Introduction

In this lesson, we'll get a quick introduction to this course.

WE'LL COVER THE FOLLOWING

Why this course?

Microservices are one of the most important software architecture trends. A number of detailed guides to microservices are already currently available, including the microservices-book written by the author of this course. **So, why do we need yet another course on microservices?**

It is **one thing to define an architecture, quite another to implement it**. This course presents technologies for the implementation of microservices while highlighting their advantages and disadvantages.

The **focus** rests specifically on **technologies** for entire microservices systems. Each individual microservice can be implemented using different technologies.

Therefore, the technological decisions about frameworks for individual microservices are not as important as the decisions at the level of the overall system. For individual microservices, the decision about a framework can be quite easily revised. However, the technologies chosen for the overall system are difficult to change.

Compared to the microservices-book, this course talks primarily about

technologies. We have another course called **An Introduction to Microservice Principles and Concepts** that does discuss architecture and reasons for and against microservices.

Basic principles

To become familiar with microservices, an introduction into microservices-based architectures and their benefits, disadvantages, and variations is essential.

An Introduction to Microservice Principles and Concepts explains the basic principles to the extent required for understanding the **practical implementations**.

Concepts

Microservices require **solutions for different challenges**. Among those are concepts for **integration** (*frontend integration, synchronous, and asynchronous microservices*) and for **operation** (*monitoring, log analysis, tracing*). Microservices platforms such as *PaaS* or *Kubernetes* represent exhaustive solutions for the operation of microservices.

Recipes

The course uses **recipes as a metaphor for the technologies**, which can be used to implement the different concepts you will learn. Each approach shares a number of features with a recipe.

- Each recipe is described in *practical terms*, including an example technical implementation. The most important aspect of the examples is their *simplicity*. Each example can be easily followed, extended, and modified.
- The course provides the reader with a *plethora of recipes*. The readers have to select a specific recipe from this collection for their projects, akin to a cook who has to select a recipe for their menu. The course shows different options because, in practice, nearly every project has to be dealt with differently. The recipes build the basis for this.
- *Variations* exist for each recipe. After all, you can cook a recipe in a

variety of ways. This is also true for the technologies described in this

course. Sometimes the variations are very simple so that they can be immediately implemented as experiments in an executable example.

Each recipe includes an associated *executable example* based on a concrete technology. The examples can be run individually as they are not based on each other. This allows the readers to concentrate on the recipes that are interesting and useful for them and to skip the examples that are less relevant for their work. In this manner, the course provides easy access for obtaining an *overview* of the relevant technologies, thereby enables the readers to select a suitable technology stack. Subsequently, the readers can use the links supplied in the course to acquire in-depth knowledge about the relevant technologies.

Source code

Sample code is provided for almost all the technologies presented in this course. If the reader wants to really understand the technologies, they should browse the code. It also makes sense to look at the code to understand how the concepts are actually implemented.

In the next lesson, we'll have a quick look at the structure of the course.