

백엔드 API 구현 목록

로그인 페이지

POST

로그아웃

Post

회원가입

Post

스터디 모집 페이지

GET

POST

마이페이지

GET

로그인 페이지

POST

- `/login`
 - LoginDto에 맞추어 로그인하는 객체를 보내주었을 때 회원 정보와 일치하면 토큰 발급 후 로그인 처리

```
/**
 * 로그인 폼
 */
@Data
public class LoginDto {
    @NotEmpty
    private String loginId;
    @NotEmpty
    private String password;
}
```

로그아웃

Post

- `/logout`
 - 토큰 삭제

회원가입

Post

- `/add`
 - `RequestBody`로 `Member`객체를 보내주면 회원 정보 save

스터디 모집 페이지

GET

- `study/recruit`
 - 전체 스터디 리스트
 - `Page<study>`
 - `SearchDto`에 페이지 번호를 보내주어야 한다.
- `study/recruit/it-programming`
 - 개발, 프로그래밍 스터디 리스트
 - `Page<study>`
- `study/recruit/it`
 - 보안, 네트워크 스터디 리스트
 - `Page<study>`
- `study/recruit/game-dev`
 - 게임 개발 스터디 리스트
 - `Page<study>`
- `study/recruit/creative`
 - 크리에이티브 스터디 리스트
 - `Page<study>`
- `study/recruit/academics`
 - 학문, 외국어 스터디 리스트
 - `Page<study>`

- `study/recruit/career`
 - 커리어 스터디 리스트
 - `Page<study>`
- `study/recruit/life`
 - 자기계발 스터디 리스트
 - `Page<study>`
- `study/recruit/search/{keyword}`
 - 스터디 검색시에 백으로 검색어를 Pathvariable로 전송해주면 해당 검색어를 포함하고 있는 스터디 리스트 반환
 - `List<study>`

POST

- `study/new`
 - StudyDto객체에 맞추어 형식을 보내주면 스터디 개설

```
/**
 * 스터디 생성 폼
 */
@Data
public class StudyDto {
    @NotEmpty
    private String name;

    @NotEmpty
    private String category;
}
```

마이페이지

GET

- `/members/member`
 - 회원 정보를 return
 - `Member`

