

Mitarbeiterverwaltung Backend DEUTSCH export

① [Einleitung](#)

Ziel der Fallstudie ist es, eine Backend-Anwendung zur Mitarbeiterverwaltung zu erstellen. Die Anwendung soll dabei die Möglichkeit bieten, Mitarbeiter

- anzulegen,
- nach verschiedenen Kriterien abzufragen,
- zu aktualisieren sowie
- zu löschen.

Ein Mitarbeiter hat mindestens die folgenden Eigenschaften:

- Vorname
- Nachname
- Alter
- Personalnummer
- Abteilung
- Berufsbezeichnung
- Jahreseinkommen

Ein Beispielmitarbeiter wäre:

Vorname	Nachname	Alter	Personalnummer	Abteilung	Berufsbezeichnung	Jahreseinkommen
Max	Mustermann	38	mm1288_se	Softwareentwicklung	Backend-Entwickler	75000

[Teilaufgabe 1: Projekt-Setup](#)

Um mit dem eigentlichen Projekt zu beginnen, soll eine Backend-Anwendung aufgesetzt werden. Hierzu zählt die Auswahl von Frameworks bzw. Libraries sowie passender Tools zur Entwicklung. Zudem soll für die Speicherung der Daten eine geeignete Datenbank gewählt werden.

Ziel: Die Anwendung fährt fehlerfrei hoch und die eigentliche Projektimplementierung kann begonnen werden.

[Teilaufgabe 2: Mitarbeiter anlegen, abfragen, aktualisieren und löschen](#)

Die Anwendung soll volle CRUD-Funktionalität anbieten, d.h. es muss eine REST-API bereitgestellt werden, mit der es möglich ist,

- einen neuen Mitarbeiter anzulegen,
- alle Mitarbeiter abzufragen,
- einzelne Mitarbeiter abzufragen,
- Mitarbeiter zu aktualisieren sowie
- Mitarbeiter zu löschen.

Ziel: Neue Mitarbeiter können angelegt, abgefragt, aktualisiert und gelöscht werden.

Tipp: Prüfe mit einem REST-Client (z.B. Postman), ob die API tatsächlich so funktioniert, wie du es vorgesehen hast.

[Teilaufgabe 3: Speziellere Suchanfragen](#)

Die Anwendung soll neben der Standard-CRUD-Funktionalität auch speziellere Suchanfragen verarbeiten können. Dazu soll die API um Methoden erweitert werden, mit deren Hilfe Mitarbeiter

- zwischen 50 und 60 Jahren gesucht,
- absteigend nach ihrem Gehalt abgefragt sowie
- aufsteigend nach ihrem Nachnamen abgefragt

werden können.

Zur Validierung kannst du gerne die Beispieldaten aus employees.json verwenden:

- Es gibt 2 Mitarbeiter zwischen 50 und 60 Jahren.
- Fragt man alle Mitarbeiter absteigend nach ihrem Gehalt ab, so muss "Miriam Müller" an erster Stelle stehen.
- Fragt man alle Mitarbeiter aufsteigend nach ihrem Nachnamen ab, so muss "Agathe Adams" an erster Stelle stehen.

Ziel: Es können Mitarbeiter zwischen 50 und 60 Jahren, absteigend nach ihrem Gehalt und aufsteigend nach ihrem Nachnamen ausgegeben werden.

Tipp: Prüfe mit einem REST-Client (z.B. Postman), ob die API tatsächlich so funktioniert, wie du es vorgesehen hast.

Bonusaufgabe: Testen der Anwendung

Die Anwendung soll in Teilen getestet werden. Nutze dafür die Test-Tools deiner Wahl.

Tipp: Zum Testen des Backends kannst du die Spring-eigene Testing-Library verwenden.

Ziel: Die Anwendung ist teilweise getestet.

Tipp: Nutze verbleibende Zeit für Refactoring.