentations



Idea validation

More details for this section in the course material.

Idea validation

- You must state your idea on your GitHub Discussion
- We might ask you to change your idea if it is too simple or too complex
- We will help you to find a good idea if needed
- **Do it as soon as possible before next week!**

Grading criteria

More details for this section in the course material.

Grading criteria

You can find all the grading criteria in the course material:

- 0 point - The work is insufficient
- 0.1 point - The work is done
- 0.2 point - The work is well done (without the need of being perfect)

Maximum grade: 25 points * 0.2 + 1 = 6

Constraints

More details for this section in the course material.

Constraints

- The application must be written in Java, compatible with Java 21
- The application must be built using Maven with the `maven-shade-plugin` plugin
- The application must use the Javalin dependency
- You can only use the Java classes seen in the course
- Your application must be slightly more complex and slightly different than the examples presented during the course (we emphasize the word **slightly**, no need to shoot for the moon!)
- The web application can only use the HTTP/HTTPS protocols

Tips

More details for this section in the course material.

Tips

You are allowed to reuse your Bases de données relationnelles (BDR) project for this practical work.

You can decide to merge the two projects into one if you want or keep the idea and implement it in a different way (in memory for example, as you will see in the course materials).

Submission

More details for this section in the course material.

Submission

Your work is due as follow:

- DAI-TIC-B (Monday mornings): **19.01.2025 23:59**
- DAI-TIC-C (Friday mornings): **23.01.2025 23:59**

Update the GitHub Discussion with the link to your repository as mentioned in the course material.

If you do not submit your work on time and/or correctly, you will be penalized (-1 point on the final grade for each day of delay).

Presentations

More details for this section in the course material.

Presentations

The practical work presentations will take place in **room B51a** (the same room as the first presentation) on:

- DAI-TIC-B (Monday mornings): **20.01.2025**
- DAI-TIC-C (Friday mornings): **24.01.2025**

We only have **7 minutes per group**. You decide what you want to show us and how you want to present it.

Come 5 minutes before your time slot with your computer. You will have access to a video projector.

Grades and feedback

Grades will be entered into GAPS, followed by an email with the feedback.

The evaluation will use exactly the same grading grid as shown in the course material.

Each criterion will be accompanied by a comment explaining the points obtained, a general comment on your work and the final grade.

If you have any questions about the evaluation, you can contact us!

Questions

Do you have any questions?

Find the practical work

You can find the practical work for this part on GitHub.

Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this practical work?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

➡️ GitHub Discussions

You can use reactions to express your opinion on a comment!

Sources

- Main illustration by Lāsma Artmane on Unsplash
- Illustration by Aline de Nadai on Unsplash
- Illustration by Josh Calabrese on Unsplash
- Illustration by Nicole Baster on Unsplash
- Illustration by Chris LaBarge on Unsplash
- Illustration by Jan Antonin Kolar on Unsplash