



---

# todo

## Dokumentation der Anwendung

### Full-Stack-Webanwendungen Sommersemester 2023

---

Lucas Schiessl  
lucas.schiessl@hs-augsburg.de  
Informatik

Hannes Ziereis  
hannes.ziereis@hs-augsburg.de  
Informatik

Christoph Herb  
christoph.herb@hs-augsburg.de  
Informatik

2795 Zeichen (mit Leerzeichen)

# Inhaltsverzeichnis

<b>1</b>	<b>Allgemein</b>	<b>2</b>
1.1	Beschreibung . . . . .	2
1.2	Organisation . . . . .	2
1.3	Aufbau . . . . .	2
<b>2</b>	<b>Frontend</b>	<b>3</b>
2.1	Design . . . . .	3
2.2	Funktionen . . . . .	3
<b>3</b>	<b>Backend</b>	<b>3</b>
3.1	Funktionen . . . . .	3
3.2	Tests . . . . .	3
<b>4</b>	<b>Datenbank</b>	<b>3</b>
<b>5</b>	<b>Anhang</b>	<b>4</b>
5.1	GitLab Issues . . . . .	4
5.2	API Dokumentation . . . . .	6

# 1 Allgemein

## 1.1 Beschreibung

\* paar infos über die Anwendung \* cooles features

## 1.2 Organisation

\* auf gitlab issues zeigen (anhang) \* feature based entwicklung erwähnen

## 1.3 Aufbau

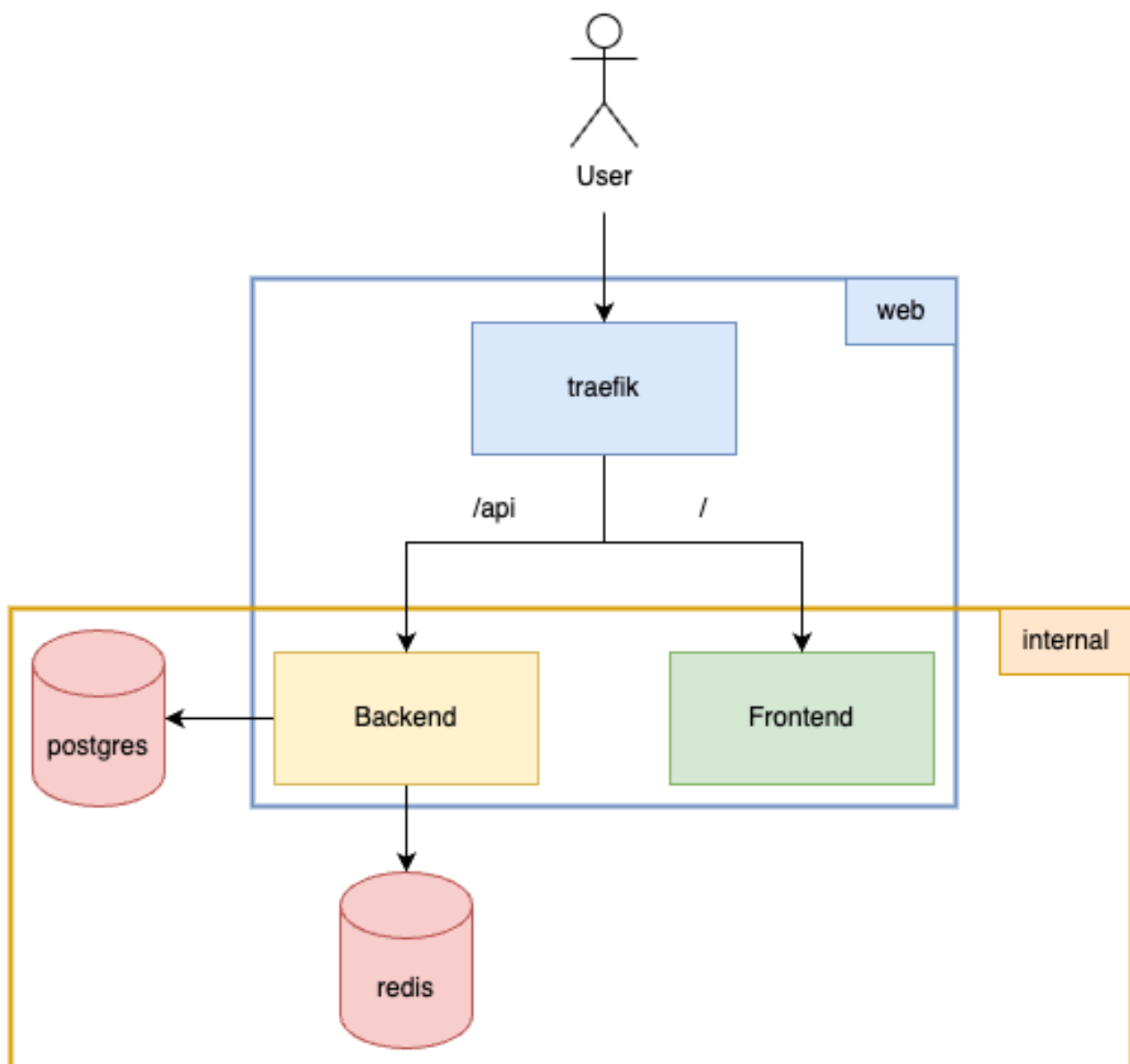


Abbildung 1: Architektur der deployten Anwendung

## **2 Frontend**

### **2.1 Design**

\* figma design

### **2.2 Funktionen**

\* kernfeatures

## **3 Backend**

### **3.1 Funktionen**

\* features

### **3.2 Tests**

\* cucumber tests erwähnen

## **4 Datenbank**

\* ER diagramm

## 5 Anhang

### 5.1 GitLab Issues

Title	Assignee
FE Install Tailwindcss	Lucas Schießl
BE Create blueprint for backend	Christoph Herb
BE Add postgresql	Lucas Schießl
BE Add refresh endpoint	Christoph Herb
BE Add getting started documentation	Christoph Herb
DOC Create documentation	NaN
FE Create Figma Design	Christoph Herb
BE Create Database model	Hannes Ziereis
BE Implement swagger docs or create REST-API diagram manually	Lucas Schießl
FE Implement basic design	Lucas Schießl
FE First interaction with BE	Christoph Herb
FE Implement authentication with BE	Lucas Schießl
FE Cleanup frontend	Christoph Herb
FE Implement Nav Bar and design Login Form	Lucas Schießl
BE Endpoints for tasks	Hannes Ziereis
FE Interaction with tasks	Hannes Ziereis
BE Implement better logger	Christoph Herb
BE Hash password before saving into DB	Christoph Herb
FE Implement login in frontend	Lucas Schießl
FE style login form	Lucas Schießl
FE create Registration Form	Lucas Schießl
BE Add logout endpoint	Christoph Herb
FE Add error handling to login and registration form	Lucas Schießl
BE Fix tasks endpoint, swagger docs and cucumber tests	Christoph Herb
BE, FE Add docker support	Christoph Herb
FE Call Logout and refresh endpoints from frontend	Lucas Schießl
BE Fix bugs in backend	Christoph Herb
DOC Add postman collection	Christoph Herb
FE Add more functionality to task bar	Lucas Schießl
FE Add functionality to settings page	Lucas Schießl
FE Refactoring	Lucas Schießl
FE Refresh does not work as expected	Lucas Schießl
BE, FE Fix docker prod environment	Christoph Herb
BE Error messages for frontend	Christoph Herb
BE Sorting of tasks with date	Christoph Herb
BE Add filtering for categories	Christoph Herb
FE Add support for multiple languages	Lucas Schießl
FE Show taskdetails in task list	Hannes Ziereis
FE Refactor AppTask into its components	Hannes Ziereis

Continued on next page

Title	Assignee
BE Move password hashing to postgres	Christoph Herb
FE Cleanup	Christoph Herb
BE Generate secrets instead of hard coding	Christoph Herb
BE Add optional Tags to Tasks	Lucas Schießl
FE Fix token refresh endless loop	Christoph Herb
FE Add calendar for scheduled tasks	Christoph Herb
FE Show Tag in overview and details modal	NaN
FE BUG - Modals opens two times	Christoph Herb
FE Detailed View - Done status doesn't update	Christoph Herb
FE Fix styling of task list	Christoph Herb
FE Edit Task in Detail modal	Hannes Ziereis
FE Improve modal	Christoph Herb

## 5.2 API Dokumentation

# Tasks API

## Overview

Backend API for the Tasks service.

MIT

## Tags

### Auth

Bearer authentication

### Me

Me endpoint

### Tasks

Task endpoint

### User

User endpoint

## Security

Type	Name	Scopes
apiKey	BearerAuth	

## Paths

### **POST** /api/v1/auth/login Login

#### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links

Code	Description	Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

## **GET /api/v1/auth/verify** Verify logged in user

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET /api/v1/auth/refresh** Refresh token after expiration of your access token

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links



Code	Description	Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

## **GET /api/v1/auth/logout** Logout user

### Responses

Code	Description	Links
204	success response	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET /api/v1/me** Get infos about current logged in user

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET /api/v1/me/tasks** Get tasks of current logged in user

### Parameters

Type	Name	Description	Schema
query	<b>orderBy</b> <i>optional</i>	orderBy	enum ( asc,desc)
query	<b>sortBy</b> <i>optional</i>	sortBy	enum ( dueDate,isChecked,createdAt,updatedAt)
query	<b>tag</b> <i>optional</i>	tag	string
query	<b>isChecked</b> <i>optional</i>	isChecked	enum ( false,true)

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **POST /api/v1/me/tasks** Create a new Task with current user

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
400	Validation error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET /api/v1/me/tags** Get Tags of tasks of current logged in user

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET /api/v1/tasks** Get all Tasks

### Parameters

Type	Name	Description	Schema
query	<b>orderBy</b> <i>optional</i>	orderBy	enum ( asc,desc)
query	<b>sortBy</b> <i>optional</i>	sortBy	enum ( dueDate,isChecked,createdAt,updatedAt)
query	<b>tag</b> <i>optional</i>	tag	string
query	<b>isChecked</b> <i>optional</i>	isChecked	enum ( false,true)

#### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **POST /api/v1/tasks** Create a new Task

#### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links

Code	Description	Links
400	Validation error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET** /api/v1/tasks/{taskId} Get specific task

### Parameters

Type	Name	Description	Schema
path	<b>taskId</b> <i>optional</i>	Task ID	string

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	NotFound error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **PUT** /api/v1/tasks/{taskId} Update a Task

### Parameters

Type	Name	Description	Schema
path	taskId <i>optional</i>	Task ID	string

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
400	Bad Request  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	NotFound error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## ***DELETE* /api/v1/tasks/{taskId} Delete a Task**

### Parameters

Type	Name	Description	Schema
path	taskId <i>optional</i>	Task ID	string

### Responses

Code	Description	Links
204	success response	No Links
400	Bad Request  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	NotFound error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## ***PATCH* /api/v1/tasks/{taskId}/toggle Toggle the isChecked field**

### Parameters

Type	Name	Description	Schema
path	taskId <i>optional</i>	Task ID	string

#### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
400	Bad Request  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	NotFound error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## **GET** /api/v1/users Get all users

#### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links



Code	Description	Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
<b>apiKey</b>	BearerAuth	

## **POST** /api/v1/users Create a new user

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
400	Validation error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

## **GET** /api/v1/users/{userId} Get user by id

### Parameters

Type	Name	Description	Schema
<b>path</b>	<b>userId</b> <i>optional</i>	userId	string

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	Not Found error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
<b>apiKey</b>	BearerAuth	

## **PUT** /api/v1/users/{userId} Update user by id

### Parameters

Type	Name	Description	Schema
<b>path</b>	<b>userId</b> <i>optional</i>	userId	string

### Responses

Code	Description	Links
200	success response  <i>Content</i> <b>application/json</b>	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links

Code	Description	Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	Not Found error  <i>Content</i> <b>application/json</b>	No Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
<b>apiKey</b>	BearerAuth	

## ***DELETE* /api/v1/users/{userId} Delete user by id**

### Parameters

Type	Name	Description	Schema
<b>path</b>	<b>userId</b> <i>optional</i>	userId	string

### Responses

Code	Description	Links
204	success response	No Links
401	Unauthorized error  <i>Content</i> <b>application/json</b>	No Links
403	Forbidden error  <i>Content</i> <b>application/json</b>	No Links
404	Not Found error  <i>Content</i> <b>application/json</b>	No Links

Code	Description	Links
500	Internal Server error  <i>Content</i> <b>application/json</b>	No Links

Type	Name	Scopes
apiKey	BearerAuth	

## Components

### Schemas

#### BaseError

BaseError

*Properties*

Name	Description	Schema
errorCode <i>optional</i>	ErrorCode	number
errorMessage <i>optional</i>	ErrorMessage	string
details <i>optional</i>	Details	< <a href="#">ErrorSchema</a> > array

#### LoginSchema

LoginSchema

*Properties*

Name	Description	Schema
email <i>required</i>	Email	string
password <i>optional</i>	Password	string

#### AuthLoginSchema

AuthLoginSchema

*Properties*

Name	Description	Schema
tokenType <i>optional</i>	Type of token	string
accessToken <i>optional</i>	Access token	string

## JwtPayloadSchema

JwtPayloadSchema

*Properties*

Name	Description	Schema
userId <i>optional</i>	UserId	string
email <i>optional</i>	Email	string
isAdmin <i>optional</i>	IsAdmin	boolean

## ErrorSchema

ErrorSchema

*Properties*

Name	Description	Schema
field <i>optional</i>	Field	string
value <i>optional</i>	Value  <i>nullable</i>	string
replyCode <i>optional</i>	ReplyCode  <i>nullable</i>	number
replyMessage <i>optional</i>	ReplyMessage  <i>nullable</i>	string

## CreateTaskSchema

CreateTaskSchema

*Properties*

Name	Description	Schema
title <i>required</i>	Title	string
userId <i>required</i>	UserId	string

Name	Description	Schema
description <i>optional</i>	Description	string
dueDate <i>optional</i>	Due date	string (date-time)
tag <i>optional</i>	Tag	string

## CreateTaskMeSchema

CreateTaskMeSchema

*Properties*

Name	Description	Schema
title <i>required</i>	Title	string
description <i>optional</i>	Description	string
dueDate <i>optional</i>	Due date	string (date-time)
tag <i>optional</i>	Tag	string

## UpdateTaskSchema

UpdateTaskSchema

*Properties*

Name	Description	Schema
title <i>optional</i>	Title  <i>nullable</i>	string
description <i>optional</i>	Description  <i>nullable</i>	string
dueDate <i>optional</i>	Due date  <i>nullable</i>	string (date-time)
isChecked <i>optional</i>	Is completed?  <i>nullable</i>	boolean
tag <i>optional</i>	Tag  <i>nullable</i>	string

## ReadTaskSchema

ReadTaskSchema

### Properties

Name	Description	Schema
id <i>optional</i>	ID	string
userId <i>optional</i>	UserId	string
title <i>optional</i>	Title	string
description <i>optional</i>	Description	string
dueDate <i>optional</i>	Due date	string (date-time)
isChecked <i>optional</i>	Is Checked	boolean
tag <i>optional</i>	Tag	string

## GetTasksWithSpecifiedTagSchema

GetTasksWithSpecifiedTagSchema

### Properties

Name	Description	Schema
tag <i>optional</i>	Tag	string

## CreateUserSchema

CreateUserSchema

### Properties

Name	Description	Schema
email <i>required</i>	Email	string
firstName <i>required</i>	First Name	string
lastName <i>required</i>	Last Name	string
password <i>required</i>	Pasword	string

## UpdateUserSchema

UpdateUserSchema

*Properties*

<b>Name</b>	<b>Description</b>	<b>Schema</b>
email <i>optional</i>	Email	string
firstName <i>optional</i>	First Name	string
lastName <i>optional</i>	Last Name	string
password <i>optional</i>	Password	string

## **ReadUserSchema**

ReadUserSchema

*Properties*

<b>Name</b>	<b>Description</b>	<b>Schema</b>
id <i>optional</i>	Id	string
email <i>optional</i>	Email	string
firstName <i>optional</i>	First Name	string
lastName <i>optional</i>	Last Name	string
isAdmin <i>optional</i>	IsAdmin	boolean