

Lv.1 (API 명세서)

기능 Method URL Request Body Response Body 상태 코드

기능	메서드	URL	Request (JSON)	Response (JSON)	상태 코드
일정 등록	POST	/schedules	요청 body	등록 정보	201 Created, 400 잘못된 요청
일정 단건 조회	GET	/schedules/{id}	요청 param	단건 응답 정보	200 OK, 404 일정 없음
일정 전체 조회	GET	/schedules/	요청 param	다건 응답 정보	200 OK
일정 수정	PUT	/schedules/{id}	요청 body, 요청 param	수정 정보	200 OK, 403 Forbidden, 404 일정 없음, 400 잘못된 요청
일정 삭제	DELETE	/schedules/{id}	-	-	200 OK, 204 No Content, 403 Forbidden, 404 일정 없음

일정 등록

- Request Body

필드명	타입	설명
content	String	등록 내용
title	String	제목
userName	String	작성자 이름
password	String	비밀번호

```
{
  "content": "강의 1주차 듣기",
  "title": "제목",
  "userName": "홍길동",
  "password": "1234"
}
```

- Response Body

필드명	타입	설명
scheduleId	Long	스케줄 고유 ID
contents	String	등록 내용

userName	String	작성자 이름
createdAt	Date	작성일 (YYYY-MM-DD)
updatedAt	Date	수정일 (YYYY-MM-DD)

Sample Response

성공:

```
{
  "message": "일정이 정상적으로 등록되었습니다.",
  "scheduleId": 1,
  "contents": "강의 1주차 듣기",
  "userName": "홍길동",
  "createdAt": "2025-05-19",
  "updatedAt": "2025-05-19"
}
```

실패(400):

```
{
  "error": "내용을 입력해주세요."
}
```

일정 단건 조회

- Request Param

변수명	타입	설명
scheduleId	String	등록 내용

Sample Request

GET api/schedules/1

- Response Body

필드명	타입	설명
scheduleId	Long	스케줄 고유 ID
title	String	글 제목
contents	String	등록 내용
username	String	작성자 이름
createdAt	Date	작성일 (YYYY-MM-DD)
updatedAt	Date	수정일 (YYYY-MM-DD)

Sample Response

성공:

```
{
  "scheduleId": 1,
  "title": "제목",
  "contents": "강의 1주차 듣기",
  "userName": "홍길동",
  "createdAt": "2025-05-21",
  "updatedAt": "2025-05-21"
}
```

실패(404):

```
{
  "error": "해당 일정이 존재하지 않습니다."
}
```

일정 전체 조회

- Request Param

변수명	타입	설명
<code>scheduleId</code>	<code>String</code>	등록 내용

Sample Request

`GET /api/schedules?userName=홍길동`

`GET /api/schedules?title=강의`

- Response Body

필드명	타입	설명
<code>scheduleId</code>	<code>Long</code>	스케줄 고유 ID
<code>title</code>	<code>String</code>	글 제목
<code>contents</code>	<code>String</code>	등록 내용
<code>userName</code>	<code>String</code>	작성자 이름
<code>createdAt</code>	<code>Date</code>	작성일 (YYYY-MM-DD)
<code>updatedAt</code>	<code>Date</code>	수정일 (YYYY-MM-DD)

Sample Response

성공:

```
{
  "scheduleId": 1,
  "title": "제목",
  "contents": "강의 1주차 듣기",
  "userName": "홍길동",
  "createdAt": "2025-05-21",
  "updatedAt": "2025-05-21"
}
```

실패(404):

```
{
  "error": "일정이 존재하지 않습니다."
}
```

일정 수정

- Request Param

변수명	타입	설명
<code>scheduleId</code>	<code>String</code>	수정할 스케줄 고유 ID

Sample Request

`PUT /api/schedules/5`

Request Body

필드명	타입	설명
<code>title</code>	<code>String</code>	수정할 글 제목
<code>contents</code>	<code>String</code>	수정할 등록 내용
<code>password</code>	<code>String</code>	비밀번호
<code>updatedAt</code>	<code>Date</code>	(기본값)

Sample Request

```
{
  "title": "수정된 제목",
  "content": "수정된 내용",
  "password": "1234"
}
```

Response Body

필드명	타입	설명
<code>scheduleId</code>	<code>Long</code>	스케줄 고유 ID
<code>title</code>	<code>String</code>	글 제목
<code>contents</code>	<code>String</code>	등록 내용
<code>updatedAt</code>	<code>Date</code>	수정일 (YYYY-MM-DD)

성공:

```
{
  "message": "일정이 정상적으로 수정되었습니다.",
  "scheduleId": 1,
  "title": "수정된 제목",
  "contents": "수정된 내용",
  "updatedAt": "2025-05-22"
}
```

실패(403):

```
{
  "error": "비밀번호가 일치하지 않습니다."
}
```

실패(404):

```
{
  "error": "해당 ID의 일정이 없습니다."
}
```

일정 삭제

Request Param

변수명	타입	설명
<code>scheduleId</code>	<code>long</code>	삭제할 스케줄 고유 ID

Sample Request

`DELETE api/schedules/2`

- Request Body

필드명	타입	설명
-----	----	----

password	String	비밀번호
----------	--------	------

Sample Request

```
{  
  "password": "1234"  
}
```

- Response Body

-

성공(200):

```
{  
  "message": "일정이 정상적으로 삭제되었습니다."  
}
```

실패(403):

```
{  
  "error": "비밀번호가 일치하지 않습니다."  
}
```

실패(404):

```
{  
  "error": "해당 ID의 일정이 없습니다."  
}
```