

스프링부트로 **RestFulAPI** 구현하기

4장 기본 **Springboot** 이해하기

- **UserController**
- **UserService**
- **UserRepository**
- **User**

박명회

2장 클래스 설명

User: 사용자 정보를 담는 엔티티 클래스입니다.

UserRepository : **JpaRepository**를 상속받아 데이터베이스와의 상호작용을 담당합니다.

UserService : **UserRepository**를 주입받아 비즈니스 로직을 처리합니다.

UserController : **HTTP** 요청을 처리하고, **UserService**를 통해 사용자 데이터를 반환합니다.

4장 UserController.class

@RestController

@RequestMapping("/users")

public class UserController {

private final UserService userService;

@GetMapping

public List<User> getAllUsers() {

return userService.getAllUsers();

}

@GetMapping("/{id}")

public User getUserById(@PathVariable Long id) {

return userService.getUserById(id);

}

@PostMapping

public User createUser(@RequestBody User user) {

return userService.saveUser(user);

}

}

4장 UserService.class

@Service

public class UserService {

private final UserRepository userRepository;

//모든 유저를 불러오는것

public List<User> getAllUsers() {

return userRepository.findAll();

}

// 한명의 유저 가져오기

public User getUserId(Long id) {

return userRepository.findById(id).orElse(null);

}

// insert update 구문

// id가 없을때는 insert , id가있을때는 update

public User saveUser(User user) {

return userRepository.save(user);

}

}

4장 **UserRepository.class & User.class**

UserRepository.class

```
public interface UserRepository extends JpaRepository<User, Long> {  
  
}
```

User.class

@Entity

```
public class User {  
  
    @Id  
  
    private Long id;  
  
    private String name;  
  
    private String email;  
  
    // getters and setters  
  
}
```

감사합니다