

## Intro - 첫 번째 세션에서

---

### 모호한 과제가 아니다, 어려운 과제일뿐.

1. 먼저 토스의 일하는 방식을 소개받았다. 자료를 보고 다시 정리해야겠지만, 구체적으로 기억나는건
  - 내가 배포한 것들의 임팩트 측정하기
  - 그러기 위해 레퍼런스와 자료, 내가 사용하는 도구를 살피기 등
  - 구체적으로 데이터와 함께 의사소통하기
  - roi를 고민하며 움직이기 등등이 있다.
2. 그 다음은 실습 과제이다.
  - 터무니 없는 과제를 받았다. 처음에는 모호해보였고, 안된다고 하는 생각이 많이 들었다.
  - "내가 지금 가지고 있는 태스크가 몇개지?", "할당받은 태스크 없이 내가 역제안 한 적이 있었나?" 와 같은 고민이 주를 이뤘다.
  - 특히 우리는 신규개발팀으로 밀려드는 태스크를 처리하기에도 벅찬 팀이기 때문에 더욱 그랬다.
3. 그래도 같이 보니까 보이더라.
  - 내생각에 경력직 개발자들을 모아놓고 이사람들이 지금 태스크가 있고 일정이 있고 어쩌고를 생각하지 못했을 리 없다.
  - 결국 토스가 일하는 방식으로 회사에서 일해보라는 것 자체가 주요한 과제이다.

결국 위의 두가지를 조합하면, 능동적으로 문제를 정의해야하고, 해당 문제를 해결하고, 해결했다는것을 수치와 함께 증명하며 커뮤니케이션하고,  
해결된 나의 코드의 임팩트를 정의하고, 위와 같은 증명과 커뮤니케이션을 통해 상위직책자를 설득하고, 일을 끝내라는 것 까지  
가 과제이다.

위와 같이 일하면 토스가 같이 일하고 싶어하는 주요 지표를 모두 달성하게 된다(기술, 운영, 제품, 커뮤니케이션 등)  
즉 명확한 과제라고 생각이 들었다.

## Days - 러너스 하이를 진행하면서 해왔던 것들

---

### 1주차, 2주차 작업내용

1. 기존에 잡혀있던 프론트엔드 태스크 진행
  1. 기존 커머스 플랫폼에 커뮤니티 기능을 추가하는 작업 수행.
  2. 레거시 CMS의 뷰 및 관련 API 작업과 프론트엔드 페이지 개발 완료.
  3. 개선점 :
    1. 검색 조건을 router에 push하고 lazyQuery로 감지하는 기존 로직을 리팩토링.
    2. 상태값을 분리하여 Apollo Client의 refetch, invalidate query key를 활용해 불필요한 깜빡임 및 중복 요청 제거.
2. 백엔드 테스트 코드 도입 및 커버리지 개선 (10%)
  1. 배경: 전시영역을 담당하는 Hasura 기반의 백엔드 프로젝트에 테스트 코드가 거의 없는 상태(0%대).
  2. 문제점
    1. 리졸버 핸들러가 여러 번 외부 컴포넌트를 호출하는 문제.
    2. 테스트 코드 작성 시 외부 API 요청을 적절히 모킹할 수 있도록 구조 개선 필요.
    3. Hasura가 기본적으로 트랜잭션을 지원하지 않아 데이터 정합성 이슈 발생.
    4. typedefs가 로우 텍스트로 관리되어 오타 및 구조적 오류 발생 가능성 높음.

### 3. 개선사항

1. 핸들러 구조 개선 : 리졸버 내부의 외부 요청을 별도 함수로 분리하여 모듈화
2. 테스트 환경 보호 : jest설정을 변경하여 테스트 실행 시 운영 데이터에 영향이 없도록 방어로직 추가.
3. 트랜잭션 처리 개선 : Hasura 트랜잭션 대신, 요청에 맞는 보상트랜잭션 추가
4. typedefs 관련 검증 옵션 추가

1주차, 2주차동안의 상세 작업 내용과 후기 링크 -> [Learners-high-1](#)

## 2주차, 3주차 작업 내용

1. 배경 : 인증관련 이슈 수정작업 진행
2. 문제점
  1. 멀티스레드 환경에서 로그가 섞여 불편했었음
  2. 특정 사용자에게 대한 연속성 로그를 보기 힘든 구조
  3. 빈약한 로깅 개선
3. 개선사항
  1. 로그와 요청 식별값 추가
  2. 레거시 로그 라이브러리 버전에서 환경에서 로그 레이아웃 커스텀하기
  3. PII를 고려한 식별값 추가
  4. 범용적으로 적용, 추가될 로그 관련 성능이슈 수정
  5. 전반적인 로깅 보완

2주차, 3주차동안의 상세 작업 내용과 후기 링크 -> [Learners-high-2](#)

## 3주차, 4주차 작업 내용

1. 배경 : 인사이동으로 인한 OJT 진행, 새로운 팀의 프로젝트를 공부하던 중 메인 프로젝트가 헥사고날 아키텍처로 구현되어있는 메세지 스트리밍 위주의 프로젝트인 것 을 확인
2. OJT 기간을 치열하게 보낼 수 있도록 온보딩 목적의 사이드 프로젝트 진행
3. 카프카를 헥사고날 아키텍처로 클론코딩하기
4. Kraft metadata stroe, Handshake성 인터페이스들 프로토콜대로 구현
5. 세그먼트 단위로, 로그기반 메세지 저장하는 로직 구현
6. producer api, consumer api 몇가지 구현

3주차, 4주차동안의 상세 작업 내용과 후기 링크 -> [Learners-high-3](#)

## 그 외 이슈들

단순 운영 작업, 혹은 빠른 시간내에 완수해야 하는 작업들도 병행하며 진행했습니다.

1. 커뮤니티 기능 개발 프론트엔드 인가 분리 작업
2. 커뮤니티 기능 개발 기획 누락/변경 대응
3. 사이트맵 현행화 작업

## Outro - 후기

### 엄청나게 바쁘고 하드코어하게 달려왔다.

휴일 반납은 기본이고, 매일 밤늦게까지 해왔다.

주차별 회고에 정리하지는 않았지만, 저기 작성된 일 외에도 운영이슈들이나 기획 변경/누락 대응과 같은 작업들도 있었고 종합적으로 봤을때 무리일정도로 열심히 했다.

물론 진짜 징하다 싶을 정도로 여건이 안받쳐 주는 일도 있었다. 예를들어 인사이동이라던지, 회사 내부 사정으로 인한 사태라던지..

결과가 만족스러운지를 봤을 때, 완벽하지는 않지만 내가 처한 상황에서 최선의 선택과 최고의 노력을 했다고 스스로 자평할 만큼은 노력해온 것 같다.

## 실제로 느끼는 것도 많았다.

항상 퇴근하고 많은 사이드프로젝트나 개인 공부를 하면서 항상 막연하게, 회사의 목표와 내 목표나 비전이 일치하면 좋았겠다라는 생각을 했다. 퇴근하고 해오던 것들이 회사에서 하는것들과는 달라서 퇴근하고 하는 집중을 출근해서 이어가지 못하는게 아쉬웠던 적이 많았기 때문이다.

그 부분이 회사 탓이라고 스스로 자평해왔던 부분도 분명히 있었던 것 같다.

'매일 프론트 업무 쉼어서 주장아..', '빨리 끝내면 다른 프론트일 주겠지..', '일정도 촉박한데 무슨 고도화/자동화야..'와 같은 생각이 많았었는데, 이번을 계기로 어느정도 생각의 흐름이 바뀐 것 같다.

내가 진짜 하고싶어서 공부해왔던 일들과 회사일들을 같은 선상에서 바라볼 수 있도록 된 것 같다.