



# 이하얀 PMP(23.01.02.ver)

## ▼ Goals(Current & After)

- Current

웹 백엔드 개발자가 되기 위한 '나'의 기술 스택을 결정하지 못한 상태

ex) 기술 스택 의미 : 언어, 데이터베이스, 프레임워크 등의 집합

웹 백엔드 개발을 위한 기초 이론만 학습하고, 서버를 개발하고 배포하여 서비스에 적용을 위한 지식과 실습 경험이 없는 상태

ex) API, 웹 프레임워크, 서버 등의 지식과 실습에 대한 경험 부족

팀프로젝트를 위한 알림 서버와 상태 확인 서버의 동작 방식을 잘 모르고, 더 나아가 여러 사람을 통해 다양한 기술 스택으로 개발된 서비스 간 통신 방법을 모르는 상태

- After

기술 스택을 결정하고, 실제 웹개발에 **적용**이 가능한 상태

실습 경험을 쌓아 **서버 개발 및 구축, 배포**가 가능한 상태

알림과 상태 확인 서버의 동작 방식 및 구축 방법에 대해 **이해**하고 **활용**할 수 있는 상태

→ 서버들 간의 통신에서 발생하는 다양한 문제를 효율적으로 조절할 수 있는 **MSA** 등의 아키텍처를 활용할 수 있는 상태

## ▼ Goals(순서)

1. 프레임워크 사용으로 개발 시간 단축

기술 스택을 결정하기 위해 언어와 데이터베이스는 이미 활용 가능한 수준

그러나, 개발에 있어 효율적인 도구는 사용하지 않았기 때문에 시간이 다소 오래 걸리며 이 문제는 배포 시간에 큰 영향을 미치기도 함.

따라서 프레임워크라는 효율적인 도구를 사용해 웹 개발의 모든 방면을 효율적으로 제어할 수 있는 기술 스택을 구축하고자 함.

## 2. 서버의 개발 및 배포 경험

그동안, 서버를 구축하기 위한 개념적인 학습을 했음.

하지만 실습에 대한 경험은 리눅스 OS를 이용하여 APM 서버를 구축하는 정도에 그치는 수준이었으며, 실제 개발 및 배포에 대한 경험은 없는 상태임.

그렇기에 서버의 개발 및 배포를 경험하여 실제로 웹서비스를 제공할 수 있는 상태가 되고 싶음.

## 3. MSA를 통한 규모있는 서비스 설계 경험

규모있는 서비스는 대부분 혼자 하기 보다는 다양한 사람들과 팀을 이뤄 협업으로 구축하는 것이 대부분임.

그렇기에, 이번 캠프를 통해 팀원들과 협업하며 서비스를 개발하기 위해 MSA 아키텍처를 학습하고, 사용하고자 함.

### ▼ How To Work

#### 1. 프레임워크 학습 및 적용

- 주언어인 Java를 기반으로 하는 Spring 프로젝트(Spring Boot, Spring Security, Spring Batch 등)를 선정해 빠르게 학습할 수 있도록 하기
- **전문가를 위한 스프링5** 교재로 학습 예정

#### 2. MSA 아키텍처 학습 및 적용

- **Spring Cloud로 개발하는 마이크로서비스 애플리케이션(MSA)** 강의를 통해 학습하기
- 완강 후에도 이해가 잘 되지 않는다면, 주저하지 않고 팀원인 다른 백엔드 개발자와 소통하며 함께 이해하기 위해 노력하기
- 더 나아가 성능 테스트를 통해 MSA 설계가 잘 이루어졌는지 팀원들과 함께 확인하며 프로젝트 진행하기

### 3. 알림 서버 및 상태 확인 서버에 대한 조사

- 디스코드 클론코딩이기 때문에 기본적으로 **Discord 기술블로그** 자료를 적극 이용해 정리하여 문서화해두기
- 그 외 필요하다면 또 다른 **공식문서, velog, github** 등 다양한 예시들을 통해 이해하고, 문서화해두기

### 4. 알림, 상태 확인 서버 구축 및 타 서버와의 통신

- 어떤 방법을 사용해야하는지에 대한 조사 후 → 알림 서버, 상태 확인 서버 구축하기
- 타 서버와의 통신을 위해 사용하는 MSA 설계에 적극 참여하여 팀에 반드시 도움이 되기 위해 노력하기

#### ▼ Planning

- Spring5 학습 및 Spring Cloud를 이용한 MSA 아키텍처 학습 및 설계 (~ 01.26)
  - 점검 : 매일 진도율 계산해 하루 진도율을 매일 확인
- 알림 서버 및 상태 확인 구축 방법 조사( ~ 01.26)
  - 점검 : 문서화된 자료
- 알림 서버 개발( ~ 02.02)
  - 점검 : 매일 개발 회고 작성하기
- 상태 확인 서버 개발( ~ 02.09)
  - 점검 : 매일 개발 회고 작성하기
- 성능 테스트 및 개선(~ 02.16)
  - 점검 : 미정

피드백

공부 하겠다 -> 어떻게 개발하겠다  
과정을 자세히 적겠다  
어떤 단계까지 하고 피드백을 받겠다 같은 내용

#### ▼ How To Work

##### 1. 기술 스택 학습 및 적용

- 주언어인 Java를 기반으로 하는 Spring 프로젝트의 핵심 기술을 강의 학습과 교재 발췌 형식으로 빠르게 학습하기
- 학습 및 실습 내용 + 리뷰 : 개인 Notion에 문서화해 정리하기

##### 2. MSA 아키텍처 학습 및 적용

- **Spring Cloud로 개발하는 마이크로서비스 애플리케이션(MSA)** 강의를 통해 학습하기
  - 섹션 단위로 개인 Notion에 문서화하기

##### 3. 알림, 상태 확인 서버에 대한 조사 및 서버 구축 & 타 서버 통신

- 조사
  - 디스코드 클론코딩 → **Discord 기술블로그, 공식문서, velog, github 등** 다양한 예시들을 통해 이해하고, 개인 Notion에 문서화하기
- 서버 구축 방식
  - 순서 : 아키텍처 설계 → API 문서 작성 → API 서버 구현
  - 비동기적 방식으로 진행, 각 단계별로 팀원 또는 캠프 관계자를 통해 피드백 예정
  - 개발 고려사항을 따져 적절한 아키텍처 선택//이 말 빼는게 나올까요?
  - 문제 발생 시 해결 과정은 개인 Notion에 기록하며 당일에 해결 → 미해결 시 코어 타임을 활용해 팀원에게 도움 요청하기
- 타 서버 통신
  - 학습한 MSA 설계 방식 적용 → 전체 서버의 아키텍처, 설계 과정 등의 모든 과정을 특정 서버의 특성을 염두하여 진행하기(추후 더 자세하게 고민)
  - 성능 테스트 → 시간, 구조 등이 기준치(추후 설정)에 부합하는지 여부를 체크하고 변경이 필요할 경우 반드시 팀원 피드백 진행 후 적용하기(코어 타임 or 그 외 시간)

##### 4. 학습 시간에 너무 많은 시간을 허비하지 않을 것

- 개발에 필요한 수준으로 분량을 조절하여 과하게 학습만 하는 시간 지양하기

- 내가 가지고 있는 스택을 활용하고, 과하게 새로운 기술 학습 및 도입은 지양하기