Practical work 4

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

Web · PDF

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

Practical work 4

- Get a virtual machine on our cloud
- 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

- A CRUD API to manage resources
- You can choose what the CRUD API does/manages:
  - Music
  - Books
  - Video games
  - A todo list
  - Groceries
  - ...
- **Groups of 4 students**

Demo

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

Locally - Compile the project:

```
./mvnw clean package
```

Locally - Build the Docker image with Docker Compose:

```
docker compose build
```

Locally - Publish the Docker image to the container registry:

```
docker compose push
```

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.yml up -d
```

On the server - Start the CRUD API:

```
docker compose -f api/docker-compose.yml 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.dedyn.io/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.dedyn.io/drinks
```

Get the list of drinks:

```
curl https://heig-vd-dai-course.dedyn.io/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.dedyn.io/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.dedyn.io/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.dedyn.io/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.dedyn.io/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.dedyn.io/drinks
```

Output:

```
Conflict
```

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

Guidelines

This is important, please read carefully!

- State your group on GitHub Discussions before the end of the day!
- Define someone in your group as the responsible
- **Write to the IT department** to get a virtual machine as soon as possible (**before the end of the day**!)
  - Use the template email provided in the course material.
  - Put the teaching staff in CC.

Practical work review

The practical work review will take place on **Tuesday 23.01.2024** in the **room B51a**, next to our classroom.

We only have **10 minutes per group** (10 minutes of presentation, no time for questions). Please be prepared to present your work. 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.

The order of presentation is random and is stated in the next slides.

#	Group	Passage
1	Alexandre Philibert, Gwendal Piemontesi, Valentin Ricard and Trueb Guillaume	13h25
2	Thomas Vuilleumier, Sebastian Diaz, Arthur Menétrey and Lionel Pollien	13h35
3	Komarov Sergey and Jano Ahmad	13h45
4	Sarah Jallon, Jonas Troeltsch, Jeremiah Steiner and Simon Guggisberg	13h55
5	Pirakas Anthon, Aurélien Richard, Romain Humair and Lucas Lattion	14h05

#	Group	Passage
6	Bleuer Rémy, Graf Calvin, Lopez Esteban and Sottile Alan	14h15
7	Massimo Stefani, Loïc Herman, Kevin Farine and Olin Bourquin	14h25
8	Jaques Colin, Iorio Alexandre, Mulugeta Theodros and Slimani Walid	14h35

Find the practical work

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

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!

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