

# REST-API mit Spring Boot

## 1 Quelle

Accessing JPA Data with REST: <https://spring.io/guides/gs/accessing-data-rest/>

## 2 Spring-Boot-Projekt mit Spring Initializr erstellen

<https://start.spring.io/>

### Dependencies

- Rest Repositories
- Spring Data JPA
- H2 Database

## 3 Konfigurationsdatei /src/main/resources/application.properties

### Port ändern

```
server.port=8081
```

### H2-Console aktivieren

```
spring.h2.console.enabled=true  
spring.h2.console.path=/h2-console
```

## 4 Entity-Klasse erstellen

```
package de.kohnlehome.springbootrestapi;  
  
import ...  
  
@Entity  
public class Person {  
    @Id  
    @GeneratedValue(strategy = GenerationType.AUTO)  
    private long id;  
  
    private String vorname;  
    private String nachname;  
    private LocalDate geburtsdatum;  
  
    // ... getter + setter fuer vorname, nachname, geburtsdatum  
}
```

## 5 Repository-Interface erstellen

```
package de.kohnlehome.springbootrestapi;  
  
import org.springframework.data.repository.PagingAndSortingRepository;  
  
public interface PersonRepository extends PagingAndSortingRepository <Person, Long> { }
```

## 6 CRUD mit REST-Client

### CREATE

POST <http://localhost:8081/persons/>

Content-Type: application/json

```
{
  "vorname": "Anna",
  "nachname": "Arm",
  "geburtsdatum": "2001-01-01"
}
```

### READ

GET <http://localhost:8081/persons/1>

### UPDATE

#### alle Elemente überschreiben

PUT <http://localhost:8081/persons/1>

Content-Type: application/json

```
{
  "vorname": "Annabell",
  "nachname": "Armdran",
  "geburtsdatum": "2001-01-01"
}
```

#### einzelne Elemente überschreiben

PATCH <http://localhost:8081/persons/1>

Content-Type: application/json

```
{
  "vorname": "Anna"
}
```

### DELETE

DELETE <http://localhost:8081/persons/1>

## 7 H2-Console

<http://localhost:8081/h2-console/>

JDBC URL: jdbc:h2:mem:testdb

User Name: sa

Password: <leer lassen>