



PMP 2차 개인목표

목표

- 무작정 코드만 쓰는 것에서 벗어나 개발의 효율과 생산성을 고려하고, 코드 외적인 부분 까지 생각하며 진짜 '개발'을 하는 백엔드 웹 개발자 되기

Goals

- Current
 - 지금까지 사용한 기술 스택(언어, DB) → 지나친 반복, API 연동 등의 작업에 과한 시간 투자 → 서버 개발의 효율성과 생산성이 부족한 상태
 - 작은 규모의 서버 개발 경험 -> 큰 규모의 서버에서 발생할 수 있는 서버 성능에 따른 부하 정도를 고려하지 않고 서버를 개발하는 상태
- After
 - 효율성•생산성 있는 개발에 도움을 받을 수 있는 기술 스택(프레임워크 등)을 도입 해보는 경험 → 서버 개발 시에 효율성과 생산성을 고려할 수 있는 상태
 - 큰 규모의 서버에서 발생할 수 있는 서버 성능에 따른 부하 정도를 고려하며 서버를 개발할 수 있는 상태

1. 서버 개발 시의 효율성과 생산성에 도움을 줄 수 있는 프레임워크를 도입해보기

- 효율성•생산성 있는 개발을 판단하는 기준
 - 1) 프레임워크 도입 자체가 추구하는 개발에 좋은 영향이 되는지
 - 2) 좋은 영향이라면, 해당 프레임워크를 계속 사용할 것인지

2. 서버의 성능 개선을 고려한 개발하기

How To Work

1. 프레임워크 학습 및 적용

- 방식 : Spring 中 팀프로젝트 개발에 필요한 핵심 기술을 강의로 빠르게 학습 → 학습 내용 : 개인 Notion에 문서화

2. 서버의 다양한 성능 개선 방식 파악

- 방식 : 다양한 방법 사용(공식 문서, 구글링, 관련 서적 등) -> 정리된 내용은 간단하게 문서화

3. 성능 개선 방식에 따른 서버 동작, 처리 과정의 차이 이해 및 명세

- 방식 : 각 서버에 최선인 성능 개선 방식에 대한 고민 → 과정, 결과를 1번 문서에 통합

4. 성능 테스트 적용을 위한 연습

- 방식
 - 개발하고자 하는 서버의 '일반적인' 부하 테스트 시나리오 작성 연습
 - 성능 테스트 툴(nGrinder 등)의 사용 → 시나리오에 의한 부하 테스트 진행(User, TPS, Time 등의 지표 활용, 테스트 타겟 시스템의 범위 등 고려)
 - 필요하다 판단되는 내용만 개인 Notion에 문서화

5. 서버에 성능 테스트 적용

- 방식 : 적용할 각 서버 성격에 맞는 성능 테스트 시나리오를 설계 및 작성해 부하 테스트 진행
- 다양한 시나리오를 작성해보고, 팀원들과 협의 과정을 반드시 거쳐 각자 개발한 서버의 모든 범위에서 문제가 없는지 검증할 것.
- 각 서버의 성격에 맞는 성능 테스트 시나리오를 직접 작성하여 부하 테스트 진행

Plan

- Spring 강의를 통한 프레임워크 학습 ~1/13
- 인증(로그인, 회원가입 등), 유저 서버 아키텍처 설계 및 수정 ~1/11

- 성능 테스트 관련 공부 ~1/15
- 인증 서버 구현 ~ 1/22
- 유저 서버 구현 ~ 1/29
- 그외 자잘한 부가 기능 구현 ~ 2/5
- 성능 테스트 및 개선 ~ 2/18