

# 14조 프로젝트 발표

201902531 김명준

# 프로젝트 설명

## 프로젝트 개요

- **일정 관리 기능이 포함된 웹캘린더**
- 사용자가 **달력에서 날짜를 선택**해 일정을 등록, 조회, 수정, 삭제 가능
- 수업시간에 배운 내용을 바탕으로 CRUD 기능 구현
- **CRUD(Create, Read, Update, delete)기능을 가지고 있지만**, 일반적인 게시판이 아닌 Mustache로 만든 달력을 이용하는 방식으로 구현

## 차별화된 포인트

- 일반적인 게시판 구조가 아닌 달력으로 이루어진 프로젝트
- 달력에 일정 등록시 반복 주기를 사용자가 일 단위로 자율 설정 가능
- 반복 일정은 달력에 자동 계산되어 **시각적으로 표시됨**

# 프로젝트 설명

📅 일정 관리

2025년 5월

이동

< 이전

일	월	화	수
4	5	6	7
11	12	13	14
18	19	20	21
25	26	27	28

📅 일정 관리

새 일정 작성

제목

내용

날짜

2025-05-26

반복 주기 (일)

반복 시작일

연도-월-일

반복 종료일

연도-월-일

저장

달력으로 돌아가기

📅 일정 관리

2025년 5월

이동

< 이전

일	월	화	수	목	금	토
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

다음 >

26

프로젝트 발표

# 구현 내용

## 일정 등록 기능 (Create)

```
@PostMapping("/schedules/create") 0개의 사용위치
public String create(ScheduleForm form) {
    log.info(form.toString());
    Schedule schedule = new Schedule(
        id: null,
        form.getTitle(),
        form.getContent(),
        form.getDate(),
        form.getRepeatIntervalDays(),
        form.getRepeatStartDate(),
        form.getRepeatEndDate()
    );

    scheduleRepository.save(schedule);
    return "redirect:/schedules/day?date=" + form.getDate();
}
```

- 사용자가 제목이나 내용을 입력해서 폼 제출
- Schedule이라는 객체에 저장됨
- 저장은 ScheduleRepository.save(schedule)로 처리 됨
- 저장이 끝나면 다시 해당 날짜로 돌아가서 등록된 일정 확인 가능

## 일정 조회 기능(Read)

```
Map<LocalDate, List<Schedule>> scheduleMap = new HashMap<>();
for (Schedule s : scheduleRepository.findAll()) {
    if (s.getRepeatIntervalDays() == null || s.getRepeatStartDate() == null || s.getRepeatEndDate() == null) {
        scheduleMap.computeIfAbsent(s.getDate(), LocalDate k -> new ArrayList<>()).add(s);
    } else {
        for (LocalDate d = s.getRepeatStartDate();
            !d.isAfter(s.getRepeatEndDate());
            d = d.plusDays(s.getRepeatIntervalDays())) {
            scheduleMap.computeIfAbsent(d, LocalDate k -> new ArrayList<>()).add(s);
        }
    }
}
```

- 달력 화면을 구성하는 핵심 기능
- 날짜 별로 일정을 분류해서 달력에 표시할 수 있도록 함
- 반복 일정은 시작일과 종료일 사이에서 며칠마다 반복할지를 계산해서 해당 날짜에 반복 일정도 자동으로 표시되도록 구성함

# 구현 내용

## 일정 수정 기능 (Update)

```
@GetMapping("/schedules/{id}/edit") 0개의 사용위치
public String edit(@PathVariable Long id, Model model) {
    Schedule schedule = scheduleRepository.findById(id).orElse( other: null);
    model.addAttribute( attributeName: "schedule", schedule);
    return "schedules/edit";
}

@PostMapping("/schedules/update") 0개의 사용위치
public String update(ScheduleForm form, @RequestParam Long id) {
    Schedule schedule = scheduleRepository.findById(id).orElse( other: null);
    if (schedule != null) {
        schedule.setTitle(form.getTitle());
        schedule.setContent(form.getContent());
        schedule.setDate(form.getDate());
        scheduleRepository.save(schedule);
    }
    return "redirect:/schedules/day?date=" + form.getDate();
}
```

- edit() 메서드를 통해 기존 일정을 화면에 띄움
- 사용자가 내용을 수정한 뒤에는 update() 메서드로 저장되도록 구성
- 이때도 scheduleRepository.save()를 사용해 수정된 내용을 다시 저장.

## 일정 삭제 기능(Delete)

```
@GetMapping("/schedules/{id}/delete") 0개의 사용위치
public String delete(@PathVariable Long id) {
    Schedule schedule = scheduleRepository.findById(id).orElse( other: null);
    if (schedule != null) {
        LocalDate date = schedule.getDate();
        scheduleRepository.deleteById(id);
        return "redirect:/schedules/day?date=" + date;
    }
    return "redirect:/schedules/calendar";
}
```

- 특정 일정을 삭제할 때는 deleteById() 메서드를 사용해서 삭제
- 삭제가 끝나면 달력 화면으로 다시 돌아가게 되도록 구성

# 구현 내용

```
{{>layouts/header}}
<h2>새 일정 작성</h2>
<form action="/schedules/create" method="post" class="p-3">
  <div class="mb-3">
    <label class="form-label">제목</label>
    <input name="title" class="form-control" required>
  </div>
  <div class="mb-3">
    <label class="form-label">내용</label>
    <textarea name="content" class="form-control" rows="3"></textarea>
  </div>
  <div class="mb-3">
    <label class="form-label">날짜</label>
    <input name="date" type="date" value="{{form.date}}" class="form-control" required>
  </div>
  <div class="mb-3">
    <label class="form-label">반복 주기 (일)</label>
    <input name="repeatIntervalDays" type="number" class="form-control" required>
  </div>
  <div class="mb-3">
    <label class="form-label">반복 시작일</label>
    <input name="repeatStartDate" type="date" class="form-control" required>
  </div>
  <div class="mb-3">
    <label class="form-label">반복 종료일</label>
    <input name="repeatEndDate" type="date" class="form-control" required>
  </div>
  <button type="submit" class="btn btn-primary">저장</button>
</form>
<a href="/schedules/calendar" class="btn btn-secondary mt-2">
  <img alt="calendar icon" data-bbox="30 770 45 785"/>
</a>
{{>layouts/footer}}
```

```
{{>layouts/header}}
<h2>{{date}} 일정 목록</h2>
<a href="/schedules/new?date={{date}}" class="btn btn-primary">
  <img alt="plus icon" data-bbox="291 535 306 550"/>
</a>
<ul class="list-group">
  {{#schedules}}
    <li class="list-group-item">
      <div class="d-flex justify-content-between">
        <strong>{{title}}</strong>
        <span>
          <a href="/schedules/{{id}}/edit" class="btn btn-sm btn-outline-secondary">수정</a>
          <a href="/schedules/{{id}}/delete" class="btn btn-sm btn-outline-danger">삭제</a>
        </span>
      </div>
      {{#content}}
        <div class="text-muted small mt-2">{{content}}</div>
      {{/content}}
    </li>
  {{/schedules}}
  {{^schedules}}
    <li class="list-group-item">일정이 없습니다.</li>
  {{/schedules}}
</ul>
<a href="/schedules/calendar" class="btn btn-secondary mt-3">
  <img alt="calendar icon" data-bbox="291 955 306 970"/>
  달력으로 돌아가기</a>
{{>layouts/footer}}
```

```
<table class="table calendar-table text-center">
  <thead class="table-light">
    <tr>
      <th>일</th><th>월</th><th>화</th><th>수</th><th>목</th><th>금</th><th>토</th>
    </tr>
  </thead>
  <tbody>
    {{#calendarRows}}
      <tr>
        {{#this}}
          <td class="align-top calendar-cell p-2 {{#isOtherMonth}}bg-light text-white{{/isOtherMonth}}">
            {{#isCurrentMonth}}
              <a href="/schedules/day?date={{date}}" class="text-decoration-none">
                <div class="fw-bold {{#isToday}}text-danger border border-danger rounded-circle p-1{{/isToday}}">
                  {{day}}
                </div>
              </a>
              {{#schedules}}
                <div class="badge bg-info text-dark text-start text-truncate d-block mb-1">{{title}}</div>
              {{/schedules}}
            </td>
          {{/this}}
        </tr>
      </tbody>
    </table>
```

# 느낀점과 향후 계획

## 느낀점

- CRUD 기능만 구현하는게 아니라 사용자가 쓰기 편한 기능을 많이 생각해봄
- 반복 일정이나 날짜 계산 등 수업 시간에 배운 것 말고 새로운 개념을 적용해보며 실력 향상
- 다 완성하고 나니 뿌듯함

## 향후 계획

- 일정 카테고리 및 색상 구분
- 통계 기능(요일별 일정 분포)
- 디스코드 등 외부 알림 기능 연동