

# Pedalo Reservation Applikation

Modul 347

Elia Kuster & Fabio Dauru

## Auftrag:

Der Auftrag des erstellten Projektes ist eine Webapplikation zu erstellen, welche auf einem Docker laufen gelassen werden kann. Die Applikation soll einen persistenten Service haben, auf welchem die CRUD-Methoden aufgerufen werden kann. Dieses Projekt soll in Zwei- bis Dreiergruppen umgesetzt werden. Für die Technologien war Java Spring vorgeschrieben und für alle anderen Technologien waren wir frei.

## Anwendung:

Unsere Applikation ist dafür gedacht, dass ein Mitarbeiter einer Seeanlage Reservationen von Pedalos eintragen kann. Der Mitarbeiter kann so überprüfen welche Pedalos an welchem Datum und um welche Zeit reserviert werden. Da die Applikation nur für Mitarbeiter gebraucht wird, ist kein Login vorhanden.

## Anforderungen:

Funktional	Nicht Funktional
Der User kann Reservationen hinzufügen. Die Reservationen werden aufgelistet. Die Reservationen werden nach Datum geordnet. Die Reservationen können gelöscht werden.	Man benötigt die NoSQL-Datenbank MongoDB. Man benötigt den Javainterpreter IntelliJ. Man benötigt Docker Desktop.

## Ressourcen:

Wir haben uns entschieden, das Projekt mit allen benötigten Dependencies mit dem «[Spring initializr](#)» zu generieren, da wir kein grosses Projekt brauchten und das so sehr einfach ging.

Die Dependencies die wir gebraucht haben sind:

- Spring Web

Wir können mit «Spring Web» unsere Webanwendung erweitern und mit RESTful eine api bauen und dann mit dieser die Daten aus der Datenbank an das Frontend senden. Das Ganze kann ganz einfach gemacht werden, dank Anmerkungsbasiertem Programmieren so können wir ganz einfach Controller zuweisen oder Daten über Formulare in die Datenbank eintragen mit dem @PostMapping.

- Thymleaf

Thymleaf benötigten wir, um den ganzen HTML-Code richtig darzustellen und laden. Es hilft uns eine moderne und konforme Webanwendung zu erstellen. Wir haben es benötigt, damit wir ganz einfach die Dateneingaben von Formularen verarbeiten konnten. Auch um Reservationen dynamisch auf unserer Webseite zu laden, war Thymleaf sehr praktisch, da der User immer wieder neue Reservationen einträgt und wir diese natürlich nicht hardcoden können.

- Spring Data MongoDB

Wir uns entschieden haben mit einer MongoDB zu arbeiten, da wir uns damit schon auskennen und es sehr einfach zu verbinden ist. Wir benötigen die «Spring Data MongoDB» Dependency, hauptsächlich um Daten im richtigen Format (BSON) zu speichern und die Verbindung mit der Datenbank herzustellen, mit zum Beispiel MongoClient oder MongoDBDatabase. Das konnten wir alles erreichen dank den erweiterten Tags, welche die Dependency hinzufügt.

- Docker File

Was wir auch noch brauchten, war ein Docker File. In diesem File haben wir die Konfiguration festgelegt, um die app.jar Datei aus unserer Applikation kopiert und dann ausführt. Dies befindet sich auf Zeile acht, dort kopieren wir das File mit dem «COPY» -Tag.

- Docker-Compose.yml

Zuletzt brauchten wir noch ein «docker-compos.yml» File um die zwei Docker Container, für die Datenbank und unseren Webserver, miteinander zu verbinden und um den Datenaustausch zu garantieren. Wichtig ist dabei, dass die Ports richtig gesetzt sind und dass die Mongo-Datenbank-URI richtig hinterlegt ist. Mit der Dockerisierung läuft die ganze Applikation bzw. die Verbindung extern und muss nicht manuell gestartet werden.