

TodoList 1조

장철희, 최명윤

Appcenter - 15.5기

프론트엔드

Develop by 최명운

화면 구성

- 로그인, 회원 가입, 유저 정보
- Todo 목록을 표시하는 home 화면
- Todo 상세 화면
- Todo 추가, 수정 화면

회원가입

```
navController.navigate(A1101.Login.name)
} else if (response.code() == 400) {
    if (pwTextState.value.length < 6)
        Toast.makeText(context, "비밀번호는 6자리 이상이어야 합니다.", Toast.LENGTH_SHORT)
    Log.e("err", response.errorBody()?.toString())
}
```

- 6자리 비밀번호 검사

회원가입

```
145
146     fun checkIdEmail(id: MutableState<String>): Boolean {
147         return "@" in id.value && id.value.count { it == '@' } == 1
148     }
149
```

- 이메일 형식 검사

유저 정보

```
10     class UserViewModel : ViewModel() {  
11         private val _userId: MutableState<Int> = mutableStateOf(0)  
12         val userId: State<Int>  
13             get() = _userId  
14  
15         private val _name: MutableState<String> = mutableStateOf("")  
16         val name: State<String>  
17             get() = _name  
18  
19         private val _loginId: MutableState<String> = mutableStateOf("")  
20         val loginId: State<String>  
21             get() = _loginId
```


유저 정보

```
23     private val _password: MutableState<String> = mutableStateOf("")
24     val password: State<String>
25         get() = _password
26
27     fun getUserInfo(response: Response<UserResponseDTO>) {
28         _userId.value = response.body()?.userId!!
29         _name.value = response.body()?.name.toString()
30         _loginId.value = response.body()?.loginId.toString()
31         _password.value = response.body()?.password.toString()
32     }
33 }
```

유저 정보

- 로그인 시 가져온 유저 정보 저장

Todo data

```
12     class TaskViewModel : ViewModel() {  
13         private val _todoList: MutableState<List<TaskResponseDTO>> = mutableStateOf(muta  
14         val todoList: MutableState<List<TaskResponseDTO>>  
15             get() = _todoList  
16  
17         private val _todoItem: MutableState<TaskResponseDTO> = mutableStateOf(TaskRespon  
18         val todoItem: MutableState<TaskResponseDTO>  
19             get() = _todoItem
```

- Todo list와 Todo 항목 하나를 저장

Todo data

```
21     fun getAllUsersTask(userViewModel: UserViewModel) {
22         val taskAPI = RetrofitBuilder.getInstanceFor().create(TaskControl::class.java)
23         val result = taskAPI.getAllTask(userViewModel.userId.value)
24         result.enqueue(object : Callback<List<TaskResponseDTO>> {
25             override fun onResponse(
26                 call: Call<List<TaskResponseDTO>>,
27                 response: Response<List<TaskResponseDTO>>
28             ) {
29                 if (response.isSuccessful && response.code() == 200) {
30                     _todoList.value = response.body()!!
31                 }
32             }
33         })
```

- 사용자의 Todo 전체를 가져오는 메서드

Todo data

```
39
40     fun storeTaskInfo(item: TaskResponseDTO) {
41         _todoItem.value = item
42     }
43 }
```

- Todo 항목 하나에 대한 정보 저장
- Todo 상세페이지에서 사용될 정보

극복해야 했던 과제

- Todo 항목이 추가 되고 삭제될 때 UI처리
- 즉각적으로 적용되지 않는 문제

극복해야 했던 과제

- Todo와 사용자에 대한 data 접근
- 화면 간에 data 전송

백엔드

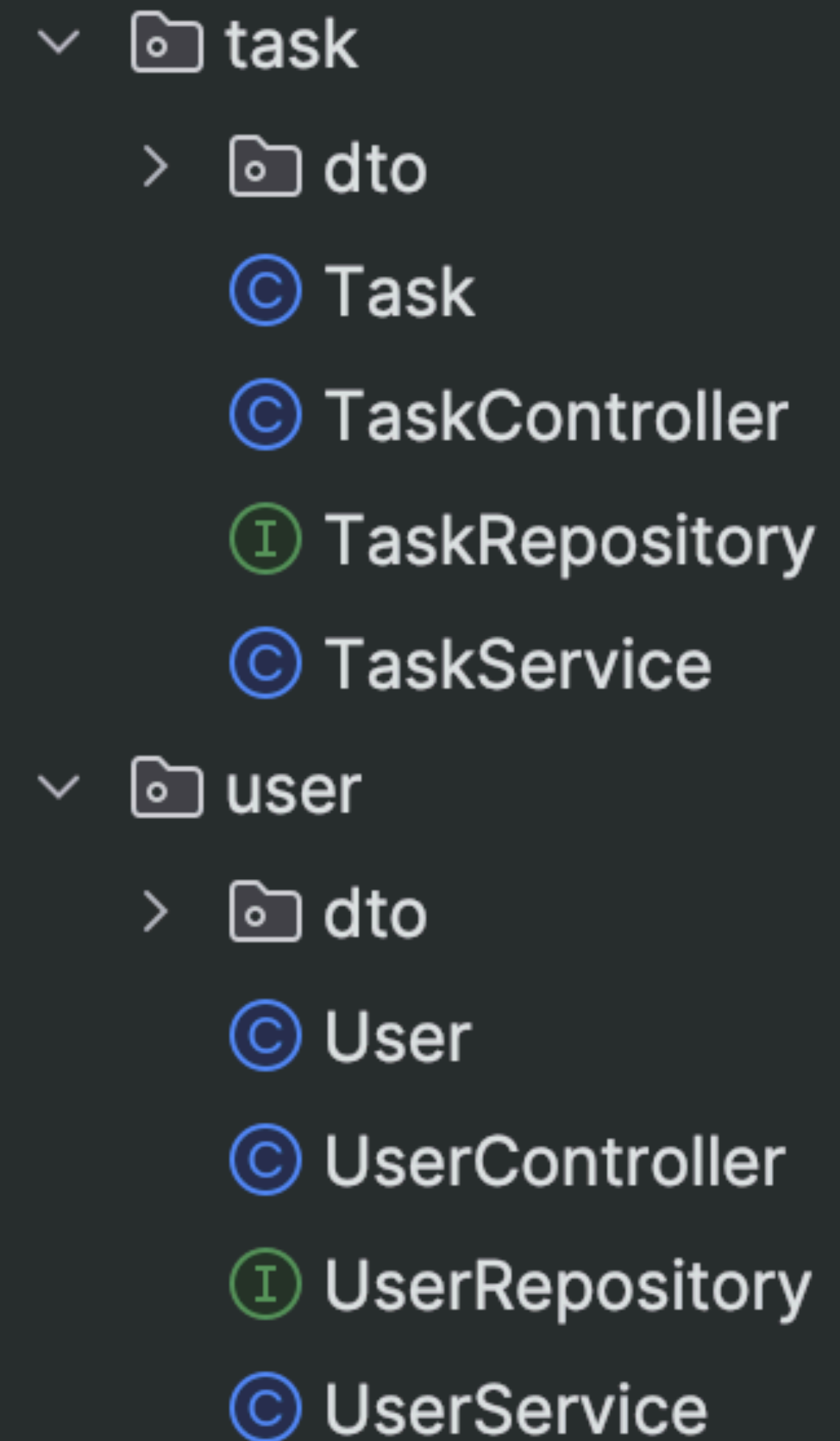
Develop by 장철희

목표

- 구현 상세코드의 품질
- 깔끔하고 이해하기 쉬운 코드
- 추후 확장가능성을 염두해둔 코드

Entity

- Task(할 일)와 User 도메인 추출
- User는 여러 개의 Task를 갖는다.



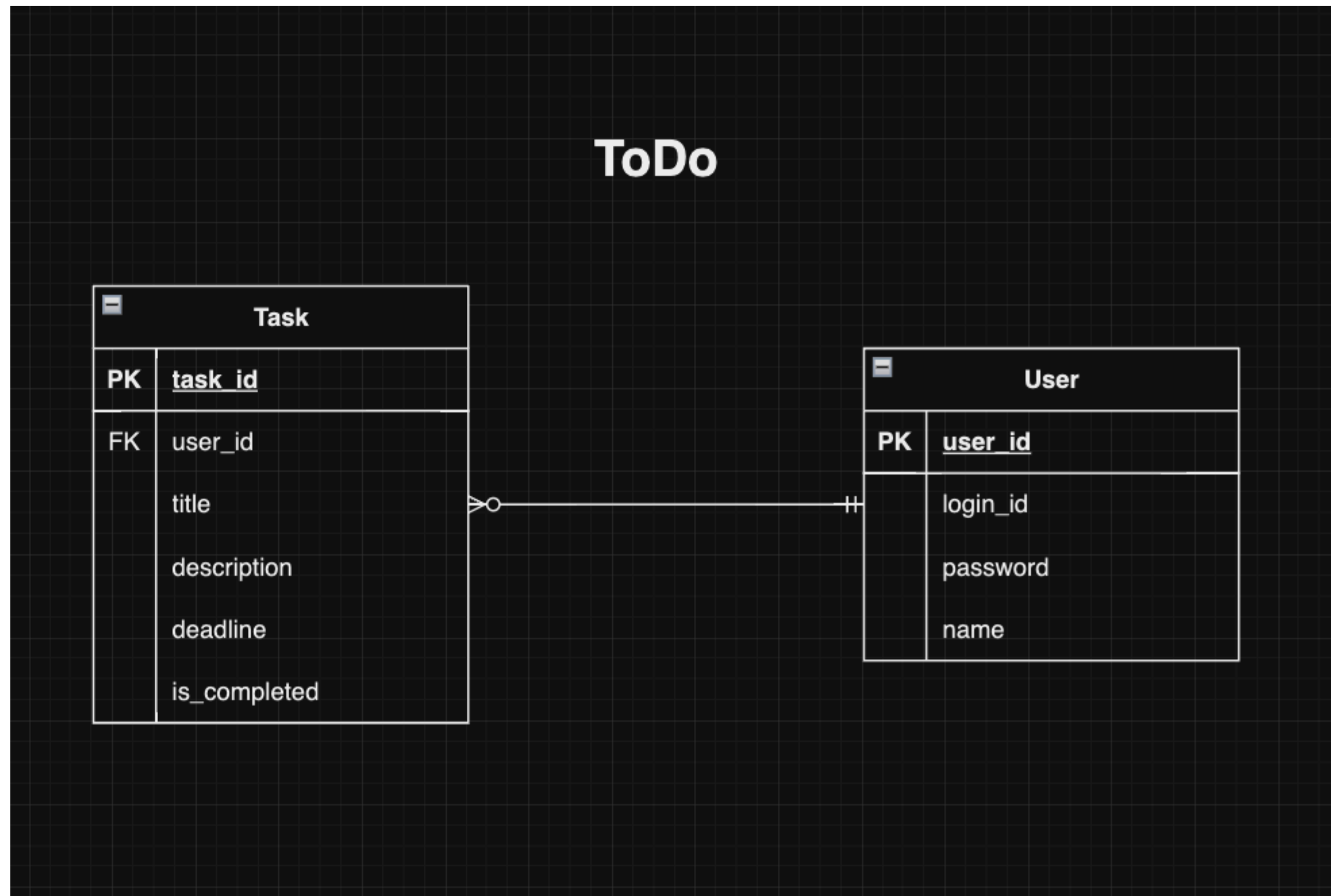
```

  task
  > dto
    Task
    TaskController
    IRepository TaskRepository
    TaskService
  user
  > dto
    User
    UserController
    IRepository UserRepository
    UserService

```


The image shows a file explorer view of a project structure. It is divided into two main sections: 'task' and 'user'. Each section contains a 'dto' folder and several files. The 'task' section includes 'Task', 'TaskController', 'TaskRepository' (marked with a green 'I' icon), and 'TaskService'. The 'user' section includes 'User', 'UserController', 'UserRepository' (marked with a green 'I' icon), and 'UserService'. The files are marked with icons: a blue circle with a 'C' for controllers and services, and a green circle with an 'I' for interfaces.

테이블 설계



- User와 Task가 1대다의 연관관계
- Task의 is_completed 필드는 완료여부 의미

Swagger

 swagger

Select a spec default

ToDo Rest API 1.0.0

[Base URL: cheolhui.na2ru2.me/]
<http://cheolhui.na2ru2.me/v2/api-docs>

앱센터 15.5기 ToDo List API

Task Task Controller

POST

/tasks 새 Task 등록 api

PUT

/tasks Task 수정 api

DELETE

/tasks/{taskId} 기존 Task 삭제 api

GET

/tasks/{userId} User의 모든 Task 조회 api

PATCH

/tasks/completed/{taskId} Task 완료 상태 변경 api

User User Controller

POST

/users 새 User 등록 api

- API 명세 및 테스트 용도

Swagger

User User Controller

POST

/users

새 User 등록 api

새로운 유저 정보 등록

Parameters

Cancel

Name	Description
<div><div>userRequestDto</div><div><div>★ required</div></div><div>(body)</div></div>	<div>userRequestDto</div> <div><div>Example Value</div><div>Model</div></div> <div><pre>{ "loginId": "namu0007@gmail.com", "name": "jang-namu", "password": "encrypt", "userId": null}</pre></div> <div><div>Cancel</div></div> <div><div>Parameter content type</div><div>application/json</div></div>

Execute

Code

Details

201

Response body

```
{  "userId": 21,  "loginId": "namu0007@gmail.com",  "password": "encrypt",  "name": "jang-namu"}
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidateconnection: keep-alivecontent-type: application/jsondate: Fri, 29 Dec 2023 04:57:30 GMTexpires: 0pragma: no-cacheserver: nginxstrict-transport-security: max-age=63072000; includeSubdomains; preloadtransfer-encoding: chunkedx-content-type-options: nosniffx-frame-options: DENYx-xss-protection: 1; mode=block
```

Responses

- Test: User 등록

Swagger

POST

/users/login

User 로그인 api

loginId, password 로그인

Parameters

Cancel

Name

Description

userRequestDto * required

(body)

userRequestDto

Example Value | Model

{
 "loginId": "namu0007@gmail.com",
 "name": "jang-namu",
 "password": "encrypt",
 "userId": 0
}

Cancel

Parameter content type
application/json

Execute

- Test: User 정보조회(로그인)

Code

Details

200

Response body

{
 "userId": 21,
 "loginId": "namu0007@gmail.com",
 "password": "encrypt",
 "name": "jang-namu"
}

Download

Response headers

cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Fri, 29 Dec 2023 04:58:33 GMT
expires: 0
pragma: no-cache
server: nginx
strict-transport-security: max-age=63072000; includeSubdomains; preload
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block

Swagger

POST

/tasks 새 Task 등록 api

새 Task 등록

Parameters

Cancel

Name

taskRequestDto * required
(body)

Description

taskRequestDto

Example Value

Model

```
{  "deadline": "2023-12-29T05:00:11.028Z",  "description": "2024년은 행복합니다.",  "taskId": 0,  "title": "연말 잘 보내세요!",  "userId": 21}
```

Cancel

Parameter content type

application/json

Execute

- Test: Task 생성

201

Response body

```
{  "taskId": 34,  "title": "연말 잘 보내세요!",  "description": "2024년은 행복합니다.",  "deadline": [    2023,    12,    29,    5,    0,    11,    28000000  ]}
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Fri, 29 Dec 2023 05:00:38 GMT
expires: 0
pragma: no-cache
server: nginx
strict-transport-security: max-age=63072000; includeSubdomains; preload
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Responses

Swagger

GET

/tasks/{userId}

User의 모든 Task 조회 api

특정 유저의 모든 Task 조회

Parameters

Cancel

Name	Description
userId <small>* required</small> integer(\$int64) <i>(path)</i>	User 식별자

21

200

Execute

Response body

```
[
  {
    "taskId": 34,
    "title": "연말 잘 보내세요!",
    "description": "2024년은 행복합시다.",
    "deadline": [
      2023,
      12,
      29,
      5,
      0,
      11
    ]
  },
  {
    "taskId": 35,
    "title": "새해 복 많이받으세요",
    "description": "2024년은 행복합시다.",
    "deadline": [
      2023,
      12,
      29,
      5,
      0,
      11
    ]
  }
]
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-type: application/json
date: Fri, 29 Dec 2023 05:01:35 GMT
expires: 0
pragma: no-cache
server: nginx
strict-transport-security: max-age=63072000; includeSubdomains; preload
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

- Test: (Task 추가 생성 후) 조회

어려운점

- DTO를 어떻게 나눠야하는지 -> wiki
- 예외처리는 어디서 해야하는지 -> Handler, ControllerAdvice
- Test 코드 작성