Example

In this lesson, we'll introduce a Netflix stack coding example.

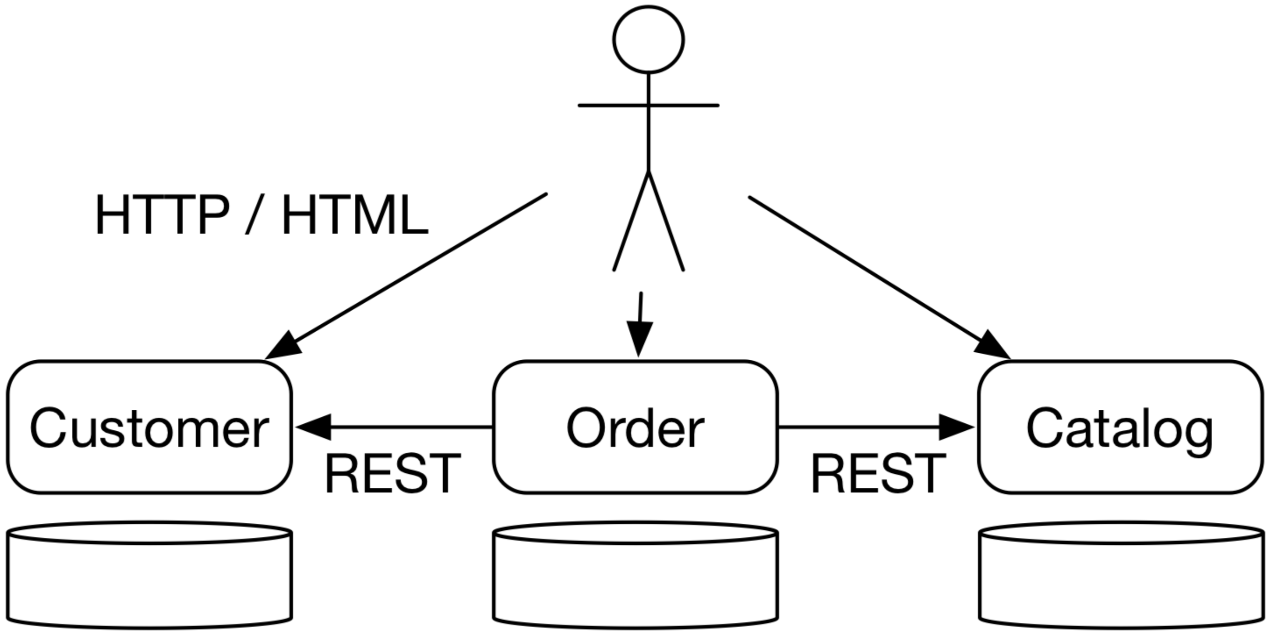
We'll cover the following ^

Introduction#

The example for this chapter can be found at https://github.com/ewolff/microservice (https://github.com/ewolff/microservice). It consists of **three** microservices:

- The **catalog** microservice that manages the information about the items.
- The **customer** microservice that stores the data of the customers.
- The **order** microservice that can accept new orders by using the catalog and the customer microservice.

Architecture of the example#



Architecture of the Netflix Example

- Each of the microservices has its own web interface with which users can interact.
- Among each other, the microservices communicate via REST.
- The order microservice requires information about customers and items from the other two microservices.

In addition to the microservices, there is a **Java application** that displays the **Hystrix dashboard** where monitoring the Hystrix circuit breakers is visualized.

The drawing in the section Docker containers and ports shows the entire example at the level of the Docker containers.

Running the example#

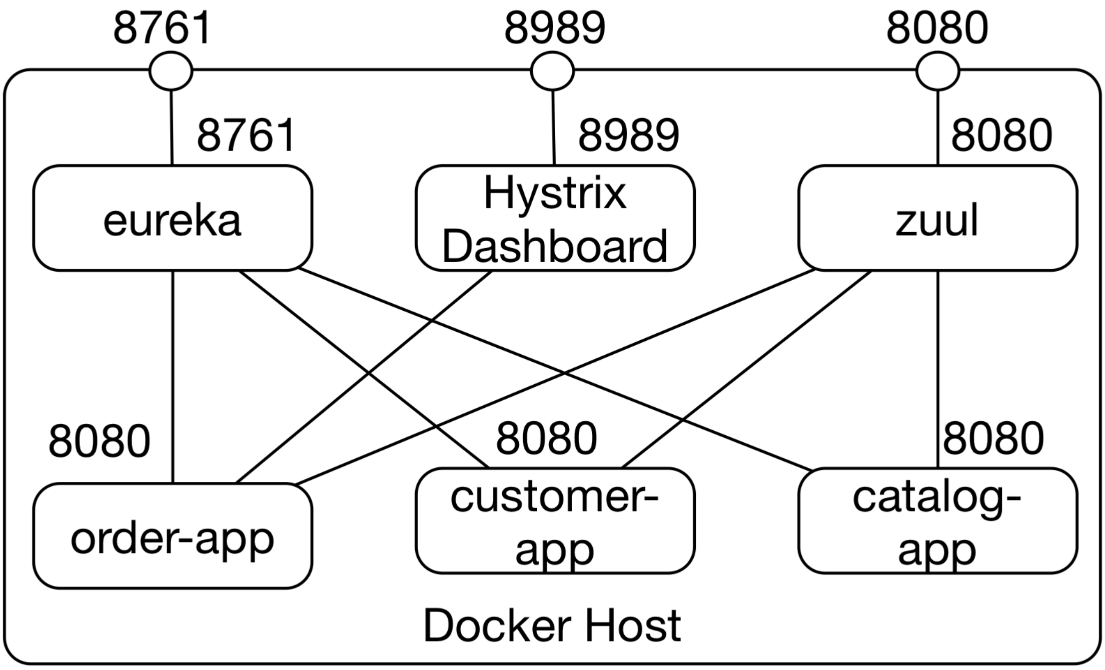
Running the example#

First, the code has to be downloaded with `git clone https://github.com/ewolff/microservice.git`. Then the code has to be compiled with `./mvnw clean package` (macOS, Linux) or `mvnw.cmd clean package` (Windows) in the directory `microservice-demo`. See this lesson (https://www.educative.io/collection/page/10370001/5441945024331776/4964597493661696) in the appendix for more details on Maven and how to troubleshoot the build. Afterwards, the Docker containers can be built with `docker-compose build` in the directory `docker` and started with `docker-compose up -d`. See this lesson (https://www.educative.io/collection/page/10370001/5441945024331776/4600297679749120) and the one after in the appendix for more details on Docker, docker-compose and how to troubleshoot them. Subsequently, the Docker containers are available on the Docker host.

https://github.com/ewolff/microservice/blob/master/HOW-TO-RUN.md (https://github.com/ewolff/microservice/blob/master/HOW-TO-RUN.md) in detail explains the steps that need to be performed to build and run the example.

```
version: '3'
services:
  eureka:
    image: educative1/mapi_ms_eureka
    ports:
      - "8761:8761"
  customer:
    image: educative1/mapi_ms_customer
    links:
     - eureka
  catalog:
    image: educative1/mapi_ms_catalog
    links:
     - eureka
  order:
    image: educative1/mapi_ms_order
    links:
     - eureka
  zuul:
    image: educative1/mapi_ms_zuul
    links:
     - eureka
    ports:
      - "8080:8080"
  turbine:
    image: educative1/mapi_ms_turbine
    links:
     - eureka
    ports:
      - "8989:8989"
```

Docker containers and ports#



Docker Containers in the Netflix Example

The Docker containers communicate via an **internal network**. Some Docker containers can also be used via a port on the Docker host. The Docker host is the computer on which the Docker containers run.

The three microservices **order**, **customer**, and **catalog** each run in their own Docker containers. Access to the Docker containers is only possible *within* the Docker network.

Routing via Zuul#

In order to be able to use the services from the outside, **Zuul** provides routing.

- The Zuul container can be accessed from outside under port `8080` and forwards requests to the microservices.
- If the Docker containers are running **locally**, the URL is http://localhost:8080 (http://localhost:8080).
- At this URL, there is also a web page available which includes links to all microservices, Eureka, and the Hystrix dashboard

all microservices, Eureka, and the Hystrix dashboard.

Service discovery via Eureka#

Eureka serves as a **service discovery** solution.

- The dashboard is available at port `8761`.
- This port is also accessible at the Docker host.
- For a **local Docker installation**, the URL is http://localhost:8761 (http://localhost:8761).

Hystrix dashboard#

Finally, the **Hystrix dashboard** runs in its own Docker container that can also be accessed under port `8989` on the Docker host, for example at http://localhost:8989 (http://localhost:8989).

1 The three microservices run ___.

○ A) directly on the host

○ B) within one single Docker container

○ C) within individual Docker containers

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

In the next lesson, we’ll discuss service discovery with Eureka in more detail.

← Back

Introduction

Next →

Eureka: Service Discovery

☑ Mark as Completed

Report an Issue