JOBARA

구인구직사이트

2조

조장 :

조원 :

CONTENTS

CONTENTS

01

구인 구직 사이트 만들기

01. 프로젝트 소개

목표

프로젝트 목표

- [ ] 팀별 협업을 통해 코드 컨벤션의 중요성 알아가기
- [ ] JAVA 언어 개념잡기
- [ ] 객체지향 프로그래밍 기법 개념잡기(추상화, 상속 등)
- [ ] CRUD, 트랜잭션, MVC패턴, Ajax, SQL(서브쿼리, 조인), DTO, 유효성 체크 개념 잡기
- [ ] IoC, DI(의존성주입), MariaDB, MyBatis, JSTL문법, 개념 잡기
- [ ] Session, Cookie 개념 잡기
- [ ] Http 프로토콜 개념 잡기 (ex, Content-Type, MIME Type)
- [ ] HTTP 통신 메서드(GET,POST,DELETE,PUT) 개념 잡기
- [ ] 템플릿 엔진(JSP) 개념 잡기
- [ ] Redis 개념 잡고 구현하기

02. 개발 환경

- IDE

- language

- VCS

- DB

- Framework

03. 코드 컨밴션

클래스/변수/패키지

클래스명

- 파스칼표기법 사용
- 가변적인 부분이 앞으로 온다. (EX) UserController)
- 뒷 부분은 공통적인 부분이 온다.

변수명

- 카멜표기법 사용
- DB에서 불러온 데이터는 변수명 뒤에 PS를 붙인다.
- 세션에서 불러온 유저 정보는 principal 사용한다.
- List형 변수의 경우 기존 변수명 뒤에 List를 붙인다.(Dtos X)
- Dto의 타입을 사용하는 경우 Dto를 붙인다.

폴더(패키지)명

- 소문자로만 작성한다. (카멜표기법 X)

JSP 파일

- 카멜 표기법 사용

03. 코드 컨밴션

메소드

메소드명

- 카멜표기법 사용
- 동사가 앞으로 온다.
- 최대 글자는 40자

Controller

- 메소드명은 엔드포인트를 사용한다.

Service

- 메소드명은 단순 select는 접두사 get사용(ex) getList)
- update는 접두사 update 사용, delete는 접두사 delete 사용
- insert가 사용되는 서비스는 접두사 insert 사용

Repository

- 메소드명은 select는 find, update는 접두사 update 사용, delete는 접두사 delete 사용, insert가 사용되는 서비스는 접두사 insert 사용
- by는 where절에 들어가는 컬럼에 사용
- and는 where절에 들어가는 and를 사용하는 경우 사용
- or는 where절에 들어가는 or를 사용하는 경우 사용
- with는 join이 된 테이블을 적을 때 사용
- 매개 변수가 두가지 이상일 경우 Entity 사용

DI(의존성 주입)

- @AutoWired 사용

03. 코드 컨밴션

테스트

테스트 클래스 / 메소드

클래스명

- 기존의 클래스에서 접미사 Test를 붙인다.

메소드명

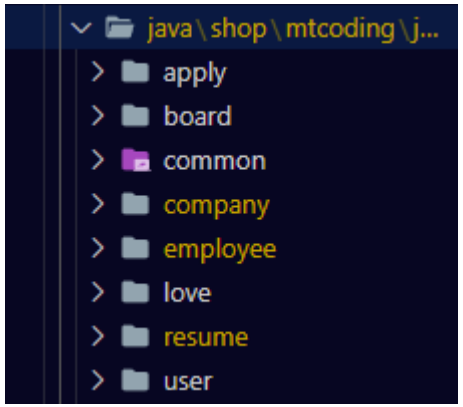
- 기존의 메소드에서 접미사 _test를 붙인다.

04. 협업 전략

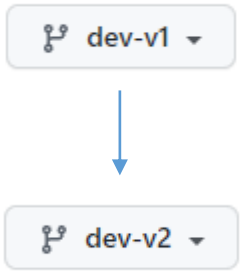
Git 전략

69 branches

기능 별로!



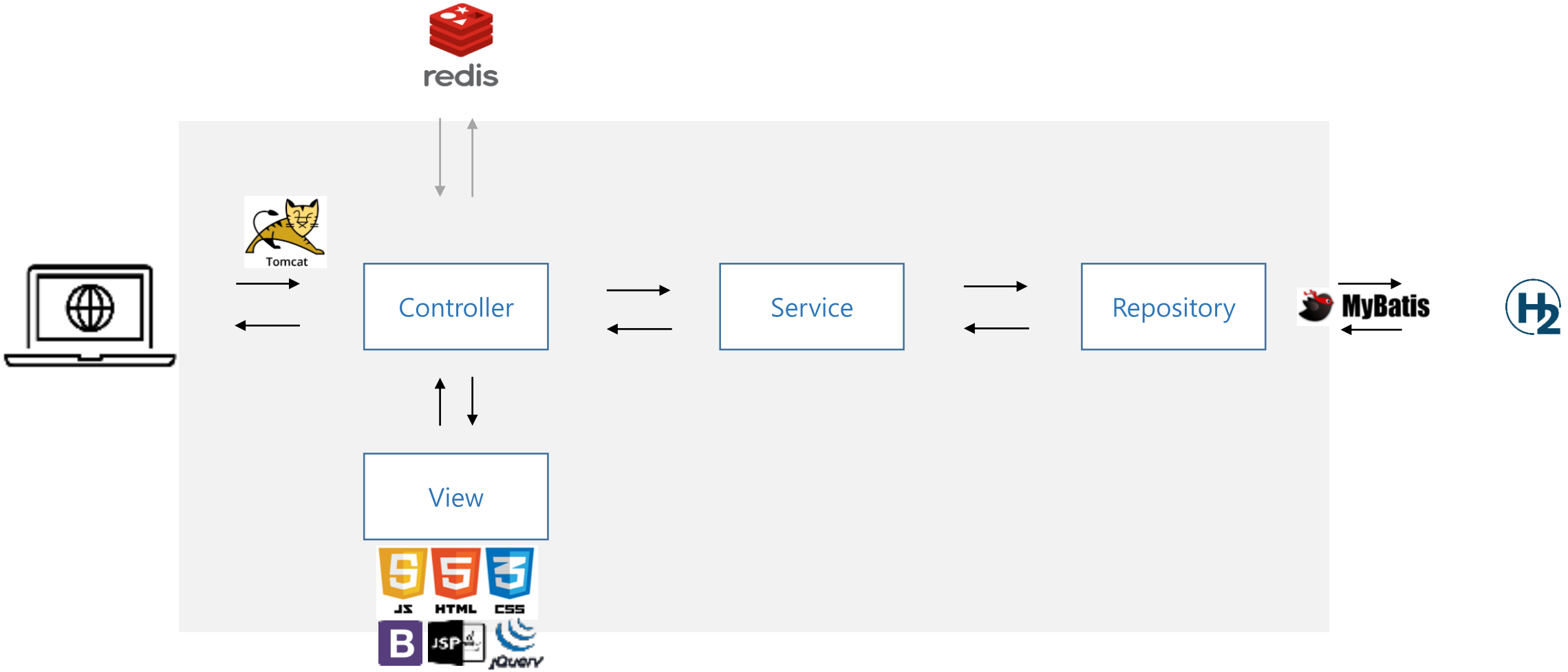
도메인 별로!



대규모 패치!

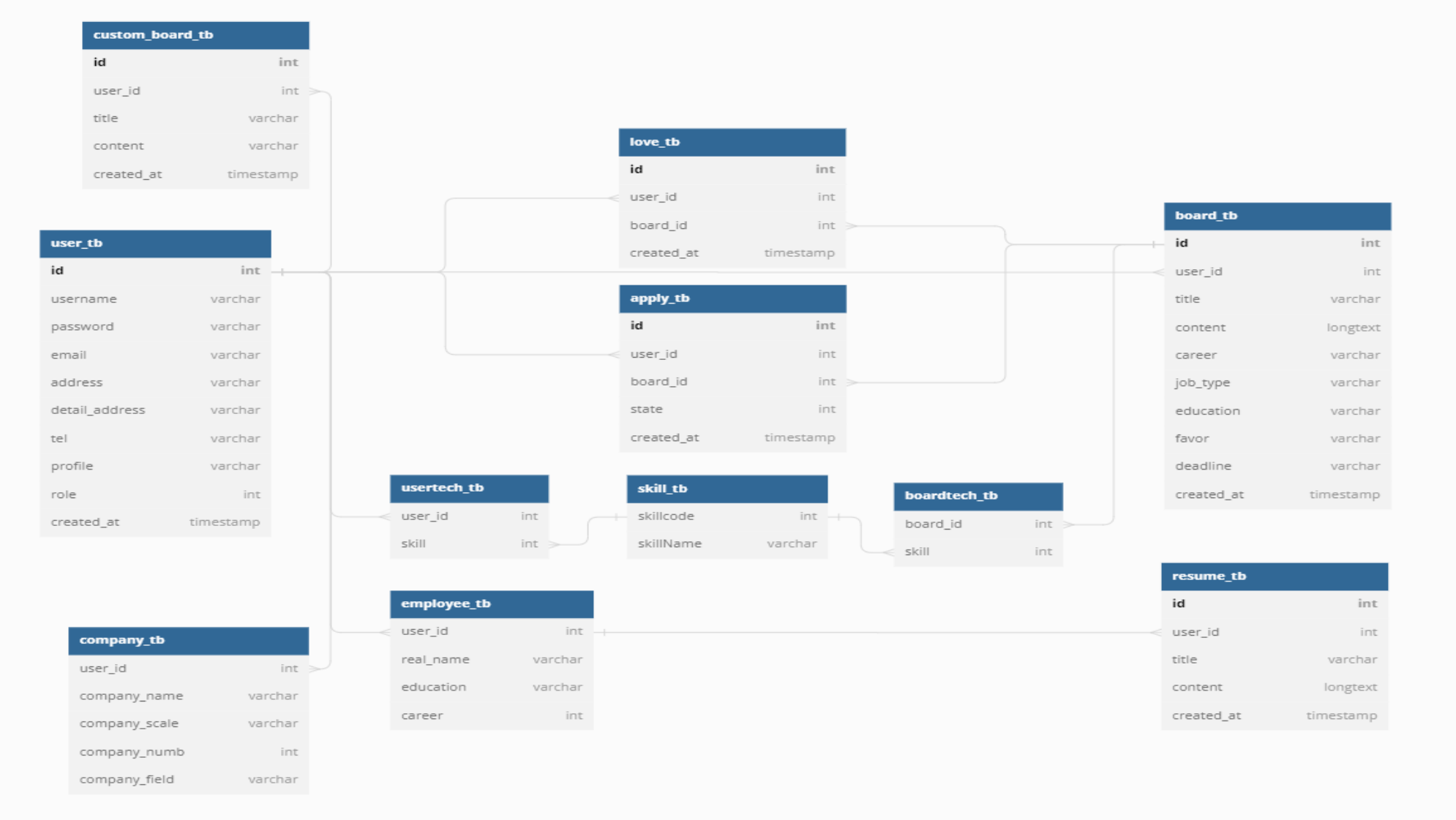
05. 프로젝트 아키텍쳐

Text



06. 테이블 설계

table



07. 서버 설계

Controller/ Service

```java
@Controller
@RequiredArgsConstructor
public class CompanyConetroller {

    @Autowired
    private CompanyService companyService;

    @GetMapping("/company/updateForm")
    @CompanyCheck
    public String updateForm(Model model) {
        UserVo principal = redisService.getValue("principal");
        CompanyUpdateRespDto companyUpdateRespDto = companyService
        model.addAttribute("companyDto", companyUpdateRespDto);
        redisServiceSet.addModel(model);
        return "company/updateForm";
    }
}
```

```java
@Service
@RequiredArgsConstructor
public class CompanyService {
    private final CompanyRepository companyRepository;
    private final UserRepository userRepository;

    @Transactional(readOnly = true)
    public CompanyUpdateRespDto getCompanyUpdateRespDto(Integer principalId) {
        User user = userRepository.findById(principalId);
        Company company = companyRepository.findByUserId(principalId);
        // CompanyUpdateRespDto companyUpdateRespDto = new
        // CompanyUpdateRespDto(user.getPassword(), user.getEmail(),
        // user.getAddress(),
        // user.getDetailAddress(), user.getTel(), company.getCompanyName(),
        // company.getCompanyScale(),
        // company.getCompanyField());
        CompanyUpdateRespDto companyUpdateRespDto = new CompanyUpdateRespDto(user, company
        return companyUpdateRespDto;
    }
}
```

07. 서버 설계

Dto / Exception

```java
@Getter
@Setter
@AllArgsConstructor
public class ResponseDto<T> {
    private Integer code;
    private String msg;
    private T data;
}
```

```java
@Getter
public class CustomException extends RuntimeException {
    private HttpStatus status;
    private String location;

    public CustomException(String msg, HttpStatus status, String location) {
        super(msg);
        this.status = status;
        this.location = location;
    }

    public CustomException(String msg) {
        this(msg, HttpStatus.BAD_REQUEST);
    }

    public CustomException(String msg, HttpStatus status) {
        this(msg, status, null);
    }
}
```

```java
@RestControllerAdvice
public class CustomExceptionHandler {
    @ExceptionHandler(CustomException.class)
    public ResponseEntity<?> customException(CustomException e) {
        if (e.getLocation() == null) {
            return new ResponseEntity<>(Script.back(e.getMessage()), e.getStatus());
        }
        return new ResponseEntity<>(Script.herf(e.getMessage(), e.getLocation())
                                            , e.getStatus());
    }

    @ExceptionHandler(CustomApiException.class)
    public ResponseEntity<?> customApiException(CustomApiException e) {
        return new ResponseEntity<>(new ResponseDto<>(-1, e.getMessage(), null)
                                                , e.getStatus());
    }
}
```

08. 특이 기술

AOP

```
implementation 'org.springframework.boot:spring-boot-starter-aop'
```

```
@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
@Documented
public @interface CompanyCheck {

}
```

```
@Aspect
@Component
public class AopHandler {
    @Autowired
    private HttpSession session;

    @Pointcut("@annotation(shop.mtcoding.jobara.common.aop.CompanyCheck)"
    public void CompanyCheck() {
    }

    @Before("CompanyCheck()")
    public void CompanyCheck(JoinPoint joinPoint) {
        UserVo principal = (UserVo) session.getAttribute("principal");
        Verify.validateObject(principal, "로그인이 필요한 기능입니다"
                , HttpStatus.UNAUTHORIZED, "/#login");
        Verify.checkRole(principal, "company");
    }
```

08. 특이 기술

Redis

```
implementation 'org.springframework.boot:spring-boot-starter-data-redis'
```

```
@Service
public class RedisService {

    @Autowired
    private RedisTemplate<String, String> redisTemplate;

    @Autowired
    private ObjectMapper om;

    public void setValue(String key, Object value) {
        String principalJson = null;
        try {
            principalJson = om.writeValueAsString(value);
        } catch (Exception e) {
            System.out.println("파싱 오류");
        }
        redisTemplate.opsForValue().set(key, principalJson);
    }
```

```
    public UserVo getValue(String key) {
        UserVo principal = null;
        try {
            String principalJson = redisTemplate.opsForValue().get("principal");
            principal = om.readValue(principalJson, UserVo.class);
        } catch (Exception e) {
            System.out.println("파싱 실패");
        }

        return principal;
    }

    public void logout(String sessionId) {
        redisTemplate.delete(sessionId);
        UserVo UserVo = new UserVo();
        try {
            String principalJson = om.writeValueAsString(UserVo);
            setValue("principal", principalJson);
        } catch (Exception e) {
            System.out.println("파싱 오류");
        }
```

```
<c:set var="principal" value="${redisService.getValue('principal')}" />
```

08. 특이 기술

AOP + Redis

```
@Aspect
@Component
public class AopHandler {
    @Autowired
    private RedisService redisService;

    @Pointcut("@annotation(shop.mtcoding.jobara.common.aop.CompanyCheck)")
    public void CompanyCheck() {
    }

    @Before("CompanyCheck()")
    public void CompanyCheck(JoinPoint joinPoint) {
        UserVo principal = redisService.getValue("principal");
        Verify.validateObject(principal, "로그인이 필요한 기능입니다"
                        , HttpStatus.UNAUTHORIZED, "/#login");
        Verify.checkRole(principal, "company");
    }
```

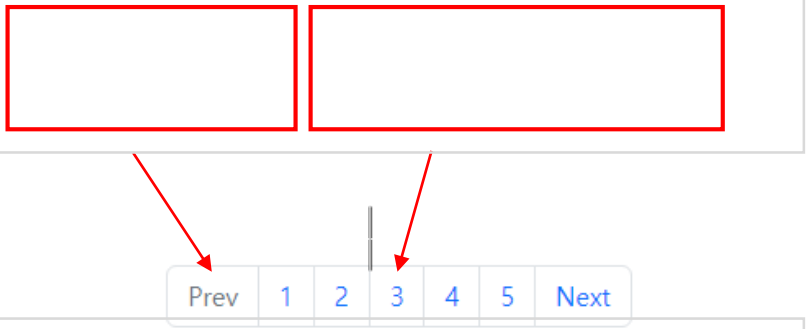
08. 특이 기술

페이징

```
<li class='page-item ${pagingDto.first ? "disabled" : ""}'><a class="page-link"
        onclick="callPrev();">Prev</a></li>

<c:forEach var="num" begin="${pagingDto.startPageNum}" end="${pagingDto.lastPageNum}">
    <li class='page-item'><a class='page-link' href="/?page=${num-1}">${num}</a></li>
</c:forEach>

<li class='page-item ${pagingDto.last ? "disabled" : ""}'><a class="page-link"
        onclick="callNext();">Next</a></li>
```

```
function callPrev() {
    let keyword = `${pagingDto.keyword}`;
    let currentPage = `${pagingDto.currentPage - 1}`
    if (keyword) {
        location.href = "/?page=" + currentPage + "&keyword=" + keyword;
    } else {
        location.href = "/?page=" + currentPage;
    }
}

function callNext() {
    let keyword = `${pagingDto.keyword}`;
    let currentPage = `${pagingDto.currentPage + 1}`
    if (keyword) {
        location.href = "/?page=" + currentPage + "&keyword=" + keyword;
    } else {
        location.href = "/?page=" + currentPage;
    }
}
```

View에 넘겨줘야할 값 4개 (2종류)



View에 넘겨줘야할 오브젝트 (Model)

08. 특이 기술

View – **Controller&Service** – List 계산 – 페이징 계산

Controller

```java
@GetMapping({ "/", "/boards" })
public String getBoardList(Model model, Integer page, String keyword) {
    PagingDto pagingDto = boardService.게시글목록보기(page, keyword);
    model.addAttribute(attributeName: "pagingDto", pagingDto);
    return "board/main";
}
```

Service

```java
@Service
public class BoardService {

    private final BoardRepository boardRepository;

    public PagingDto 게시글목록보기(Integer page, String keyword) {

        if (page == null) {
            page = 0;
        }
        int startNum = page * PagingDto.ROW;

        List<MainDto> boardsList = boardRepository.findAll(startNum, keyword, PagingDto.ROW);
        PagingDto pagingDto = boardRepository.paging(page, keyword, PagingDto.ROW);

        pagingDto.makeBlockInfo(keyword);
        pagingDto.setMainDtos(boardsList);

        return pagingDto;
    }
}
```

```java
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
public static class PagingDto {
    public static final int ROW = 5;
    private boolean isNotResult;
    private String keyword;
    private Integer blockCount;
    private Integer currentBlock;
    private Integer currentPage;
    private Integer startPageNum;
```

```
<c:forEach var="num" begin="${pagingDto.startPageNum}" end="${pagingDto.lastPageNum}">
    <li class='page-item'><a class='page-link' href="/?page=${num-1}">${num}</a></li>
</c:forEach>
```

```
function callPrev() {
    let keyword = `${pagingDto.keyword}`;
    let currentPage = `${pagingDto.currentPage - 1}`
    if (keyword) {
        location.href = "/?page=" + currentPage + "&keyword=" + keyword;
    } else {
        location.href = "/?page=" + currentPage;
    }
}
```

```java
    this.totalPage = (int) Math.ceil((double) totalCount / ROW);

    if (this.currentPage == totalPage - 1) {
        this.isLast = true;
    } else {
        this.isLast = false;
    }
    if (totalPage < lastPageNum) {
        this.lastPageNum = totalPage;
    }
}
}
```

08. 특이 기술

Controller

```
@GetMapping({ "/", "/boards" })
public String getBoardList(Model model, Integer page, String keyword) {
    PagingDto pagingDto = boardService.게시글목록보기(page, keyword);
    model.addAttribute(attributeName: "pagingDto", pagingDto);
    return "board/main";
}
```

Service

```
@Service
public class BoardService {

    private final BoardRepository boardRepository;

    public PagingDto 게시글목록보기(Integer page, String keyword) {

        if (page == null) {
            page = 0;
        }
        int startNum = page * PagingDto.ROW;

        List<MainDto> boardsList = boardRepository.findAll(startNum, keyword, PagingDto.ROW);
        PagingDto pagingDto = boardRepository.paging(page, keyword, PagingDto.ROW);

        pagingDto.makeBlockInfo(keyword);
        pagingDto.setMainDtos(boardsList);

        return pagingDto;
    }
}
```

currentPage 0 : OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY

currentPage 1 : OFFSET 5 ROWS FETCH NEXT 5 ROWS ONLY

currentPage 2 : OFFSET 10 ROWS FETCH NEXT 5 ROWS ONLY

Query

```
<select id="findAll" re
    SELECT
    b.id, b.title, u.us
    FROM
    board_tb b
    LEFT OUTER JOIN
    user_tb u
    ON
    b.user_id = u.id
    <if test="keyword !
        WHERE title lik
    </if>
    ORDER BY b.id DESC
    OFFSET #{startNum}
    FETCH NEXT #{row} R
</select>
```

ID	TITLE	CONTENT	USER_ID	CREATED_AT
1	1번째 제목	내용	1	2023-03-08 16:58:29.742581
2	2번째 제목	내용	1	2023-03-08 16:58:29.743581
3	3번째 제목	내용	1	2023-03-08 16:58:29.744583
4	4번째 제목	내용	1	2023-03-08 16:58:29.744583
5	5번째 제목	내용	1	2023-03-08 16:58:29.745583
6	6번째 제목	내용	1	2023-03-08 16:58:29.745583
7	7번째 제목	내용	1	2023-03-08 16:58:29.746583
8	8번째 제목	내용	1	2023-03-08 16:58:29.747583
9	9번째 제목	내용	1	2023-03-08 16:58:29.747583
10	10번째 제목	내용	1	2023-03-08 16:58:29.748582
11	11번째 제목	내용	1	2023-03-08 16:58:29.749588
12	12번째 제목	내용	1	2023-03-08 16:58:29.750586
13	13번째 제목	내용	1	2023-03-08 16:58:29.752586
14	14번째 제목	내용	1	2023-03-08 16:58:29.752586
15	15번째 제목	내용	1	2023-03-08 16:58:29.753582
16	16번째 제목	내용	1	2023-03-08 16:58:29.753582

View – Controller&Service – List 계산 – **페이징 계산**

```java
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
public static class PagingDto {
    public static final int ROW = 5;
    private boolean isNotResult;
    private String keyword;
    private Integer blockCount;
    private Integer currentBlock;
    private Integer currentPage;
    private Integer startPageNum;
    private Integer lastPageNum;
    private Integer totalCount;
    private Integer totalPage;
    private boolean isLast;
    private boolean isFirst;
    private List<MainDto> mainDtos;
    public void makeBlockInfo(String keyword) {
        this.keyword = keyword;

        this.isLast = true;
    } else {
        this.isLast = false;
    }
    if (totalPage < lastPageNum) {
        this.lastPageNum = totalPage;
    }
}
}
```

nt / ROW);

View에 넘겨줘야할 값 4개 (2종류)

~~${pagingDto.first}~~ ~~${pagingDto.startPageNum}~~

~~${pagingDto.last}~~ ~~${pagingDto.lastPageNum}~~

```xml
<select id="paging" resultType="shop.mtcoding.paging.dto.board.BoardResp$PagingDto">
    SELECT count(*) total_count, #{page} current_page,
    case when #{page} = 0 then 1
         else 0
         end first
    FROM board_tb
    <if test="keyword != null">
        WHERE title like CONCAT('%',#{keyword},'%')
    </if>
</select>
```

```xml
<li class='page-item ${pagingDto.first ? "disabled" : ""}'><a class="page-link"
        onclick="callPrev();">Prev</a></li>

<c:forEach var="num" begin="${pagingDto.startPageNum}" end="${pagingDto.lastPage~
    <li class='page-item'><a class='page-link' href="/?page=${num-1}">${num}</a>
</c:forEach>

<li class='page-item ${pagingDto.last ? "disabled" : ""}'><a cla
        onclick="callNext();">Next</a></li>
```

currentPage(요청) | blockCount * currentBlock | startPageNum | lastPageNum

1, 2, 3, 4, 5 | 5 * 0 = 0 | 1 | 5

6, 7, 8, 9, 10 | 5 * 1 = 5 | 6 | 10

11, 12, 13, 14, 15 | 5 * 2 = 10 | 11 | 15

Prev 1 2 3 4 5 Next

Prev 6 7 8 9 10 Next

Prev 11 12 13 14 15 Next

startPageNum, lastPageNum은 항상 1,5 / 6,10 / 11,15 ... 구조

만약, totalPage : 8인 상태에서 요청페이지가 6, 7, 8 이라면

왼쪽 코드 조건문 → if (8 < 10), lastPageNum 저장 값 10 → 8

Prev 6 7 8 ~~9~~ ~~10~~ Next → Prev 6 7 8 Next

09. 테스트 코드

employee / company

```
@AutoConfigureMockMvc
@SpringBootTest(webEnvironment = WebEnvironment.MOCK)
public class EmployeeControllerTest {

	@Autowired
	private MockMvc mvc;

	@Mock
	private EmployeeUpdateReqDto employeeUpdateReqDto;

	@Autowired
	private RedisService redisService;

	private MockHttpSession mockSession;

	@BeforeEach
	public void setUp() {
		UserVo principal = new UserVo();
		principal.setId(1);
		principal.setUsername("ssar");
		principal.setRole("employee");
		principal.setProfile(null);
		redisService.setValue("principal", principal);
		mockSession = new MockHttpSession();
		mockSession.setAttribute("principal", principal);
	}
```

```
@Test
public void join_test() throws Exception {
	// given
	String requestBody = "username=asdf&password=1234&email=asdf@nate.com";
	// when
	ResultActions resultActions = mvc.perform(post("/employee/join")
			.content(requestBody)
			.contentType(MediaType.APPLICATION_FORM_URLENCODED_VALUE));

	// then
	resultActions.andExpect(status().is3xxRedirection());
}
```

09. 테스트 코드

employee / company

```
	@Test
	public void update_test() throws Exception {
		// given
		int id = 1;
		MockMultipartFile file = new MockMultipartFile(
				"profile",
				"filename.txt",
				"image/jpeg",
				"Test data".getBytes()
		);

		// when
		ResultActions resultActions = mvc.perform(multipart("/employee/update/" + id)
				.file(file) // 파일을 첨부합니다.
				.param("password", "1234")
				.param("email", "ssar@nate.com")
				.param("address", "부산시")
				.param("detailAddress", "12구")
				.param("tel", "01099876554")
				.param("career", "2")
				.param("education", "고졸")
				.session(mockSession));

		// then
		resultActions.andExpect(status().is3xxRedirection());
	}
}
```

09. 테스트 코드

board

```
@Transactional
@AutoConfigureMockMvc
@SpringBootTest(webEnvironment = WebEnvironment.MOCK)
public class BoardControllerTest {
    @Autowired
    private MockMvc mvc;

    @Autowired
    private ObjectMapper om;

    private MockHttpSession mockSession;
    private MockHttpSession mockSession2;

    @Autowired
    private RedisService redisService;

    @BeforeEach
    public void setUp() {
        UserVo principal = new UserVo();
        principal.setId(6);
        principal.setUsername("cos");
        principal.setRole("company");
        principal.setProfile(null);

        redisService.setValue("principal", principal);

        mockSession = new MockHttpSession();
        mockSession.setAttribute("principal", principal);
    }
```

09. 테스트 코드

board

```java
@Test
public void myBoardList_test() throws Exception {
    // given
    // int id = 7; 여기서 id:userId, 열람권한 없음 체크
    int id = 6;

    // when
    ResultActions resultActions = mvc.perform(
            get("/board/boardList/" + id).session(mockSession));

    Map<String, Object> map = resultActions.andReturn().getModelAndView().getModel();
    List<MyBoardListRespDto> boardListDto =
                        (List<MyBoardListRespDto>) map.get("myBoardList");
    // String model = om.writeValueAsString(boardListDto);
    // System.out.println("테스트 : " + model);

    // then
    resultActions.andExpect(status().is2xxSuccessful());
}
```

```java
@Test
public void boardList_test() throws Exception {
    // given
    String keyword = null;
    Integer page = 0;

    // when
    ResultActions resultActions = mvc.perform(
            get("/board/list?page=" + page + "&keyword=" + keyword)
                    .session(mockSession));

    Map<String, Object> map = resultActions.andReturn().getModelAndView().getModel();
    PagingDto boardList = (PagingDto) map.get("pagingDto");

    String model = om.writeValueAsString(boardList.getBoardListDtos());
    System.out.println("테스트 : " + model);

    // then
    resultActions.andExpect(status().isOk());
    assertThat(boardList.getBoardListDtos().size()).isEqualTo(8);
    // assertThat(boardList.getBoardListDtos().get(0).getCss()).isEqualTo("fa-solid");
}
```

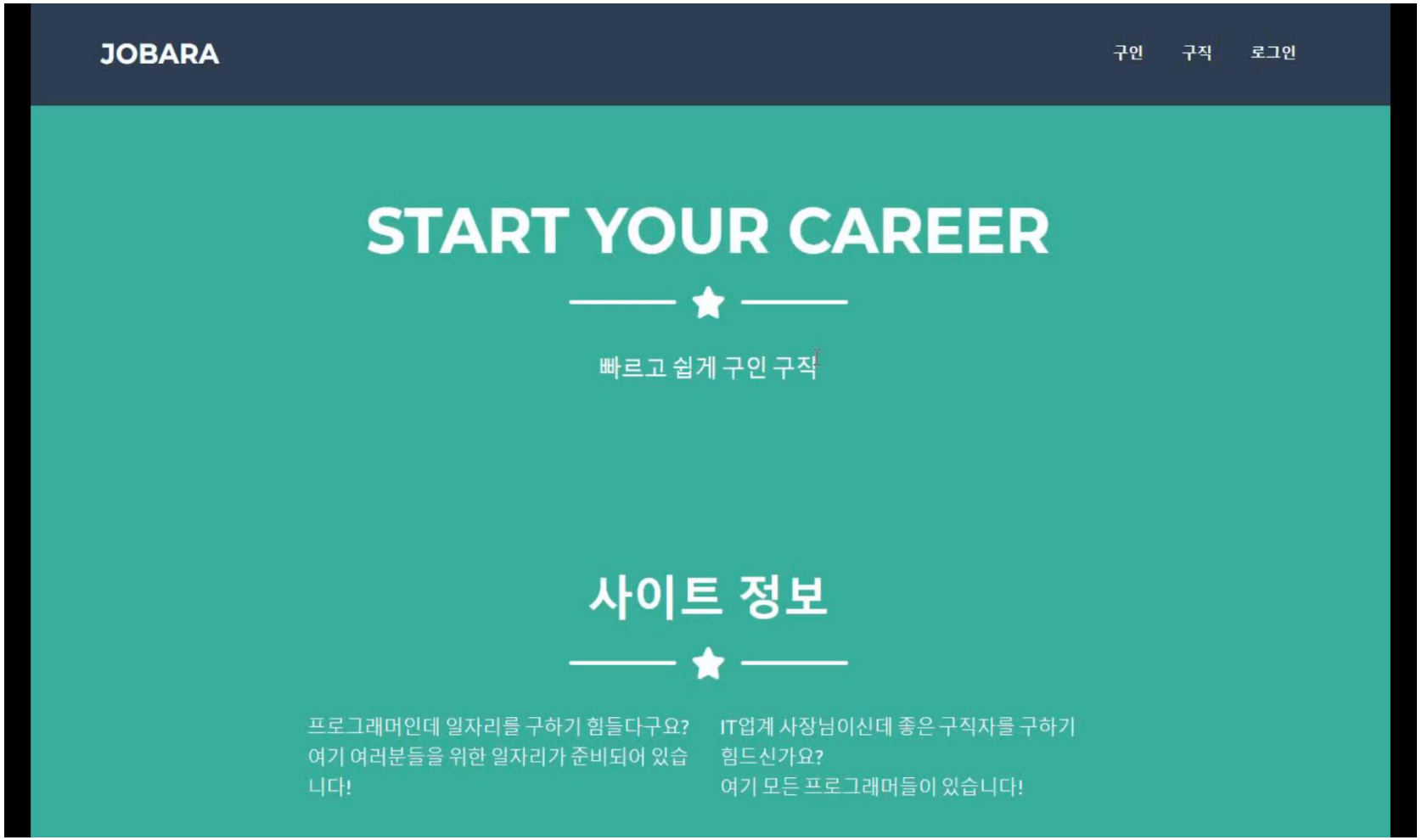
09. 테스트 코드

Repository

```
@MybatisTest
public class BoardRepositoryTest {

    @Autowired
    private BoardRepository boardRepository;
    @Autowired
    private BoardTechRepository boardTechRepository;
```

```
@Test
public void findAllByIdWithCompany_test() throws Exception
    // given
    ObjectMapper om = new ObjectMapper();
    int userId = 6;

    // when
    List<MyBoardListRespDto> myBoardListRespDto =
            boardRepository.findAllByIdWithCompany(userId);

    // then
    assertThat(myBoardListRespDto.get(1).getTitle())
                .isEqualTo("인공지능 솔루션 (AI Solution) 개발");
}
```

10. 기능 동영상 시연


JOBARA
구인
구직
로그인
START YOUR CAREER
빠르고 쉽게 구인 구직
사이트 정보
프로그래머인데 일자리를 구하기 힘들다구요?
여기 여러분들을 위한 일자리가 준비되어 있습니다!
IT업계 사장님이신데 좋은 구직자를 구하기 힘드신가요?
여기 모든 프로그래머들이 있습니다!

11. 후기

Q&A

JOBARA

Thank you