forBIT

forBIT, 비트캠프를 위한 커뮤니티

4BIT TEAM

류혜영, 이중호, 이채연, 조성헌, 주영빈, 홍다경, 황서영

목차

forBIT

기획의도

forBIT

Why?

Project_지각왕

자바하다 죽을지 몰라 Ver7 럭키 7조

```
Console
Client [Java Application] C:₩Program Files₩Java₩jre1.8.0_201₩bin₩javaw.exe (2019. 9. 6. 오전 6:29:57)
==========QUIZ==========
1.일반상식
2.문화
3.철학
4.컴퓨터
5.종료
=========================
번호를 입력하세요
4
***컴퓨터 퀴즈를 선택 하셨습니다.***
1번: 프로그램(코드)를 기계가 이해할 수 있는 언어로 바꾸는 작업을 무엇인가?
컴파일
정답입니다~♬
2번: 클래스를 호출할 때 제일 먼저 호출되며 멤버 변수를 초기화하는 것은?
생성자
정답입니다~♬
3번: 일반 함수에 객체 지향 개념을 추가하여, 클래스 내부에 선언하고 클래스 멤버변수를 사용하여 클래스 기능을 구현하는 것은?
```

```java
    public void run() { // run에서 통신을 수행
        try {
            printMenu();
            sel = (br.read() - 48); // Ascii Code 라서 문자열로 1은 49이기 때문에 48을 빼줌
            PlayThread pt = new PlayThread(sock);
            pt.selMenu(sel);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }// run() method
}// PlayThread class

public class quiztest {
    public static void main(String[] args) {
        try {
            ServerSocket server = new ServerSocket(10001);
            System.out.println("~ 접속을 기다립니다 ~");
            while (true) {
                // 1. 클라이언트가 접속
                Socket sock = server.accept();
                // 2. 클라이언트 담당 쓰레드 생성하여 전담시킴
                PlayThread playthread = new PlayThread(sock);
                new Thread(playthread).start();
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

forBIT

Why?

비전공자는 어떻게 공부하지?

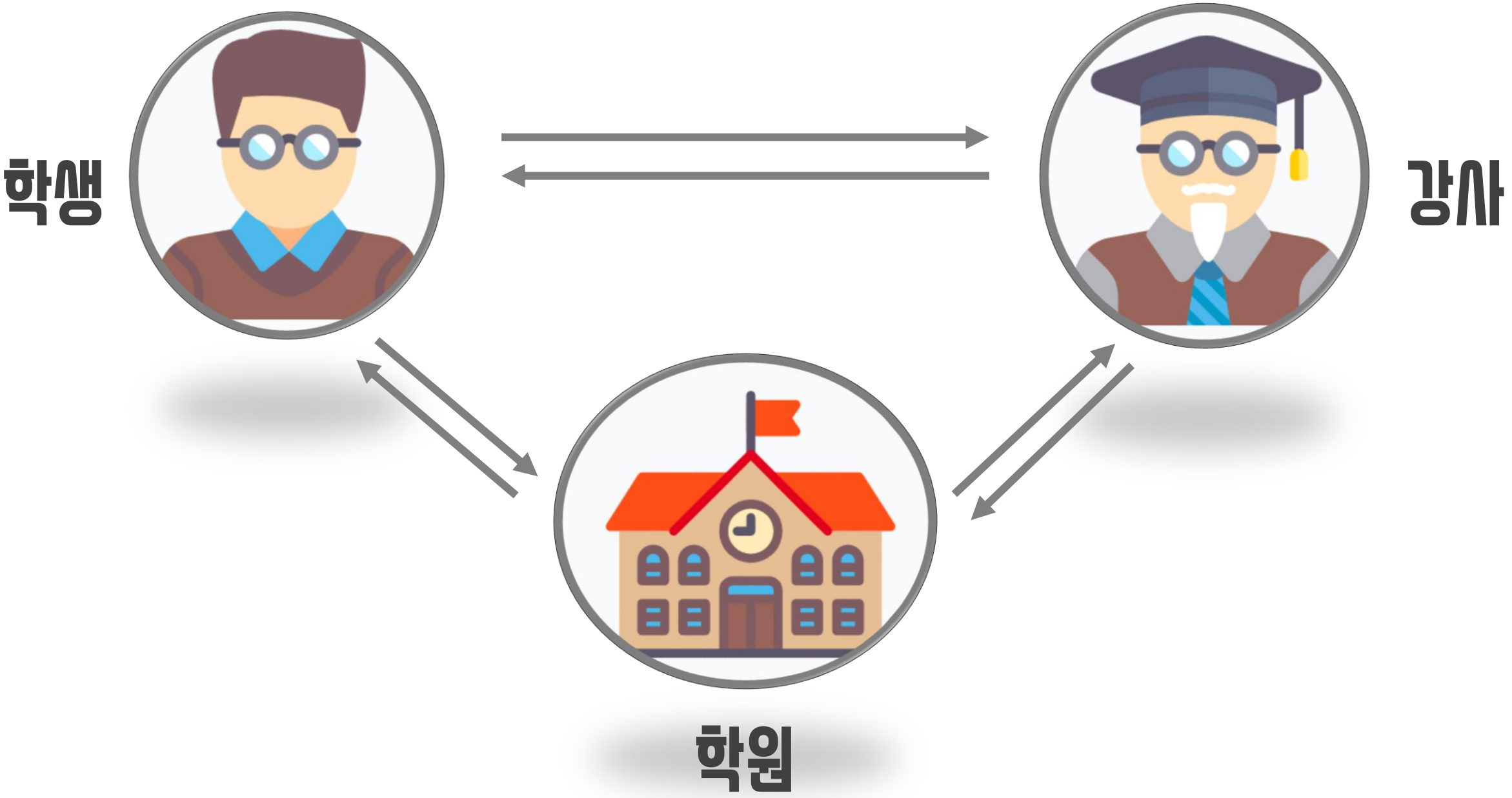
새로운 분야에 대한 도전

낯선 언어

넓은 사막에서 길을 잃다

선생님은 비행기, 나는 뚜벅이

Why?




forBIT
진행과정

Steps

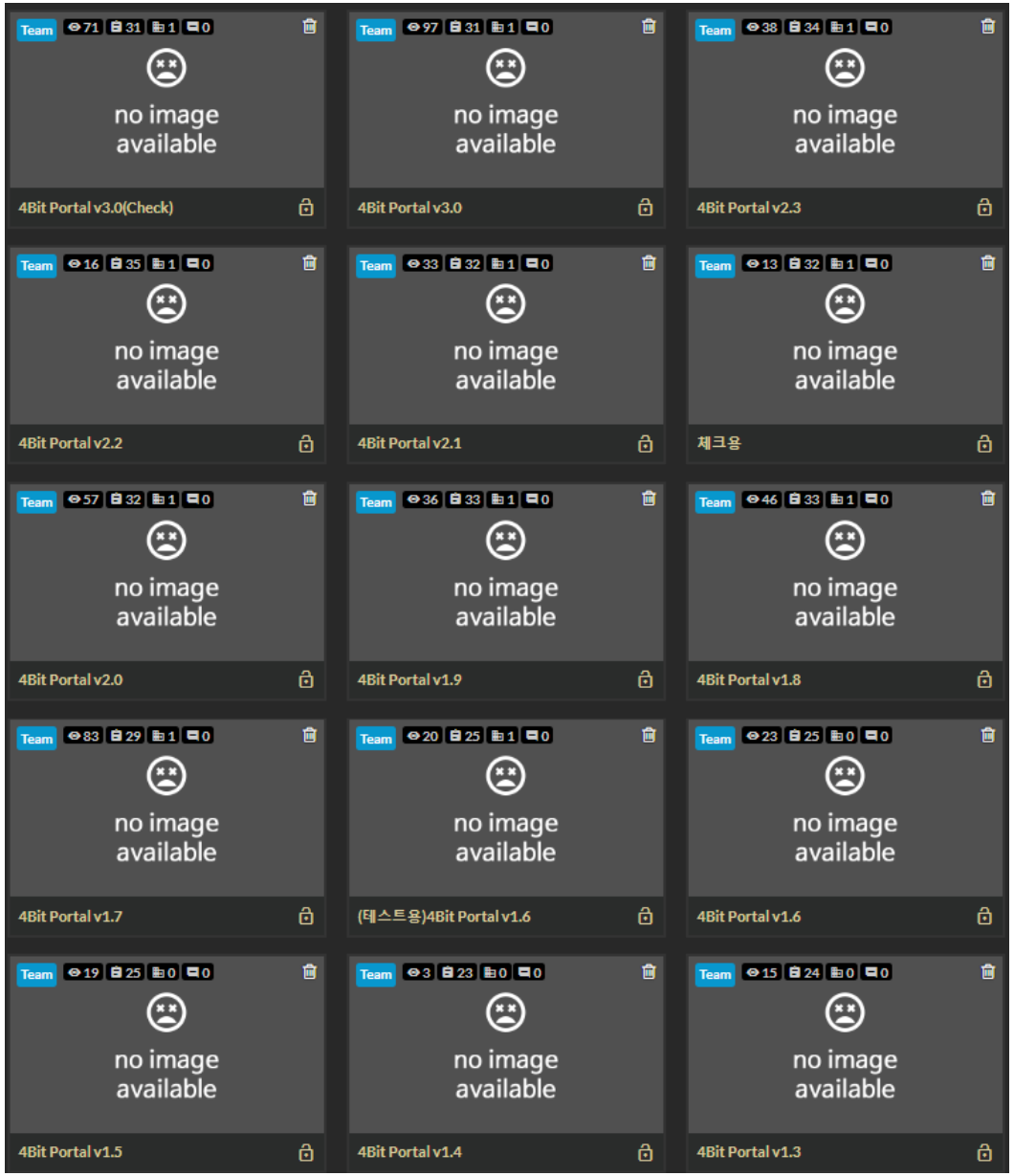
주간	주간 목표	비고	월	화	수	목	금	토	일
7월 5주차	**개발 start!** 더미 코드 작성 DB 세팅		end point 마무리 DB 세팅 더미 코드 작성 더미 DB 쿼리문 작성 소스트리 코드 관리 연습	entity 작성 controller 초안 작성 더미 쿼리문 작성 git 연습	Entity 작성 DB 로그인 성공	게시판 입력 Component DB 입력 성공	React 공부내용 공유	개인 공부	- 댓글 입력 Component DB 입력 성공 - 게시판 리스트 Component 완성
8월 1주차	소스 코드 작성 Entity, Repository, Controller 작성 Component 작성	DB 최종 수정 Entity 검증	8/5 DB 수정 entity,repository 작성	8/6 entity,repository UnitTest (~목까지 완료 예정)	8/7 선생님 DB 스키마 점검 DB 수정 권한 정리 => 변경DB로 entity 다시 작성 (~금까지 test 진행)	8/8 entity 유닛테스트 마무리	8/9 각 파트 코드 작성 중 중호 : article 영빈 : homework 혜영,채연 : test 다경 : attend 성현 : roadmap 서영 : admin user	8/10 각 파트 코드 작성 중 // 다경 프론트엔드 route 작성중	8/11 article CRUD 구현 성공
8월 2주차	백엔드 코드 작성 Controller 기본 기능 작성 완료 프론트엔드 코드 작성 기본 set 작성	15,16 학원 휴가 >> 프로젝트 정상 진행 예정	8/12 - 각 담당파트 CRUD 계속 이어서 작성 - 프론트엔드 path 작성	8/13 프론트엔드 path별 리스트 확인	8/14 담당 파트 개발 파트 취합 진행 프론트 베이스 프로젝트 작성 진행	8/15 - DB 수정된 부분 조율 >> 쿼리문 수정 - 각 파트 취합, 프로젝트 병합	8/16 백엔드 코드 마무리	8/17 *휴식	8/18 - 백엔드 개발 계속 진행 - 프론트엔드 코드 git 세팅 - 프론트 코드 오리엔테이션
8월 3주차	백엔드 코드/ 프론트 엔드 코드 연결 확인	프론트 컴포넌트 작성 프론트에 맞춰서 백엔드 계속 수정중	8/19 각 페이지 완성(~22일까지) 로그인, 게시판, 회원등록 화면 구현 완료	8/20 백엔드 코드 마무리 프론트 페이지 작성 문제 생성 화면 구현	8/21 프론트 코드 계속 작성중 리스트 부분 대부분 완성	8/22 프론트 코드 계속 작성중 수정 부분 작성	8/23 프론트 코드 계속 작성중	8/24 - 프론트 코드 계속 작성중 템플릿 코드 분석 - 백엔드 출결관련 컨트롤러 개발중	8/25 ~~템플릿 마무리~~ 템플릿 시작
8월 4주차	프로젝트 마무리 작업 문서 작업 시작	프론트 템플릿 정리 오류, 디테일 수정	8/26 백 예외처리 프론트 템플릿 코드 분석 >>템플릿에 아티클 컴포넌트 붙였음 컴포넌트 계속 작성중	8/27 프론트 템플릿 코드 레파지토리 생성 >> 기존 컴포넌트 이식중	8/28 프론트 템플릿에 기존 컴포넌트 붙이면서 수정중 백엔드 출책 마무리	8/29 ~~템플릿에 컴포넌트 합치기~~ ~~마무리~~	8/30 템플릿 작업 계속 진행중	8/31 맡은 부분 템플릿 작업 마무리 오후 시나리오 정리	9/1 템플릿 코드 취합 오류 수정

Steps

약 10주간의 진행 과정

forBIT

Steps



총 27번의 DB 수정…

이중호 5:19 PM
DB_ver_1.4 입니다

4Bit_DB_ver_1.4.sql

```
# 4BIT Scheme v 1.4
# Written by Hwang
# Updated by Lee
```

Tuesday, July 30th

황서영 10:05 AM
더미 쿼리문 서영다경성현 작성

insert_v1.1.sql

```
-- 서영작성 start
-- user 테이블 학생 5/ 강사 3/ 어드민 2
INSERT INTO `user` VALUE
('2019022601','1234','김OO','kim@bitcamp.com','010-1111-1111','role_student',5,1),
('2019022602','1234','이OO','lee@bitcamp.com','010-2222-2222','role_student',5,1),
```

황서영 3:30 PM
쿼리문 v.1.5

4Bit_DB_ver_1.5.sql

```
# 4BIT Scheme v 1.5
# Written by Hwang
# Updated by Lee
# Updated by Hong,Cho
```

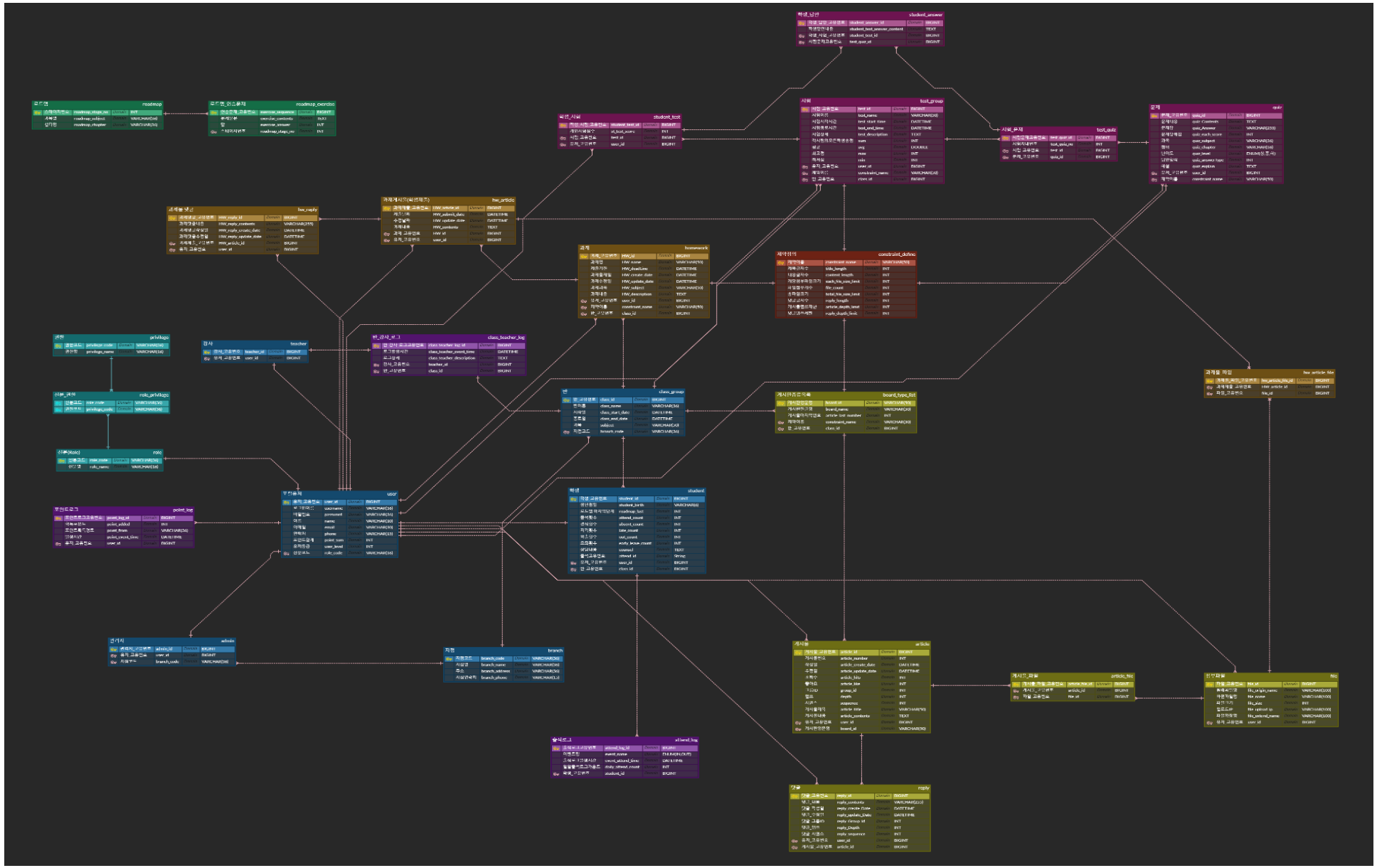
홍다경 3:31 PM
인서트문 1.1

insert_v1.1.sql

```
INSERT INTO `user` (`username` ,`password`,`name`,`email`,`phone`,`role_code`
,`point_sum`,`user_level`)
VALUE ('2019022601','1234','김OO','kim@bitcamp.com','010-1111-1111','role_student',5,1),
('2019022602','1234','이OO','lee@bitcamp.com','010-2222-2222','role_student',5,1),
('2019022603','1234','박OO','park@bitcamp.com','010-3333-3333','role_student',5,1),
('2019022604','1234','조OO','cho@bitcamp.com','010-4444-4444','role_student',5,1),
```

forBIT

Steps



ER-Diagram

forBIT

Steps

공지 - 전체공지

공지 - 전체공지 내용

학생

공지 - 취업공고

공지 - 취업공고 내용

수업-시험출제-DELE...

수업-과제출제-LIST

강사

수업-과제출제-READ

수업-과제출제-UPD...

관리-회원관리

관리-회원관리-삭제...

관리자

관리-회원관리-상세

관리-회원등록

UI 데모 제작

구현화면_강사용0...

비트캠프 학원용 통합포털

2019년 7월 27일

특정유저만 테스트 허용

구현화면_학생_0...

비트캠프 학원용 통합포털

2019년 7월 27일

특정유저만 테스트 허용

구현화면_관리자 ...

비트캠프 학원용 통합포털

2019년 7월 25일

특정유저만 테스트 허용

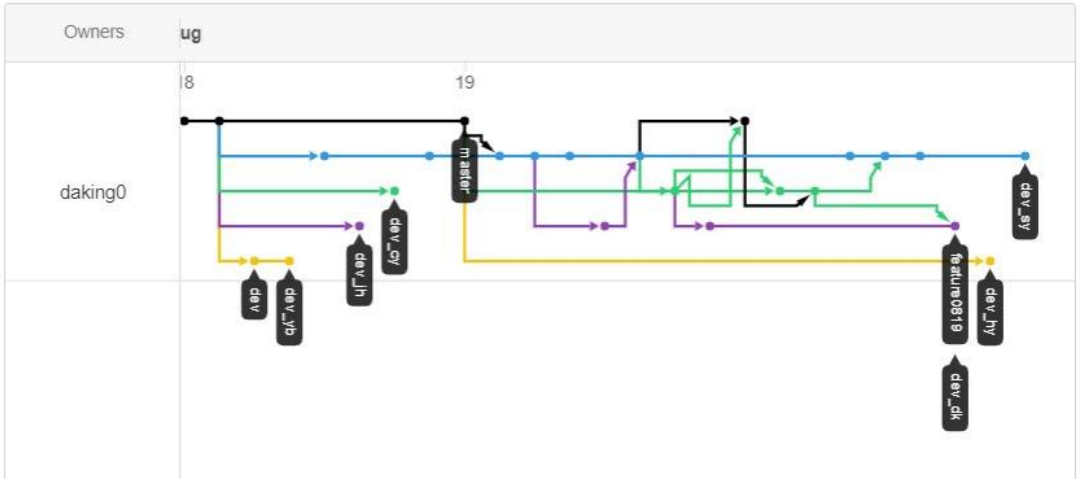
forBIT

Steps

Git Repository 협업

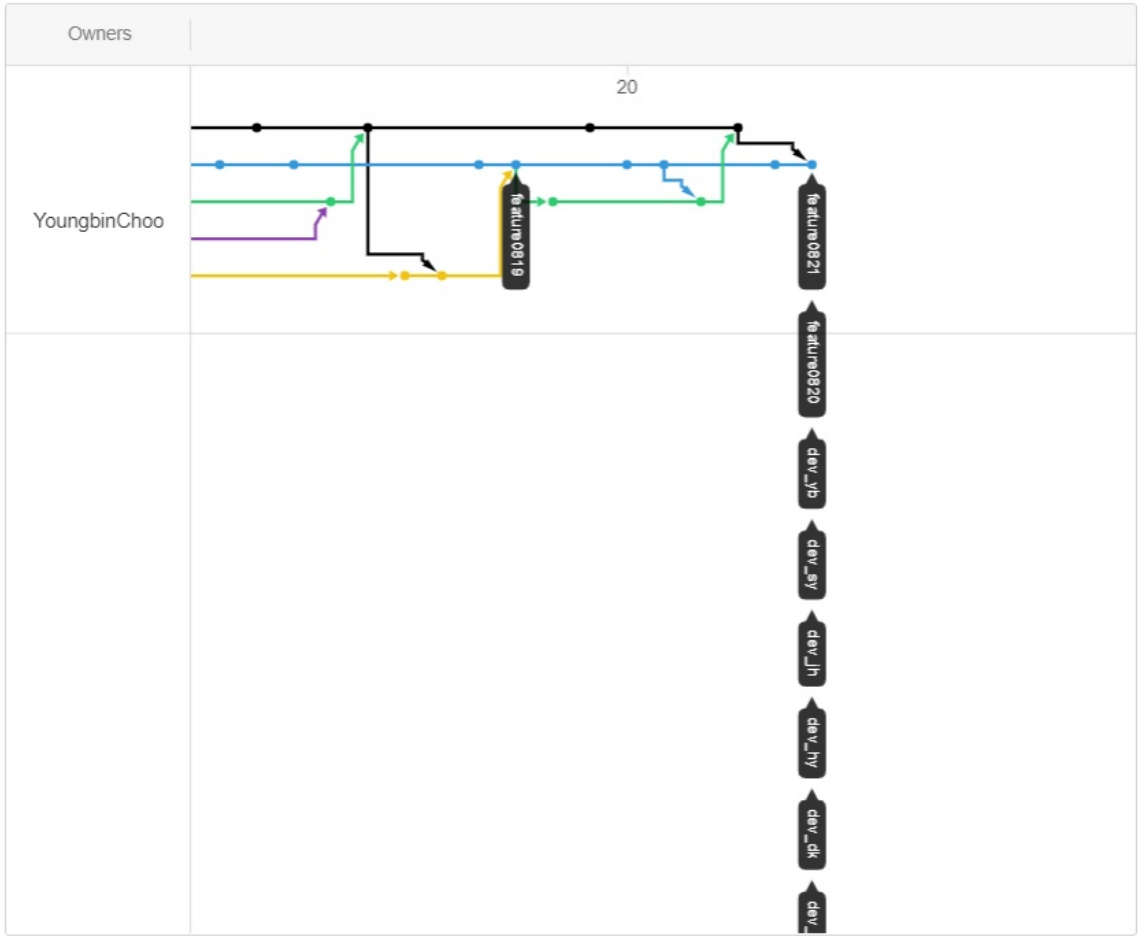
Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Keyboard shortcuts available

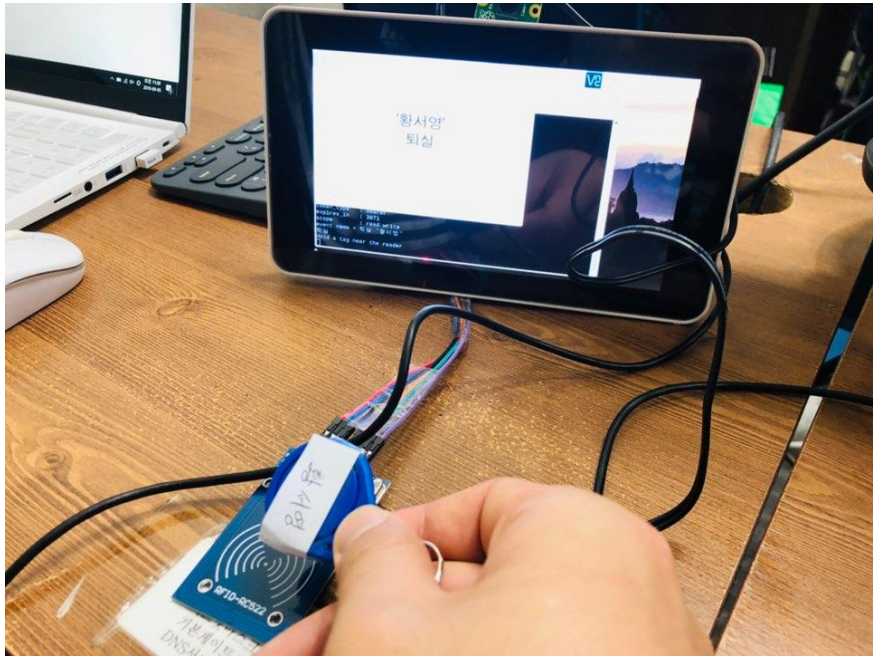
forBIT


forBIT
주요기능

Features

학생 서비스

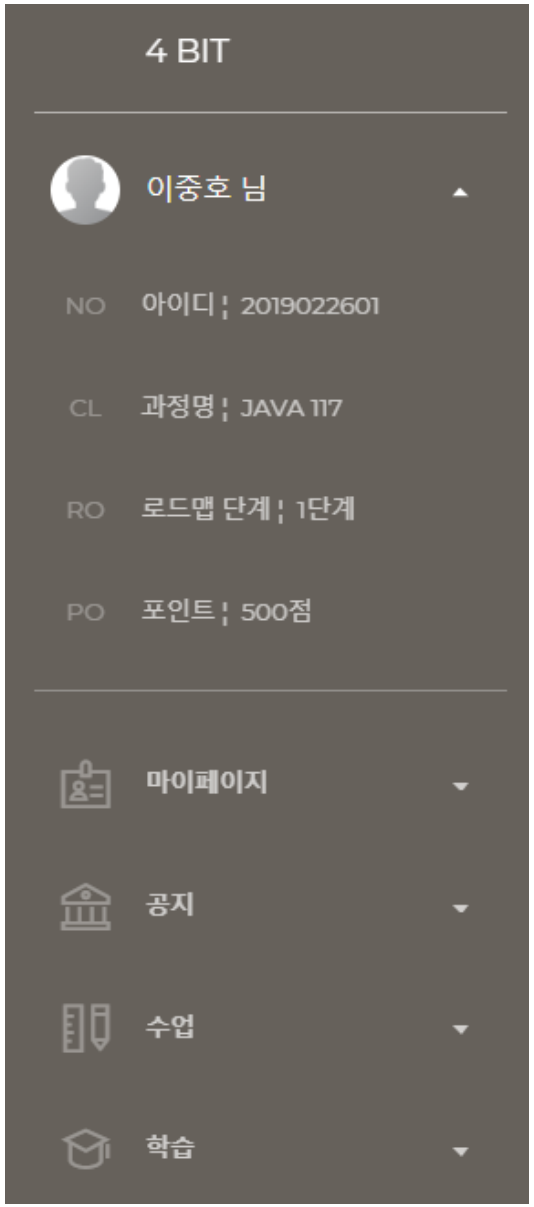
RFID를 이용한 입퇴실 기능



'황서영'
입실

'홍다경'
조퇴

Features

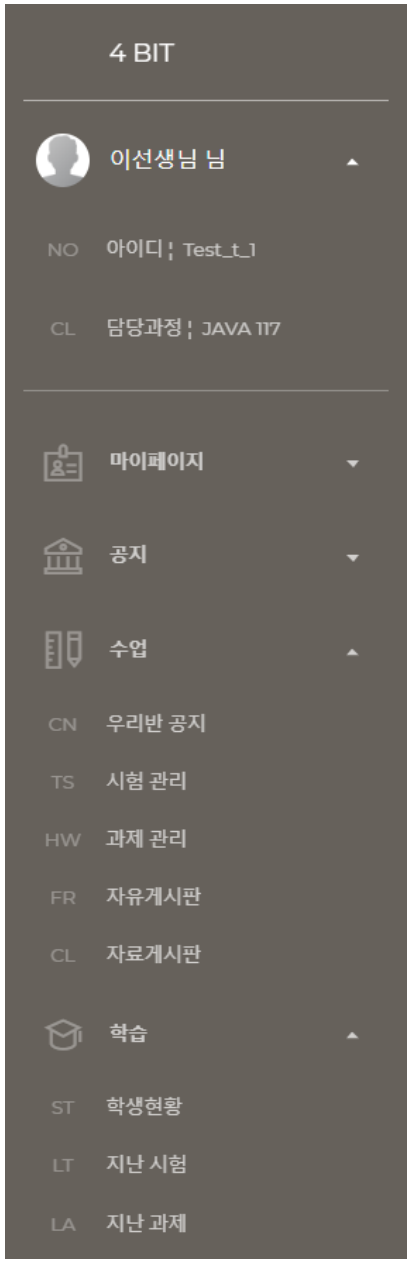


학생 서비스

- 학원 전체 공지, 취업 게시판
- 과정별 공지, 자료, 자유 게시판
- 로드맵 문제풀이
- 포인트 적립

forBIT

Features

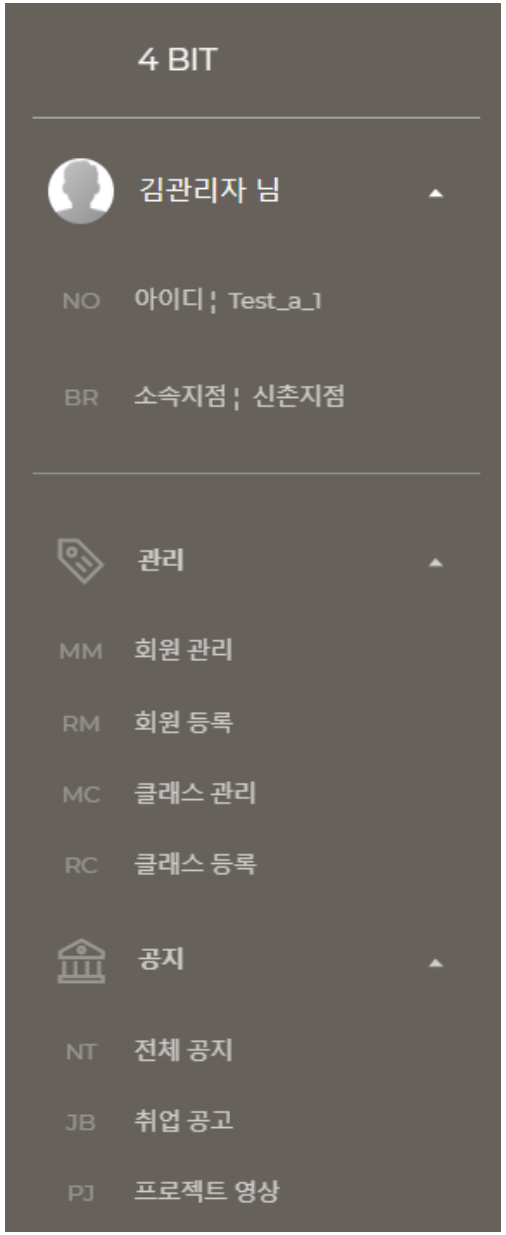


강사 서비스

- 시험 출제, 과제 출제
- 학생현황 관리
- 담당 과정 게시판 관리

forBIT

Features



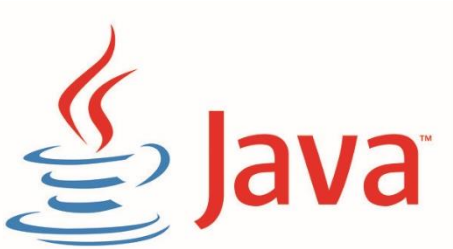
관리자 서비스

- 회원, 클래스 등록 관리
- 학원 전체 게시판 관리

forBIT


forBIT
기술소개

Tech

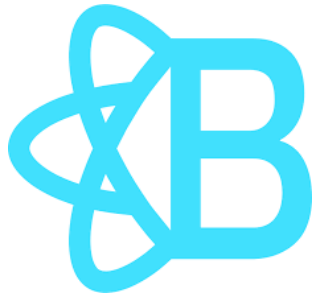

Java


Spring Boot


HIBERNATE


React


Redux




MySQL


IJ


CODE
Visual Studio Code

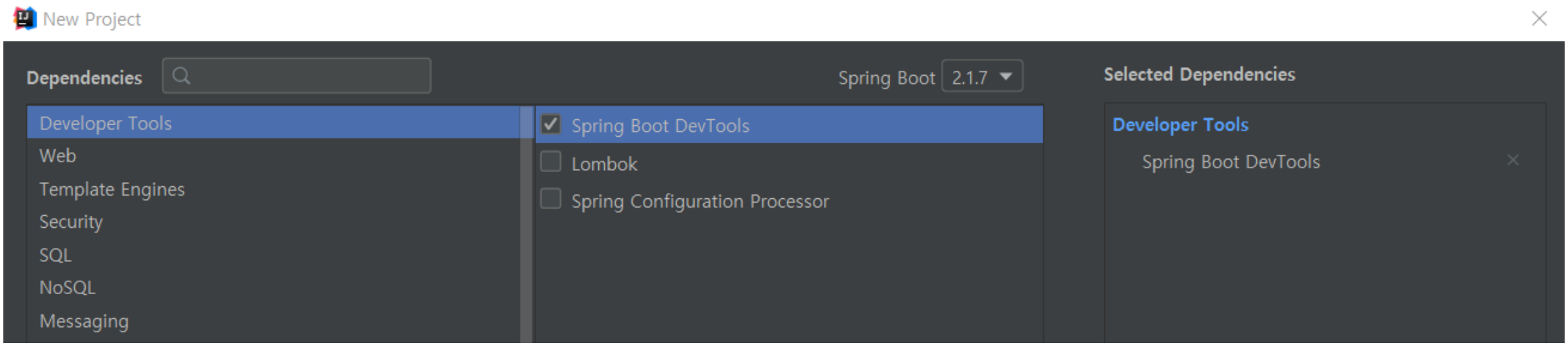

MySQL
Workbench

forBIT

Tech

Back-End 주요 사용 기술

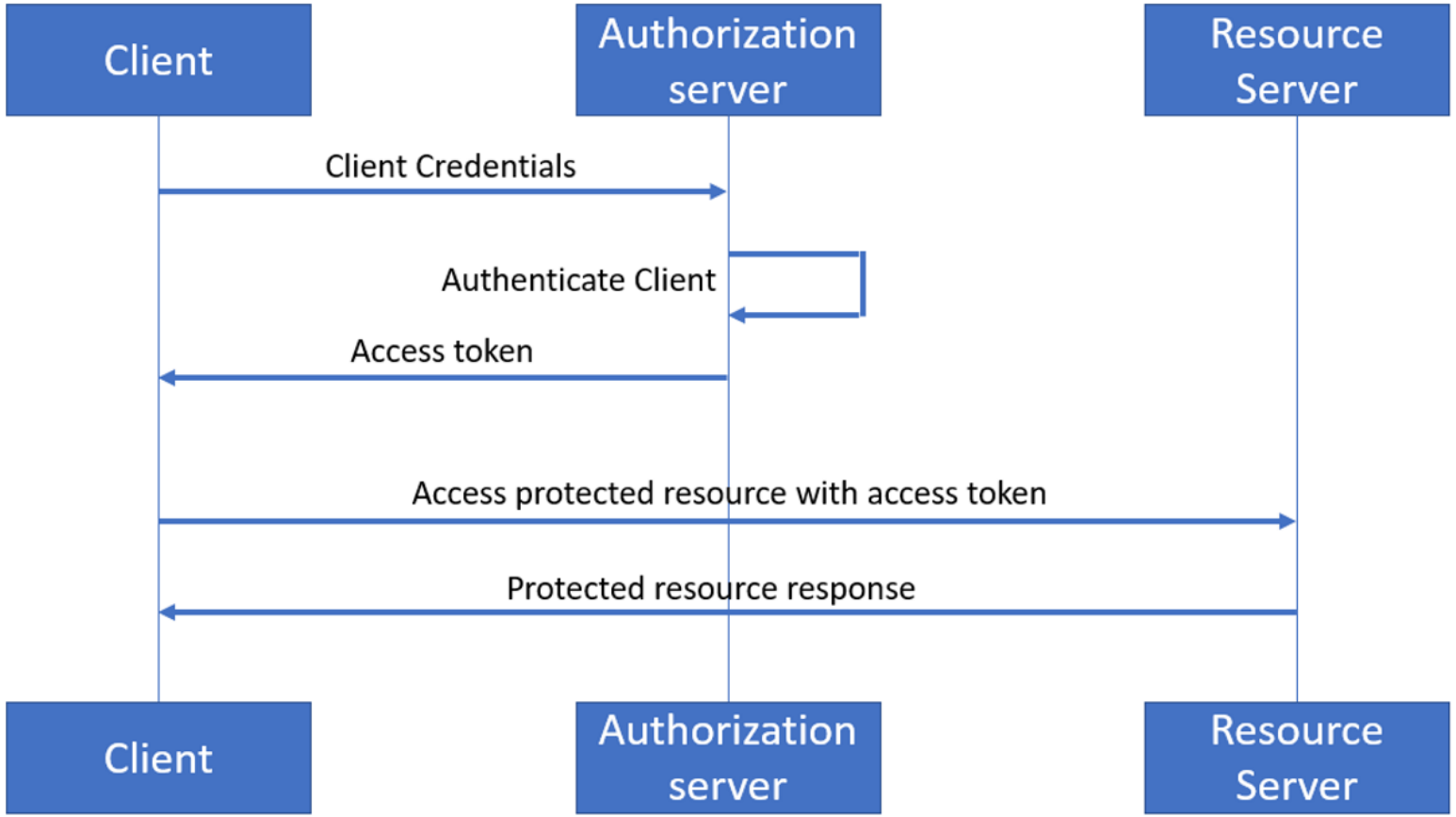
Spring Boot



forBIT

Tech

OAuth 2.0



Tech

application.properties

```
# DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/project_4bit?serverTimezone=Asia/Seoul&useSSL=false
spring.datasource.username=master
spring.datasource.password=Test1234

# JPA (JpaBaseConfiguration, HibernateJpaAutoConfiguration)
spring.jpa.generate-ddl=false
spring.jpa.hibernate.ddl-auto=none
spring.jpa.open-in-view=false
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.show-sql=true
```

forBIT

Tech

@Entity

```java
// article 테이블
@Entity
@Table(name = "article")
@DynamicInsert
public class Article implements Serializable {

    // PK : article_id (게시물_고유번호)
    @Id
    @Column(columnDefinition = "BIGINT", name = "article_id", updatable = false, nullable = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long articleId;

    //article_number (게시물번호)
    @Column(name = "article_number")
    private int articleNumber;

    //article_create_date (작성일)
    @CreationTimestamp
    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "article_create_date")
    private Date articleCreateDate;
```

@Repository

```java
@Repository
public interface ArticleRepository extends JpaRepository<Article, Long> {

    // 해당 Article 테이블에서 where 조건으로 ArticleId를 줘서 찾는것.
    Article findByArticleId(Long articleId);

    // 해당 BoardId에 대한 모든 article을 출력
    Page<Article> findAllByBoardTypeList_BoardId(String boardId, Pageable pageable);

    // 해당 BoardId에 해당하는 article 하나를 조회
    Optional<Article> findByBoardTypeList_BoardIdAndArticleId(String boardId, Long articleId);
```

forBIT

Tech

```java
@RunWith(SpringRunner.class)
@DataJpaTest
@Transactional
@AutoConfigureTestDatabase(replace = AutoConfigureTestDatabase.Replace.NONE)
public class ArticleRepositoryTest {

    @Autowired
    private TestEntityManager entityManager;

    @Autowired
    private ArticleRepository articleRepository;

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private BoardTypeListRepository boardTypeListRepository;


    @Test
    public void testCreateArticle() {
        User user = userRepository.findByUsername("test_s");
        BoardTypeList boardTypeList = boardTypeListRepository.findByBoardId("class_1_board");

        Article article = new Article();
        article.setUser(user);
        article.setArticleContents("Item #1");
        article.setBoardTypeList(boardTypeList);
```

UnitTest

Repository 검증

forBIT

Tech

@Service

```java
@Service
public class ArticleService {

    @Autowired
    private ArticleRepository articleRepository;

    @Autowired
    private ArticleFileRepository articleFileRepository;

    @Autowired
    private FileRepository fileRepository;

    // 역할    :게시글 쓰기 (controller 에서 사용)
    // 설명    : 실제 DB에 접근해서 .save() 를 통해 저장한다. Article 클래스 안의 내용을 article 객체로 반환하여 DB에 저장한다.
    //           .save() 가 insert 문을 생성해준다.
    @Transactional
    public Article createArticle(Article article) { return articleRepository.save(article); }

    // 역할    : BoardId 에 따라서 게시물 전체출력
    // 설명    : findAll 은 Page 인터페이스 안에 구현되있는 구현체이다.
    //           findAll 뒤에 By를 붙여 세부 조건을 걸어준다.
    //           ByOOO : OOO 에 테이블을 넣으면 그 테이블을 where 절에 넣어 쿼리를 만들어준다.
    //           ByOOO_ㅁㅁㅁ : OOO 테이블 안의 ㅁㅁㅁ 컬럼을 조건으로 검색을 한다.
    @Transactional(readOnly = true)
    public Page<Article> listOfArticleByBoardId(String boardId, Pageable pageable){
        return articleRepository.findAllByBoardTypeList_BoardId(boardId, pageable);
    }
```

forBIT

Tech

JPA Query

```
Hibernate:
    select
        user0_.user_id as user_id1_28_0_,
        user0_.email as email2_28_0_,
        user0_.name as name3_28_0_,
        user0_.password as password4_28_0_,
        user0_.phone as phone5_28_0_,
        user0_.point_sum as point_su6_28_0_,
        user0_.role_code as role_cod9_28_0_,
        user0_.user_level as user_lev7_28_0_,
        user0_.username as username8_28_0_,
        role1_.role_code as role_cod1_20_1_,
        role1_.role_name as role_nam2_20_1_,
        privilege2_.role_code as role_cod1_21_2_,
        privilege3_.privilege_code as privileg2_21_2_,
        privilege3_.privilege_code as privileg1_15_3_,
        privilege3_.privilege_name as privileg2_15_3_
    from
        user user0_
```

Custom Query

```
@Modifying
@Query(value = "UPDATE Article a SET a.article_title =?1, a.article_contents =?2, a.article_update_date = now() WHERE a.article_id =?3", nativeQuery = true)
int updateArticle(String articleTitle, String articleContests, Long articleId);
```

forBIT

Tech

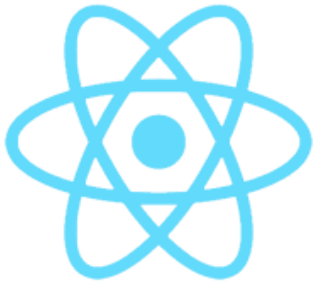
@RestController

```
@RequestMapping("/board")
public class ArticleController {

    @Autowired
    private LocalUserDetailsService userDetailsService;

    @Autowired
    private ArticleService articleService;

    @Autowired
    private BoardTypeListService boardTypeListService;

    @Autowired
    private UserIdToClassIdConverter userIdToClassIdConverter;

    // 게시물 작성
    // EndPoint :  http://localhost:8080/board/class_1_board/write
    @PreAuthorize("hasAnyAuthority('NOTICE_WRITE','JOB_WRITE','PRO_WRITE','CBOARD_WRITE','CNOTICE_WRITE','LIBRARY_WRITE')")
    @RequestMapping(
            path = "/{boardId}/write",
            method = RequestMethod.POST,
            produces = {
                    MediaType.APPLICATION_JSON_UTF8_VALUE,
                    MediaType.APPLICATION_XML_VALUE})
    public Article create(
                Principal principal,
        @PathVariable("boardId") String boardId,
        @RequestBody Article article) {
```

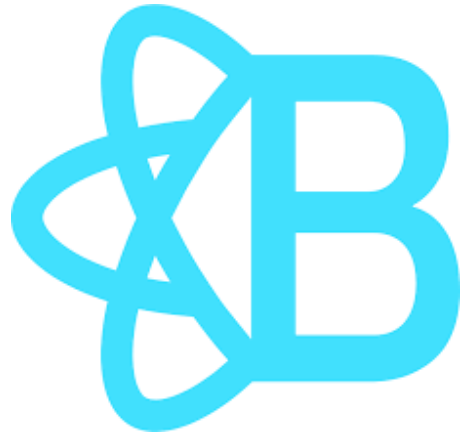
forBIT

Tech

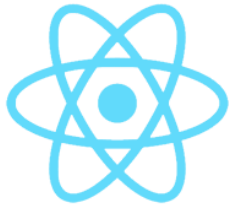
Front-End 주요 사용 기술



React



Tech

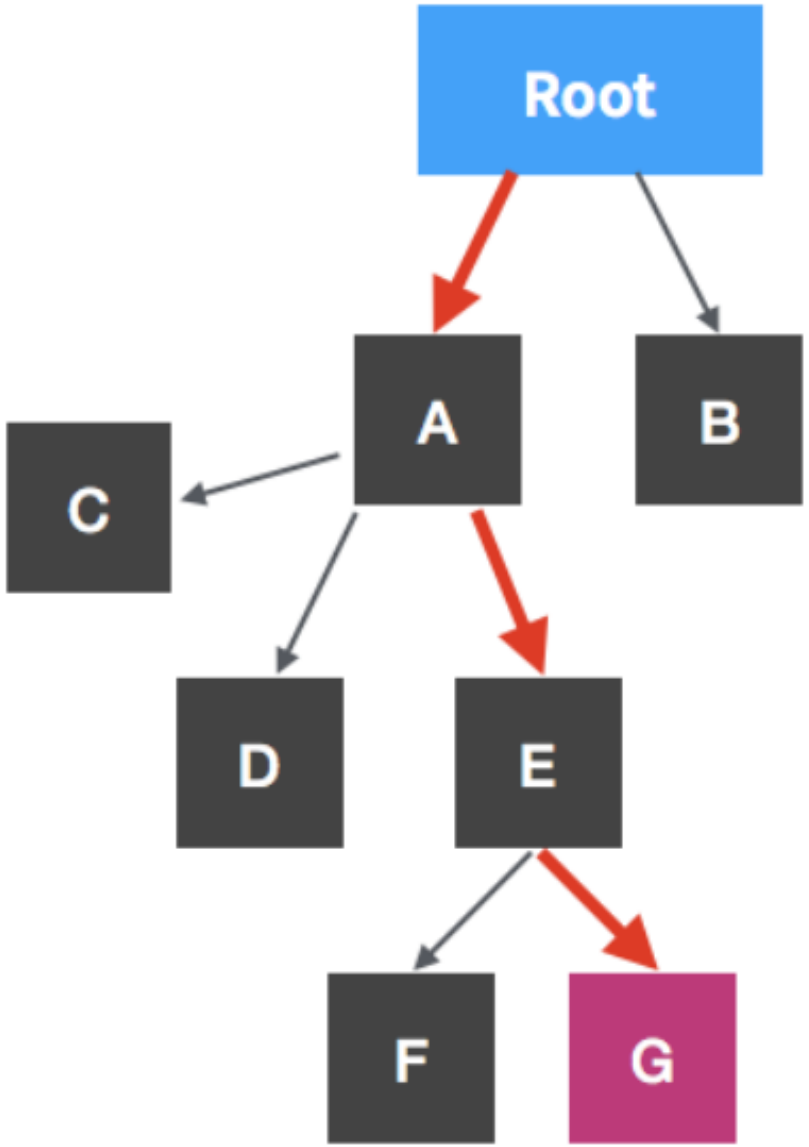


React

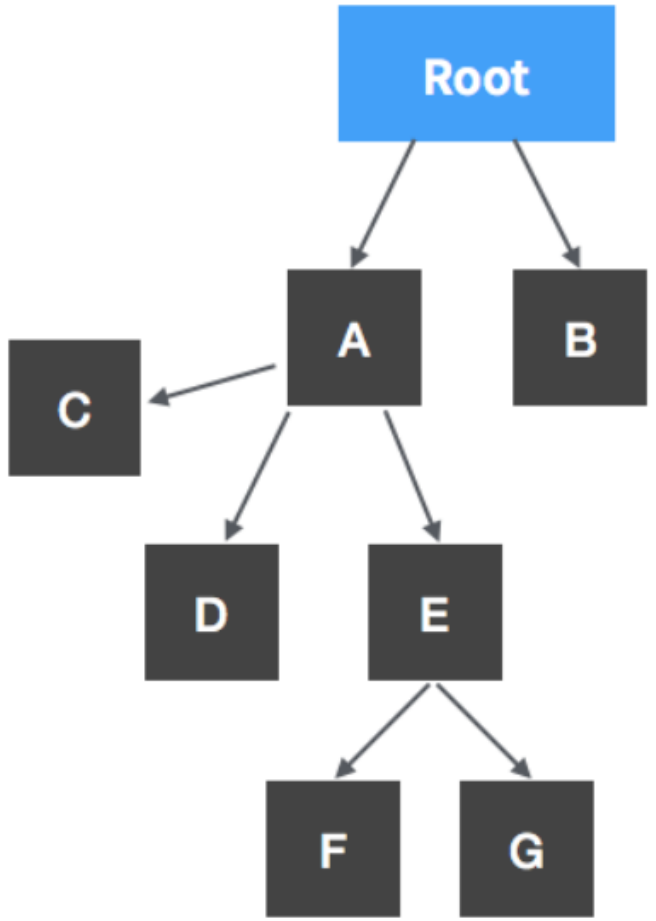
```
ReactDOM.render(
  <Provider store={store}>
    <Router history={hist}>
      <Switch>
        <Route path="/auth" render={props => <AuthLayout {...props} />} />
        <Route path="/admin" render={props => <AdminLayout {...props} />} />
        <Route path="/teacher" render={props => <TeacherLayout {...props} />} />
        <Route path="/student" render={props => <StudentLayout {...props} />} />
        <Route exact path="/login/compare/forgotpassword" component={FindPassword}/>
        <Route exact path="/login/compare/forgotid" component={FindUserId}/>
        <Route exact path="/login/compare" component={CompareUserInfo}/>
        <Route exact path="/" component={Login}/>
      </Switch>
    </Router>
  </Provider>,
  document.getElementById("root")
);
```

forBIT

Tech



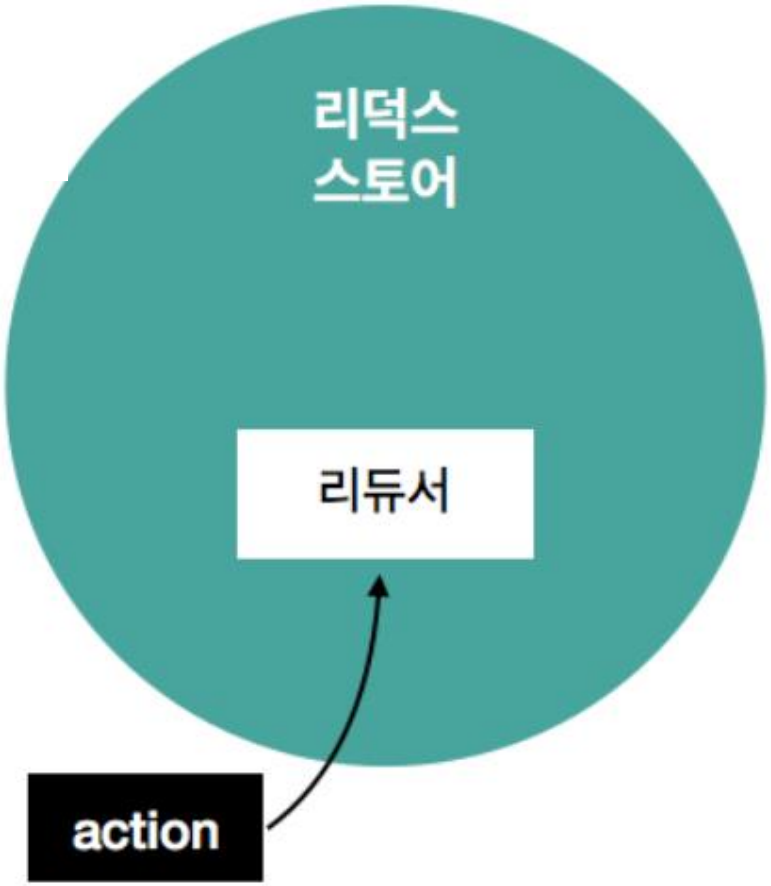
Redux

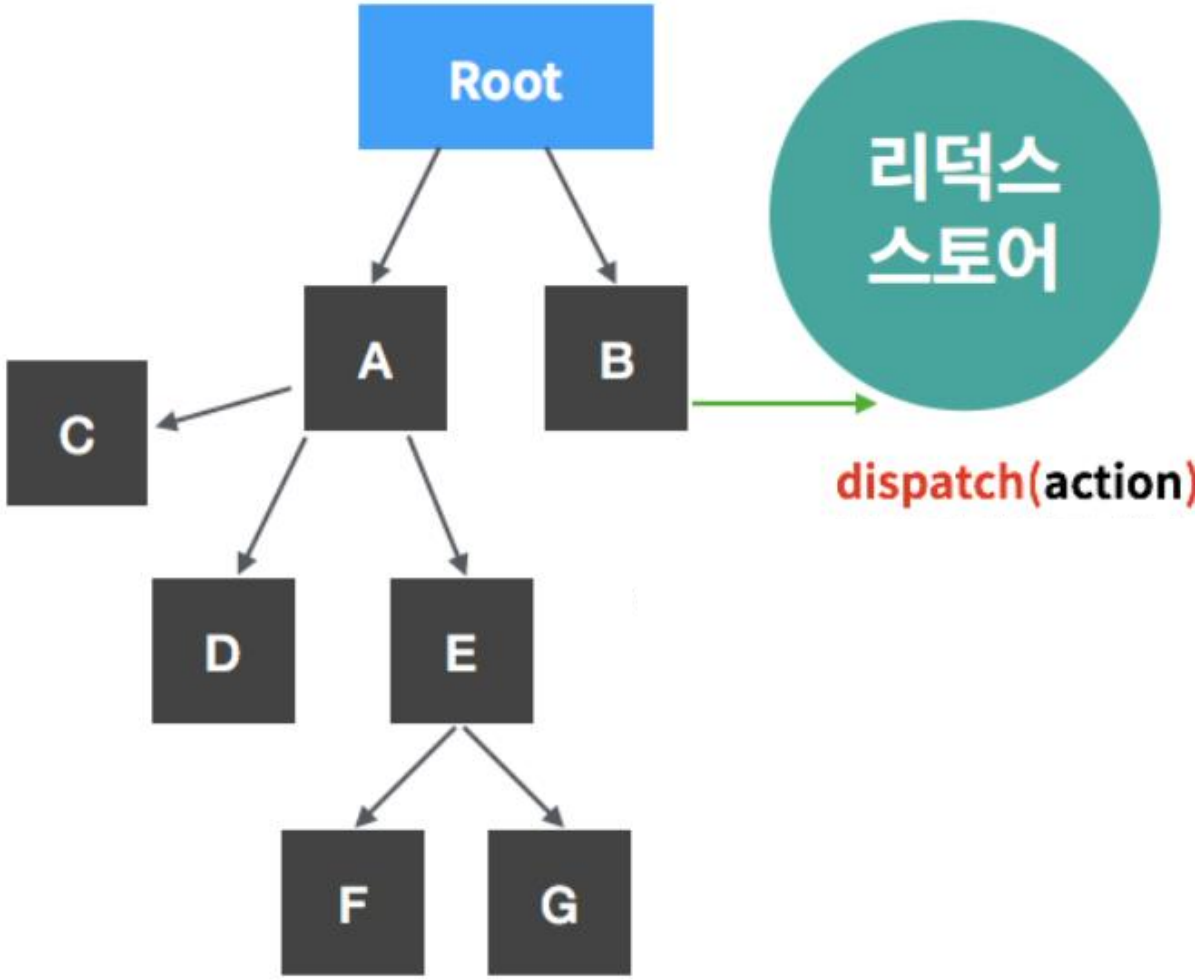


리덕스 스토어

Tech

Redux





Action

```
//Create
// EndPoint :  http://localhost:8080/board/class_1_board/write
const CreateArticle = (boardId, title, content) => {
  return ({
    type: ActionTypes.CREATE_ARTICLE,
    payload: {
      request: {
        method: 'POST',
        url: `/board/${boardId}/write`,
        headers: {
          'Content-Type': 'application/json; charset: utf-8'
        },
        data: JSON.stringify({ articleTitle: title, articleContents: content })
        //articleTitle =해당 엔티티컬럼명, : 뒤에 title = 리액트 내에서 사용된 변수
      }
    }
  });
};
```

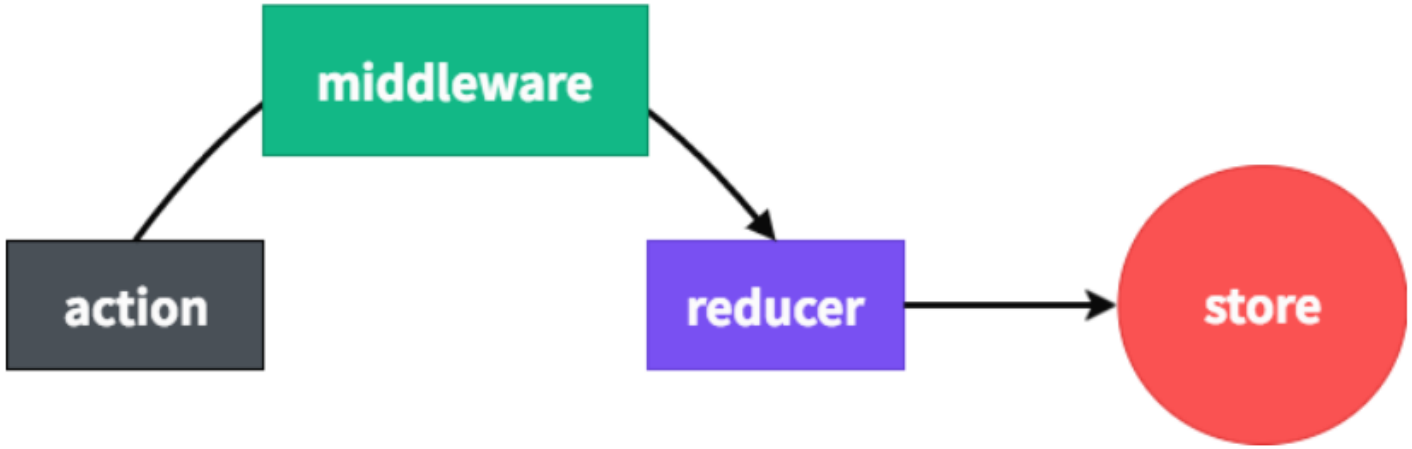
Reducer

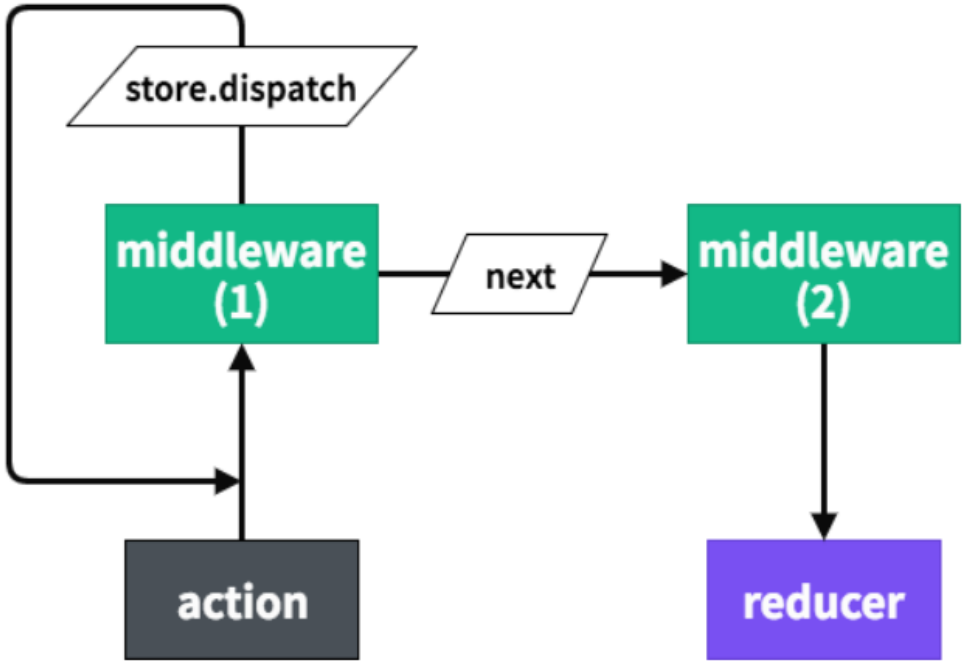
```
//The initial state
const initialState = {
  items: [],
  page: 1,
  size: 20,
  totalCount: 0,
};


const ArticleReducers = (state = initialState, action) => {
  const { items, totalCount } = state;
  const { payload } = action;

  switch (action.type) {
    case ActionTypes.CREATE_ARTICLE_SUCCESS :
      if (payload !== undefined && payload !== null) {
        return {
          ...state,
          items: [
            ...items,
            payload.data
          ],
          totalCount: totalCount + 1
        };
      }
      return state;
```

미들웨어

-axios -logger -thunk

```
const store = createStore(
  rootReducers, //combine해서 한번에 리듀서 받기
  stateLoader.loadState(),
  applyMiddleware(axiosMiddleware(client, middlewareConfig), logger, thunk)
);
```





```
const logger = createLogger({
  collapsed: true
});
```

redux-logger.js:388
redux-logger.js:392
redux-logger.js:401
redux-logger.js:377
redux-logger.js:388
redux-logger.js:392
redux-logger.js:401
redux-logger.js:377
redux-logger.js:388
redux-logger.js:392

logger

```
useEffect(() => {
  if (replyitems !== undefined) { //undeined 체크를 안해주면 무한패치됨
    if (replyitems.length === 0) { //아무것도 없으면
      listofReply(boardId, articleId) //댓글 가져오기
        .then(response => {//listofReply액션 성공했으면
          const { items } = response.payload.data; //꺼내서
          setItems(items); //set해주기
        })
        .then(response => {
          listofFiles(articleId)
            .then(response => {
              if (filedata !== []) {
                const filedatas = filedata
              }
              console.log(response)
              console.log(filedata)
            });
        })
        ;
    }
  }
});
```

thunk

```
const mapDispatchToProps = (dispatch) => ({
  retrieveArticle: (boardId, articleId) => dispatch(ArticleActions.RetrieveArticle(boardId, articleId)),
  deleteArticle: (articleId) => dispatch(ArticleActions.DeleteArticle(articleId)),
  listofReply: (boardId, articleId) => dispatch(ReplyActions.ListofReply(boardId, articleId)),
  listofFiles: (articleId) => dispatch(FileActions.ListOfFiles(articleId)),
  downloadFiles: (filename) => dispatch(FileActions.DownloadFiles(filename))
});

export default withRouter(connect(mapStateToProps, mapDispatchToProps)(RetrieveArticles));
```

axios

```
const client = axios.create({
  baseURL: 'http://localhost:8080',
  headers: {
    'Authorization': `Basic ${btoa(`${clientId}:${clientSecret}`)}`,
    'Cache-Control': 'no-cache'
  },
  responseType: 'json'
});

const middlewareConfig = {
  interceptors, //권한주는 미들웨어
  onError: onErrorHandler
};
```

ReactStrap

```
import {
  Card,
  CardHeader,
  CardBody,
  Table,
  Row,
  Col,
  Container,
  Form,
  FormGroup,
  Input,
  Label,
  Button
} from "reactstrap";
```

```
<Row>
  <Col>
    <Card>
      <CardHeader>
        <h5>게시판 글쓰기</h5>
      </CardHeader>
      <CardBody>
        <Form onSubmit={e => this.onSubmit(e)}>
          <FormGroup className="has-success">
            <Input type="text" value={this.state.title} onChange={this.onChangeTitle}
          </FormGroup>
          <FormGroup className="has-success">
            <ToastEditor
              content={content}
              onChange={this.onChangeContent}
            />
          </FormGroup>
        </Form>
      </CardBody>
    </Card>
  </Col>
</Row>
```


forBIT
사업


forBIT
소감발표


forBIT
감사합니다
4BIT TEAM
류혜영, 이중호, 이채연, 조성현, 주영빈, 홍다경, 황서영