Tasks Application’s  
docs (20191219)

### About

This app makes task management easer. You have every task in one place, wherever you are and whenever you want.

### Version

This is documentation’s version number 20191219 and it corresponds with the app version 0.0.1

### Content

[Architecture](#_Architecture)

[Installation](#_Installation)

[Dependencies](#_Dependencies)

[API methods](#_API_methods)

[Relations](#_Relations)

[Classes](#_Classes)

### Architecture

MySQL Database

Web REST API

Tasks App

Client / User

### Installation

Suggested installation is based on local server with MySQL database (it can be downloaded here: <https://www.mysql.com/downloads/>).

App configuration file *application.properties*:

spring.jpa.hibernate.ddl-auto=update  
spring.jpa.database=*mysql*spring.jpa.properties.hibetnate.dialect=org.hibernate.dialect.MySQL5Dialect  
spring.datasource.url=jdbc:mysql://localhost:3306/task\_crud?serverTimezone=Europe/Warsaw&useSSL=False  
spring.datasource.username=kodilla  
spring.datasource.password=your\_password  
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

### Dependencies

dependencies **{** compile('org.springframework.boot:spring-boot-starter-web')  
 compileOnly('org.projectlombok:lombok')  
 compile('org.springframework.boot:spring-boot-starter-data-jpa')  
 compile('mysql:mysql-connector-java')  
 annotationProcessor 'org.projectlombok:lombok'  
  
 testCompile 'org.springframework.boot:spring-boot-starter-test'  
**}**

### API methods

GET /v1/task/getTasks – shows all tasks

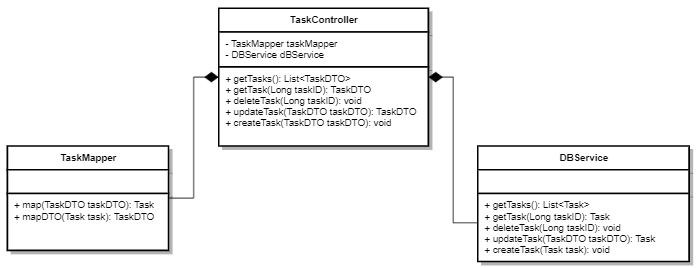
GET /v1/task/getTask?taskID=*taskID* – shows specific task

POST /v1/task/createTask – needs additional data in the request body in order to create new task

PUT /v1/task/updateTask – needs additional data in the request body in order to update new task

DELETE /v1/task/deleteTask?taskID= *taskID* – deletes specific task

### Relations



### Classes

Task class defines how the SQL database entity Tasks has to be created.

@Getter  
@AllArgsConstructor  
@NoArgsConstructor  
@Entity  
@Table(name = "Tasks")  
public class Task  
{  
 @Id  
 @GeneratedValue(strategy = GenerationType.*AUTO*)  
 @NotNull  
 @Column(name = "taskID")  
 private Long taskID;  
  
 @Column(name = "taskTitle")  
 private String taskTitle;  
  
 @Column(name = "taskContent")  
 private String taskContent;  
}

TaskDTO class defines java representation of the SQL database objects/entries.

@Getter  
@AllArgsConstructor  
@NoArgsConstructor  
public class TaskDTO  
{  
 private Long taskID;  
 private String taskTitle;  
 private String taskContent;  
}