

# Spring 1차과제(일정관리)

## ▼ 테이블 정보 [ERD]

### ▼ schedule

- 일정 테이블

### ▼ writer

- 작성자 테이블

### ▼ 캡처본참고

- 



## ▼ API명세서

### ▼ 일정 테이블

- ▼ 일정 등록

- POST
- /api/schedule
- 요청 body
- 등록정보
- 200 정상등록
- ▼ 일정 전체조회
  - GET
  - /api/schedule
  - 요청 param
  - 다건 응답정보
  - 200 정상조회
- ▼ 일정 검색
  - GET
  - /api/schedule/search
  - 요청 param
  - 다건 응답정보
  - 200 정상조회
- ▼ 일정검색 - 등록된 작성자 목록만 조회
  - GET
  - /api/schedule/wList
  - 요청 param
  - 다건 응답정보
  - 200 정상조회
- ▼ 일정 수정
  - PUT
  - /api/schedule/{id}
  - 요청 body
  - 수정 정보
  - 200 정상수정
- ▼ 일정 삭제

- DELETE
- /api/schedule/{id}
- 요청 param
- -
- 200 정상삭제
- ▼ 현재 데이터 개수(페이징세팅)
  - GET
  - /api/schedule/listcount
  - 요청 param
  - 단건 응답정보
  - 상태 200 정상조회
- ▼ 작성자 테이블
  - ▼ 일정등록 - 작성자 목록 조회
    - GET
    - /api/writer
    - 요청 param
    - 다건 응답정보
    - 200 정상조회
  - ▼ 작성자 등록
    - POST
    - /api/writer
    - 요청 body
    - 등록정보
    - 200 정상등록
- ▼ 캡처본참고

[API명세서-일정관리]		Overview	Authorization	Scripts	Tests	Variables	Runs
POST /api/schedule - 일정등록	...	2. 일정 등록 - 작성자 목록 조회 - /api/writer					
GET /api/schedule - 일정전체조회		<ul style="list-style-type: none"> <li>일정등록 작성자 select박스에 작성자목록 조회</li> </ul>					
PUT /api/schedule/{id} - 일정수정		3. 작성자 등록 - /api/writer					
DEL /api/schedule/{id} - 일정삭제		<ul style="list-style-type: none"> <li>작성자 등록 - 일정등록시에 select로 선택 가능하도록</li> </ul>					
GET /api/schedule/search - 일정검색		4. 일정 전체조회 - /api/schedule					
GET /api/schedule/wlist - 등록된작...		<ul style="list-style-type: none"> <li>페이지 로딩되자마자 일정 전체 데이터 조회</li> </ul>					
GET /api/schedule/listcount - 일정 ...		5. 일정 검색조회 - /api/schedule/search					
GET /api/writer - 일정등록(작성자...		<ul style="list-style-type: none"> <li>날짜, 작성자, 일정내용 3개 검색항목을 만들었음.</li> <li>날짜는 일수를 활용해서 1일, 일주일, 1개월, 3개월로 범위 검색</li> <li>작성자는 일정테이블에서 등록된 작성자목록만 select박스로 쿼리를 따로 만들어서 조회해서 등록된 작성자목록만 검색할 수 있음</li> </ul>					
POST /api/writer - 작성자 등록		6. 일정 검색 - 등록된 작성자 목록 조회 - /api/schedule/wList					
		<ul style="list-style-type: none"> <li>상단 SELECT박스에 일정에 등록된 작성자 목록만 조회</li> </ul>					
		7. 일정 수정 - /api/schedule/{idx}					
		<ul style="list-style-type: none"> <li>등록된 비밀번호와 일치해야 수정 가능</li> <li>수정시 수정일은 현재시간으로 수정</li> <li>일정내용과 작성자만 수정 가능</li> </ul>					
		8. 일정 삭제 - /api/schedule/{idx}					
		<ul style="list-style-type: none"> <li>선택한 일정 삭제</li> <li>등록된 비밀번호와 일치해야 삭제 가능</li> </ul>					
		9. 일정 데이터 개수 추출 - /api/schedule/listcount					
		<ul style="list-style-type: none"> <li>페이징영역 생성을 위한 데이터 개수 추출</li> </ul>					

## ▼ 요구사항

### ▼ 레벨1

- API명세서작성
- ERD작성
- SQL작성

### ▼ 레벨2

#### ▼ 일정 생성

- 할일, 작성자명, 비밀번호, 작성일, 수정일 저장
- 작성 수정일은 DATETIME형태
- ▼ 최초입력시 작성일과 수정일은 동일한 시간으로 남도록
  - 등록될때 수정일도 함께 등록
- ▼ 일정 ID는 자동으로 생성 및 관리
  - 기본키로 관리

#### ▼ 일정 조회

- ▼ 전체일정조회
  - 수정일, 작성자명 목록에 보이도록

- 수정일 기준 내림차순 정렬
- ▼ 선택일정조회
  - 일정 ID를 기준으로 선택 조회되도록
- ▼ 레벨3
  - ▼ 일정 수정
    - 비밀번호 입력하면 수정할 수 있도록
    - 내용과 작성자명만 수정 가능
  - ▼ 일정 삭제
    - 비밀번호 입력하면 삭제할 수 있도록
- ▼ 레벨4
  - 작성자와 일정연결
- ▼ 레벨5
  - 페이징 기능 넣기
  - Paging 객체 활용해보기