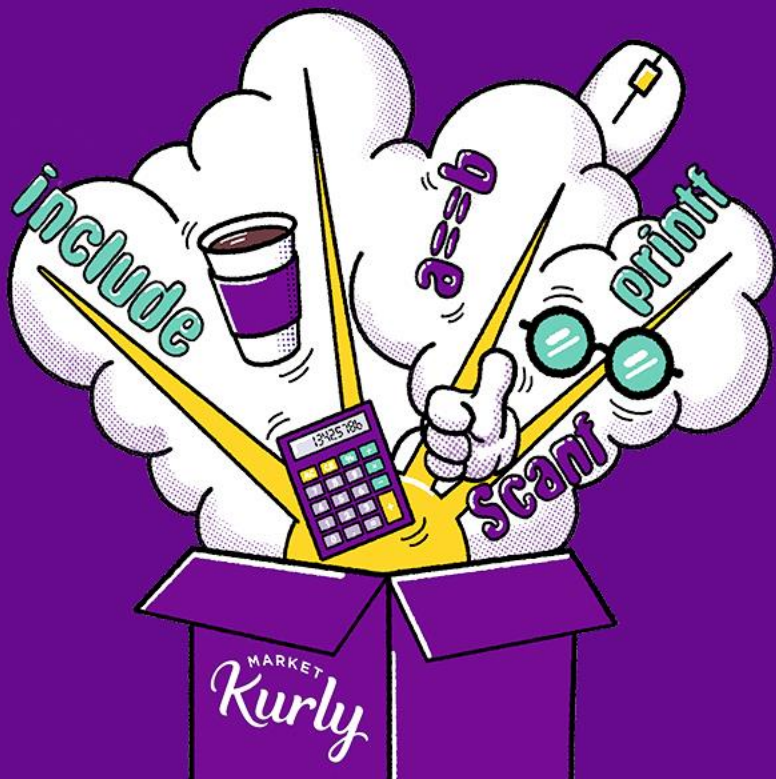


MARKET KURLY DEVELOPER MEET UP

FUN 하지만 뻔하지 않은, 컬리의 개발자 성장기



컬리 개발자 성장기

컬리 도메인별 현업 개발자의 성장 스토리를 소개합니다.





컬리 개발자성장기

첫번째 이야기의 주인공

저는

“컬리에서 상품을 구매하는 과정이 즐거울 수 있게 만드는 개발자”

김민중 입니다.



> 김민중님 | 커머스플랫폼 웹개발

[컬리에서 서비스와 함께 자라기]

“컬리 웹 프론트 개발 조직엔 흥미로운 문제 정의들이 많습니다. e-commerce 서비스에서 레거시를 걷어내고, 새로운 시스템을 만들어가는 성장 이야기를 나누고 싶습니다.

또한, 컬리에선 기술위원회가 열리는데요, 기술위원회는 '왜 그리고 무엇을 위해' 탄생했는지 소개합니다.”





첫번째 이야기-김민중님

컬리에 합류하기 전, 저의 경험을 소개합니다.

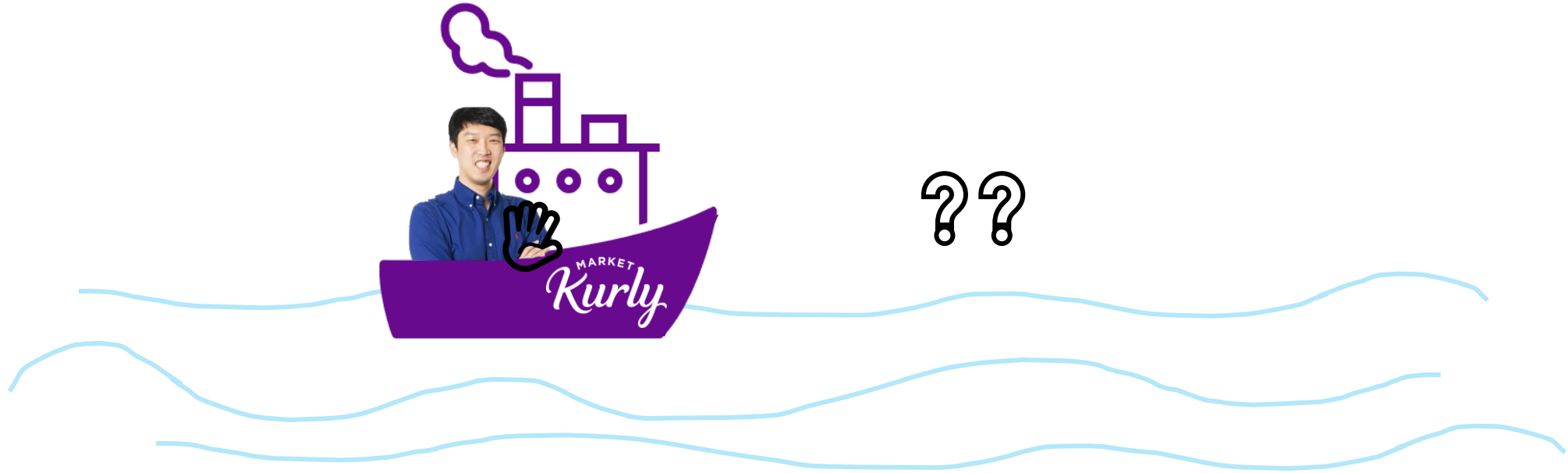
NAVER





첫번째 이야기-김민중님

그렇게 저는 컬리에 합류하게 되었습니다.





첫번째 이야기-김민중님

“컬리에 왜 합류하셨나요?” 라는 많은 질문에 대해 이렇게 답합니다.



내가 컬리에 합류한 가장 큰 이유?
= 커리어를 성장시키는 것
※ 커리어란? 내가 조직에서 어떤 부분을 기여했는지가 기록되는 것



커리어에서 무엇보다 중요한 것!
= 내가 했던 일 < 내가 한 일이 회사에 '어떻게' 기여했는가



= 내가 기여할 수 있는 부분이 정말 크겠군!!





첫번째 이야기-김민중님

'경험한 것'과 '경험을 만드는 것'의 차이



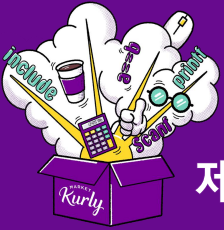
어느 쪽이 더 개발자로서 훨씬 높은 성장을 하게 될까요?



코드리뷰를 하지 않던 조직에 코드리뷰 문화를 정착시키는 것

- ✓ 코드리뷰가 필요한 이유 & Practice 정립
- ✓ 동료들 설득하고 동의를 얻는 과정

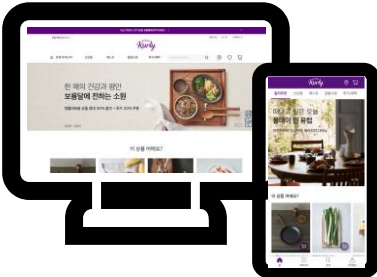




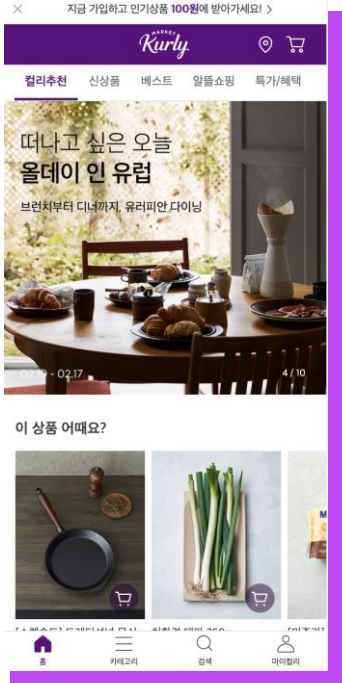
첫번째 이야기-김민중님

제가 속한 커머스 플랫폼 - 웹 프론트 팀은,

커머스 플랫폼 - 웹 프론트 개발



PC, 모바일 웹 도메인에서
고객분들이 마켓컬리 몰에서 **물건을 구매하는 전 과정**을 책임지는 팀입니다.





첫번째 이야기-김민중님

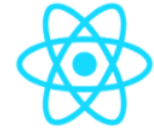
저희 팀의 테크 스택을 소개합니다.



웹 프론트엔드 팀에서는 아래의 테크 스택을 사용하고 있습니다.



Redux



ReactJS



Storybook



ReactiveX



NextJS



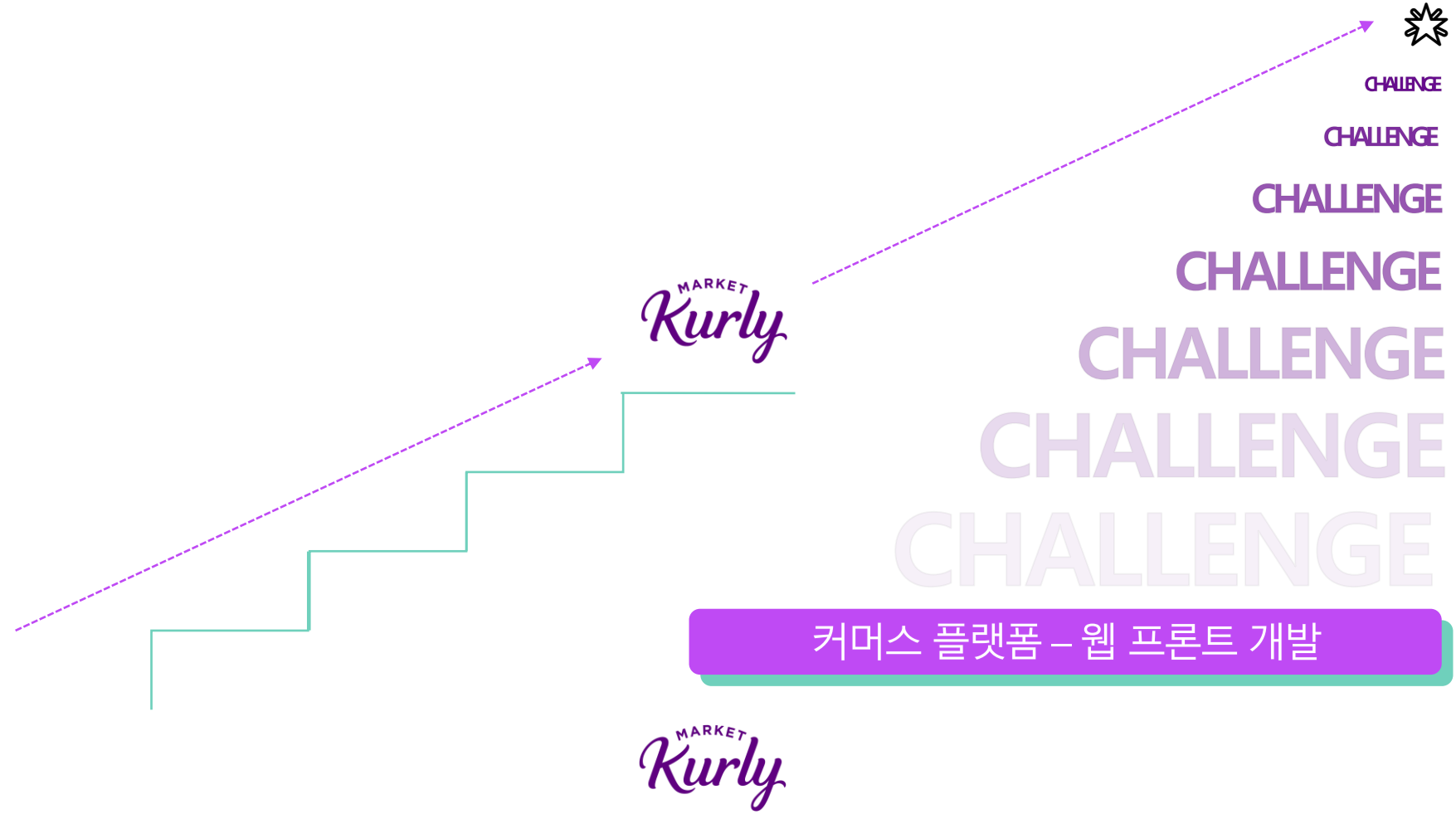
TypeScript





첫번째 이야기-김민중님

컬리에서 여전히 더 도전하고 만들어 나가야 할 과제는 많습니다.





첫번째 이야기-김민중님

저희 팀의 굵직굵직한 과제를 들여본다면,

레거시 → MSA

BFF 설계 및 구현

디자인 시스템 구축

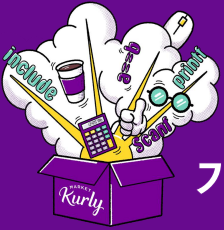


그 외 비즈니스 요구사항을 효율적으로 처리하는 방법!



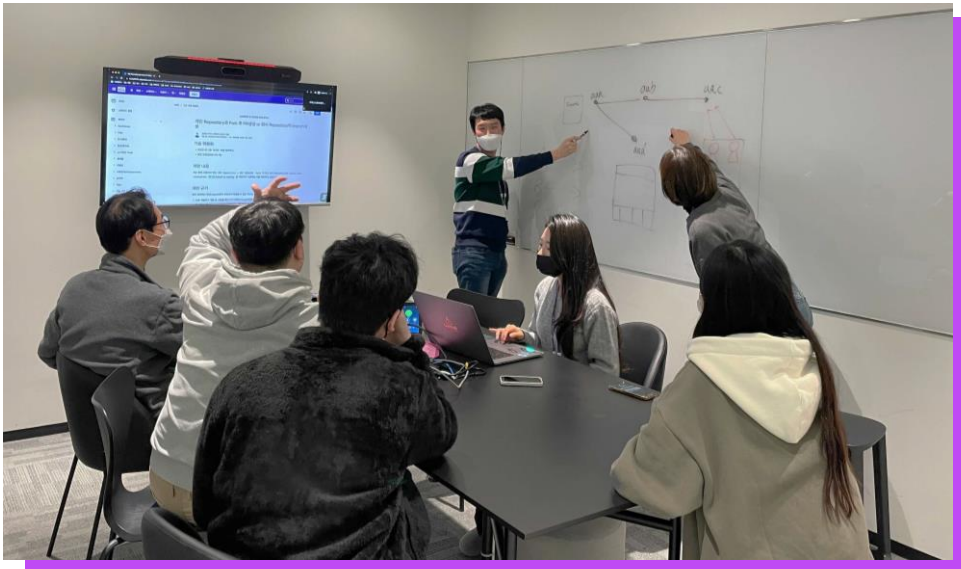
이 모든 과제들을 비즈니스 요구 사항을 구현해 나가면서 함께 성장시켜 나가야 하죠.






첫번째 이야기-김민중님

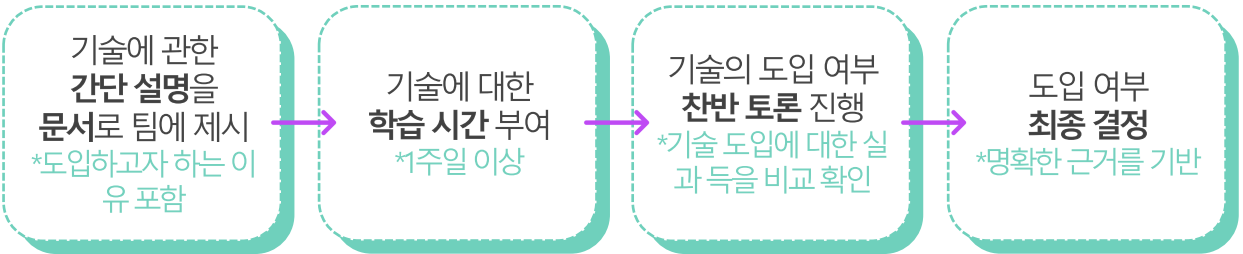
기술위원회를 소개합니다!



 언제 열리나요?

- 새로운 기술을 도입할 때
- 기존의 사용하던 기술을 새로운 기술로 대체할 때

 어떻게 진행되나요?



자발적으로 근거를 만들고 그 근거에 따른 결정을 하게 되는 것이며 이러한 문화는 개발자 성장에 큰 도움이 되는 문화입니다.





첫번째 이야기-김민중님

컬리에서 얻을 수 있는 높은 가치

컬리에서는 높은 기여를 통한 빠른 성장을 할 수 있습니다.

Why?

구현이 필요한 과제 有
설계 단계부터 참여 가능

신규 기술 도입에 대한 근거를
만드는 과정을 통해 성장 가능





첫번째 이야기-김민중님

저희는 이런 동료의 합류를 기다려요!



성장을 하고 싶은 개발자



객관화 근거를 기반으로
합리적인 결정을
팀에서 할 수 있도록 돕는 개발자





컬리 개발자성장기

두번째 이야기의 주인공

저는

“느리더라도 꾸준히 성장 하고픈 거북이 같은 개발자”

방신철 입니다.



저는 조금 더 나은 세상을 만들기 위한 서비스를 개발하는 것 그리고 트렌드에 맞는 새로운 기술을 적용하여 개발하는 것을 좋아합니다.

> 방신철님 | 커머스플랫폼 회원마케팅 서비스개발

[컬리의 코프링 & 헥사고날 아키텍처 적용 사례]

“개발자란 비즈니스 생태계를 이해하고 발견된 문제점을 가장 효율적인 기술로 해결하는 사람들이죠.

컬리에서 스터디와 몸 프로그래밍, 코드리뷰로 팀원들과 협업하여 특정 비즈니스 문제점을 포착하고 개선했던 스페셜 경험을 공유합니다.”





두번째 이야기-방신철 님

SYSTEM: 문제가 발생했습니다.



계정 찾기 인증 방식의 부족
※ 기존에는 이메일 인증 방식만 제공

→ 컬리 가입 시 사용했던 이메일 주소를 잊었을 때 계정을 찾을 방법이 없음



수치화 된 값으로 문제를 정의해보자!
※ 월 약 4,300건 가량의 고객 불편 사항이 지속적으로 접수됨

→ 컬리 서비스 내에 전체 인입되는 고객 불편사항에서 4%의 큰 비중을 차지함





두번째 이야기-방신철 님

어떻게 해결해볼까? - 목표의 수립



다양한 인증 수단의 옵션 고려

→ 가장 대중적이고 익숙한 '휴대폰 인증 방식'을 추가 제공하자!



좀 더 욕심을 내볼까?

기존의 PHP 기반의 모놀리식 애플리케이션을 통해 제공되던 계정 찾기 기능은,
다른 기능의 문제로 장애가 발생했을 경우에도 영향을 줄 수 있는 상황.

확장성, 장애 격리성을 보장할 수 있는 아키텍처의 새로운 마이크로 서비스로 개선해보자.

→ 기존의 모놀리식 애플리케이션에서 기능을 분리해보자!



두번째 이야기-방신철 님
저희는 이런 도구들을 사용합니다.



프로젝트를 진행하면서 기술적인 성장도 함께 챙기자!



기존 개발 스택



새로운 언어 채택

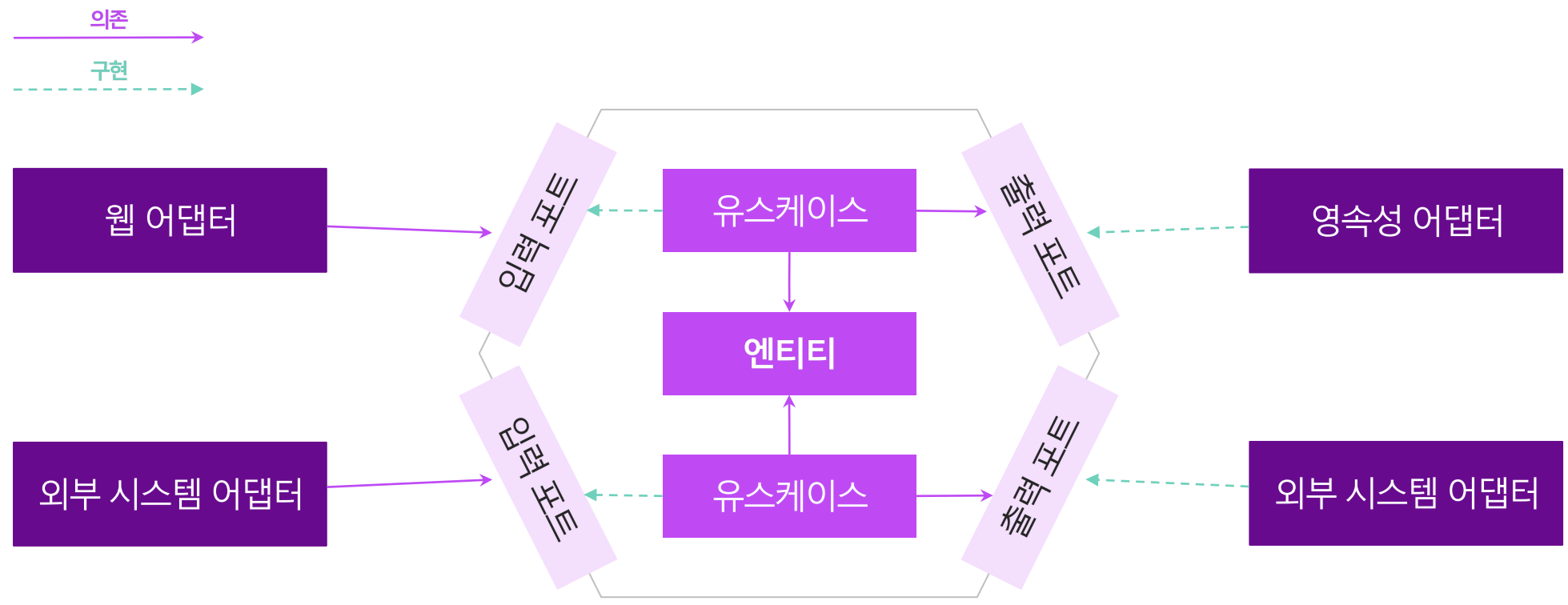


First Language로 코틀린 언어를 채택한 이유는?
조금 더 가독성 좋은 코드를 작성, 유지보수하기 좋은 클린 아키텍처로 개발해보자.
팀 내의 자바 & PHP 개발자의 효율적인 co-work을 위해 채택!





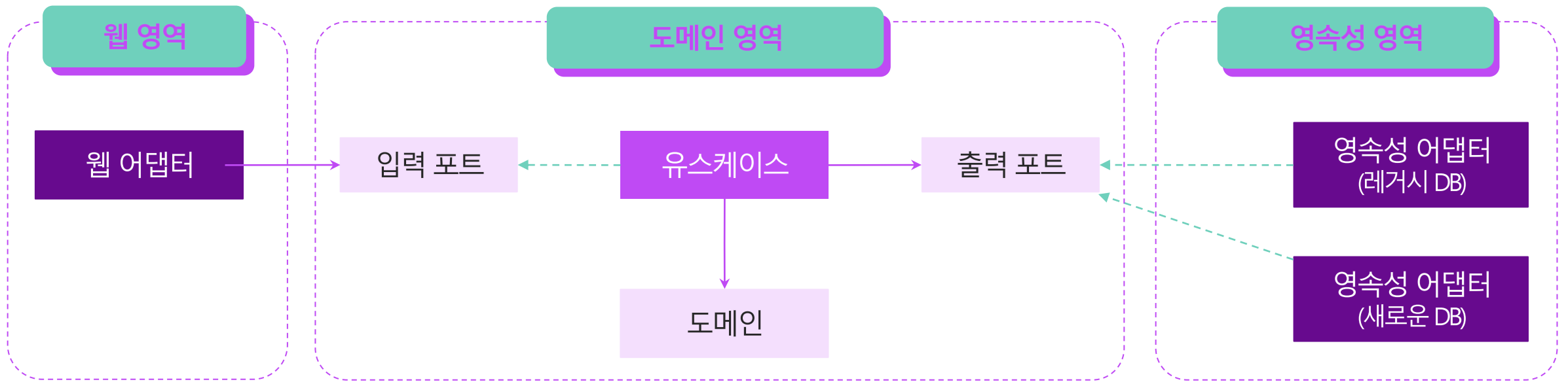
두번째 이야기-방신철 님
헥사고널 아키텍처의 도입 배경



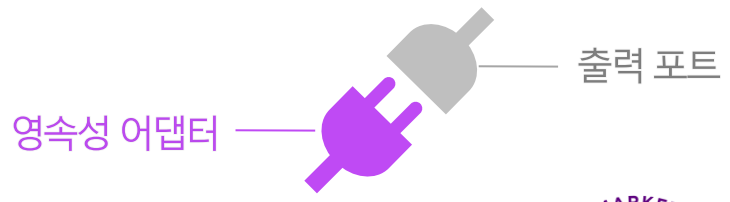


두번째 이야기-방신철 님
우리는 이렇게 적용해보았습니다!

 헥사고날 아키텍처 적용을 위한 멀티 모듈 구조 프로젝트 구성



→ 의존
- - - 구현



➤ 출력 포트 인터페이스를 통해서 영속성 어댑터를 갈아 끼우기만 하면 되도록 설계





두번째 이야기-방신철 님

과정을 통해 느낀 어려움과 값진 경험



“이런 어려움을 겪었어요!”

- 러닝 커브
- 많은 포트 및 어댑터 생성 → 구조적으로 복잡



“하지만 이런 장점을 경험했어요!”

- 유스케이스별 깔끔한 분리 → 협업이 편리해짐
- 깨진 창문 이론 방지 → 깔끔한 객체지향 개발 가능



두번째 이야기-방신철 님
우리가 만들어 낸 결과는?



그래서 해당 프로젝트에서 어떤 결과를 얻으셨나요?

수량적 결과

- ✓ 월에 약 4300건 정도 발생하던 고객 불편사항을 9~60건으로 대폭 줄임.
- ✓ 확장이 용이하고 장애 격리된 새로운 애플리케이션을 성공적으로 런칭함.

역량 성장 결과

- ✓ 팀 내에 코틀린+스프링부트 개발 스택을 잘 안착 시킬 수 있었음.
- ✓ 동료들의 기술적 역량 성장.





두번째 이야기-방신철 님
특별한 컬리만의 협업 방식

몹 프로그래밍

- 한명의드라이버와여러명의프로그래머가 하나의PC로 코딩또는 문서화작업을 진행하는 방식으로 1:1 방식인페어 프로그래밍을 1N으로확장시킨 형태의 개발방식
- Covid19로 인한재택근무방식에따라, IntelliJ IDEA의 Code With Me 활용하여 온라인방식의몹 프로그래밍 진행



코드리뷰

- 서로의코드에대해자유롭게논의
- 단,코드리뷰를통해PRApprove가되어야 코드베이스에머지할수있도록규칙을생성
 - *나의동료가어떤,왜,어떻게작업했는가파악
 - *동료가보지못한포인트더블체크하여버그를 최소화,클린코드로개선피드백
 - *동료의작업물을통해인사이트개발지식을얻도록



참여하는 모두가 오너십을 가지고 임하며 서로 어렵거나 모르는 부분에 대해 부끄러워하지 않고 공유할 수 있는 분위기로, 함께 스터디하면서 즐겁게 진행함. 기술적 커뮤니케이션을 통해 기술적 지식과 경험의 전달이 빠르고 효과적으로 이루어 짐





두번째 이야기_방신철님 건강한 코드리뷰 문화

PR 포맷

BangShinChul commented on 9 Dec 2021 · edited

개요 📄

카카오 API 서버로 카카오톡 채널 관례를 조회하는 Port 및 Adapter를 작성하였습니다. 추후에 해당 Port 및 Adapter를 사용하여 카카오톡 연동/회원가입 할 때 카카오톡 채널 관계 확인 API 를 호출해서 정보를 DB에 저장 하는 기능을 작성할 예정입니다.

작업 사항 📋

- 카카오 채널 관계 enum 클래스를 작성하였습니다.
- 카카오톡 채널 관계 조회 Port 및 Adapter를 작성하였습니다.
- 카카오톡 채널 관계 조회 Adapter의 테스트 코드를 작성하였습니다.

참고 문서 📖

- 카카오톡 채널 관계 확인하기 API 문서 링크
- Spring Cloud Config : 카카오톡 채널 uuid 프로퍼티 참고

체크 리스트 ✅

PR을 발행하기 이전에 아래의 항목을 확인해 주세요.

- 작성된 코드에 대해 테스트 코드를 적절히 추가, 변경, 삭제 하였습니까?
 - Test Results
 - com.kurly.member.auth.adapter.social.kakao.FindKakaoChanneRelationAdapterTest **797 ms**
 - find 메서드는
 - 정상적인 카카오톡 채널 관계 정보를 응답받을 경우 382 ms
 - 마켓컬리 채널의 관계 정보를 리턴한다. 313 ms
 - 빈 카카오톡 채널 관계 정보를 응답받을 경우 79 ms
 - null을 리턴한다. 9 ms
 - 기준에 정의되지 않은 카카오톡 채널 관계 정보를 응답받을 경우 3 ms
 - 예외를 리턴한다. 9 ms
 - 예외를 리턴한다. 2 ms
- 모든 테스트 코드가 성공 하였습니까? (명령어: ./gradlew test)

실제 코드리뷰내용

```
common/src/main/kotlin/com/kurly/member/auth/exception/ErrorCode.kt
37 + * @param HttpStatusCode Http Status Code
38 + * @return 공통 오류 코드 객체 code 값
39 + */
40 + fun getCode(httpStatusCode: Int): String {
```

binary-park on 3 Sep 2021
@if 메서드 명을 사용하면 어떨까요?

BangShinChul on 3 Sep 2021
@binary-park 리뷰 감사합니다 환영합니다

binary-park on 3 Sep 2021 · edited
@if 메서드 명이 더 가독성이 좋을까요..?

binary-park on 3 Sep 2021 · edited
조수아 일이 이메티비 자바에서 알면지던 메서드 메서드 명을 소개 할때 of, from, getInstance 등을 소개 해주고 있었어요 **

전 주로 아래와 같이 구분해서 사용 합니다. 주입된 인자가 primitive 타입이면 of를 주입된 인자가 객체라면 from을 사용합니다. 싱글 톤이라면 getInstance

BangShinChul on 3 Sep 2021
맞 ㅋㅋㅋ 조수아형. 감사합니다 환영합니다 개선해서 다시 전달드리겠습니다!

BangShinChul on 6 Sep 2021
@binary-park 기존 @getCode 메서드를 @if 메서드로 명칭을 수정하였습니다.

```
+ companion object {
+ /**
+ * Http Status Code에 달맞은 공통 오류 코드 객체 code 값을 리턴.
+ *
+ * @param HttpStatusCode Http Status Code
+ * @return 공통 오류 코드 객체 code 값
+ */
+ fun of(httpStatusCode: Int): String {
+     return when (httpStatusCode) {
+         400 -> BAD_REQUEST.code
+         401 -> UNAUTHORIZED.code
+         403 -> FORBIDDEN.code
+         404 -> ENTITY_NOT_FOUND.code
+         406 -> NOT_ACCEPTABLE.code
+         415 -> UNSUPPORTED_MEDIA_TYPE.code
+         422 -> INVALID_INPUT_VALUE.code
+         500 -> INTERNAL_SERVER_ERROR.code
+         504 -> GATEWAY_TIMEOUT.code
+         else -> INTERNAL_SERVER_ERROR.code
+     }
+ }
```

binary-park on 9 Sep 2021
로그에 대한 궁금증...

- 해당 로그가 정말 필요한지?
- 해당 로그의 성격이 info 인지 debug 인지
- 해당 로그가 보안정책에 위배되지 않는지... (개인정보를 로그에 찍으면 안됨, 마스킹 여부에 따라 다르지는 않으셨는데요)

binary-park on 9 Sep 2021 · edited
자 이제 코멘트에 익숙해 지신거 같으니 조금 더 가독성 좋은 코드를 작성해 봅시다-

아래의 구조는 find 메서드에서 의사코드 느낌으로 수행되는 절차를 메서드로 묶어서 이렇게 코드를 하게되면 절차가 명확하게 보이는 장점이 있습니다. 아래의 코드로 예를 들

- 취급 목적을 조회하고 있다면 오류를 발생한다.
- 첫 번째 회원은 기준으로 포문을 생성한다.
- 이메일을 발송한다.

저렴 요구사항이 명확하게 보이고, 해당 절차의 상세 구현을 메서드 내부에 정의되게 된 코드를 안부 로지더라도 다른 분명이 해당 유스케이스를 쉽게 이해할 수 있겠죠? 상세 메서드 단위로 명명이 구분되어 있기 때문에 이메일 발송이 잘 안된다면 @sendEmail 메 나중에 해당 클래스의 역할/책임이 커진다면 메서드가 클래스로 도출할 수도 있겠죠?

BangShinChul on 9 Sep 2021 · edited
@binary-park 환영합니다 피드백 주셨던 대로, 가독성 좋은 코드를 위해 개선작업을 해보았습니다! #58 (comment)

그리고 일단 로그는 제거하였습니다

앞으로는 이런 방법을 우선적으로 고려해서 개발하도록 하겠습니다 🙏

binary-park on 9 Sep 2021 · edited
find 메서드에서 구현이 되는 절차마다 공개 위임으로 구분해 주시면 더 좋을까요

BangShinChul on 9 Sep 2021
@binary-park 리뷰 감사합니다 환영합니다 피드백 주신대로 공개 위임으로 구분을 해보았습니다 **

```
override fun find(command: FindPasswordByEmailCommand) {
+     val members = findMembersIfEmptyThrowsException(command)
+     val (member, token) = createTokenByFirstMember(members)
+     sendEmail(command, member, token)
+ }
```

BangShinChul on 9 Sep 2021

@binary-park ... 정말 많이 배웁니다..! 🙏 감사합니다 환영님! 피드백 주신대로 개선해보도록 하겠습니다!

아, 그리고 로그에 대해서는 제가 생각이 짧았던 것 같습니다. 일단 비밀번호 재설정 링크 이메일 발송 의 경우, 하나의 계정에만 전송해야 하는데 즉, 회원 정보 조회 시 memberId 라는 유니크 값으로 조회하는데 만약 여러 회원 정보가 리턴된다면, 왠지 파악하고 있어야 할 것 같아서 로그를 찍어두면 좋겠다 싶어서 작성했던 코드였습니다.

이 부분에 대해서는 추후에 저회끼리 내부적으로 논의한 후에(로그를 찍어야 하는지, 로그를 찍어야 한다면 어떤 구조로 출력해야 하는지 등) 작성하는것이 좋을 듯 싶습니다! 따라서 해당 로그 출력 코드는 제거하도록 하겠습니다!





두번째 이야기-방신철 님
저희는 이런 동료의 합류를 기다려요!



✓ 본인이 무엇을 모르는지 알고,
이를 학습해서
기술적 성장을 하시는 분



✓ 모르는 것을 부끄러워
하기 보다는 더 드러내어
성장하기 위해 노력하시는 분



✓ 학습한 기술이나 경험을
동료에서 전파하여
기술적 상향 평준화를
이루고자 하시는 분





컬리 개발자성장기

세번째 이야기의 주인공

저는 “양질의 코드와 지속 가능한 프로젝트를 고민하고
비즈니스를 코딩과 같은 열정으로 이해하고자 노력하는 평범한 개발자”

최준영

입니다.

> 최준영님 | 플필먼트프로덕트 출고서비스개발

[더 나은 출고 서비스를 위한 향해 일지]

“컬리에서 안정적인 출고 서비스를 제공하기 위해 겪었던
흥미로운 향해 일지와 앞으로의 방향성을 공개합니다!

더불어 팀원들과 어떤 방법으로 논의하며 함께 성장할 수 있었는지,
저희만의 커뮤니케이션 문화에 대해서도 소개합니다.”

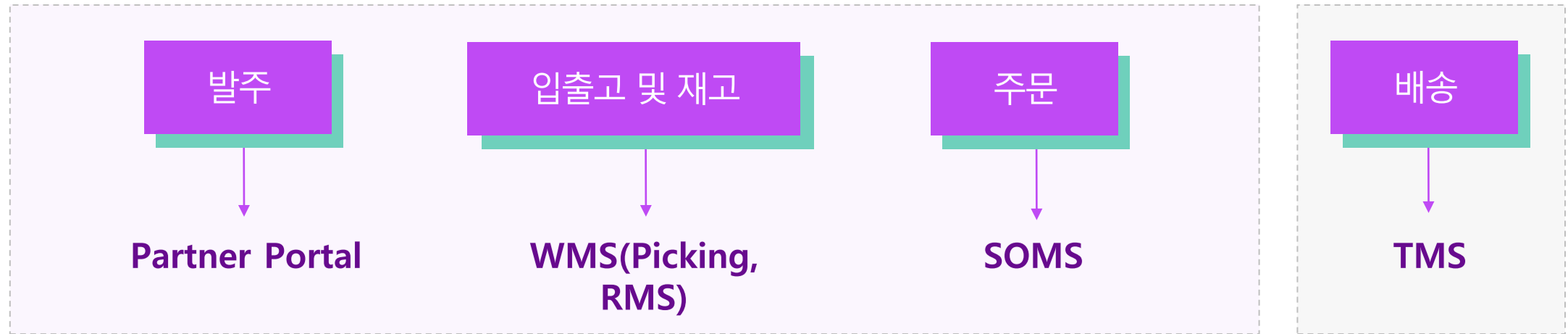




세번째 이야기-최준영 님
풀필먼트 소개

컬리 풀필먼트

컬리 딜리버리





세번째 이야기-최준영 님
풀필먼트 소개



피킹(Picking)이란?



피킹 (picking)

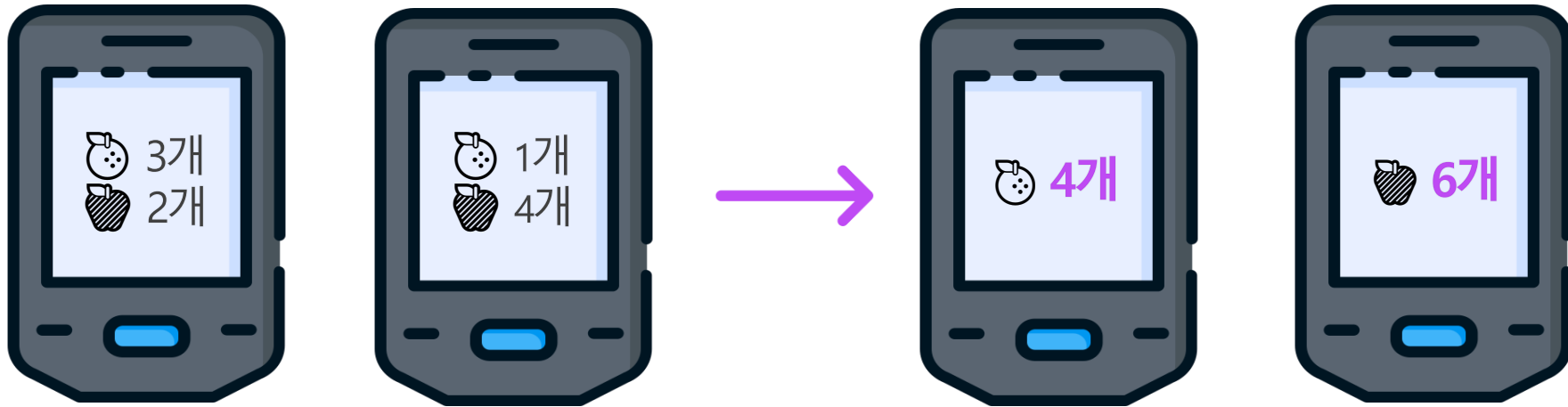
[명사] [경제] 출고할 상품을 물류 창고의 보관 장소에서 꺼내는 일



세번째 이야기-최준영 님
풀필먼트 소개

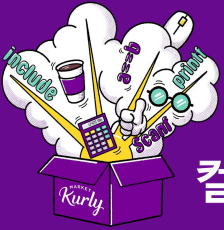


최적화란?



최적화

》 주문을 상품으로 그룹화하여, 작업자가 최적의 동선으로 더 빠른 피킹을 가능하게 데이터를 가공하는 방법



세번째 이야기-최준영 님
컬리 풀필먼트(물류) 센터

송파, 화도 물류센터



DAS

*DAS(Digital Assorting System):
회차별 각 오더유형별 주문 품목을 한번에 피킹(총괄 피킹)하고 피킹된 상품은 DAS존에 비치된 빈 바구니에 각 고객별 주문상품을 재분류하는 시스템

김포 물류센터




QPS, DAS

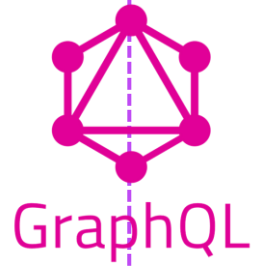
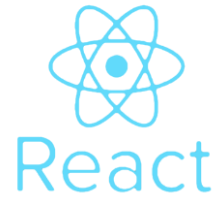
*QPS(Quick Picking System):
상품 분리 담당자가 레일을 통해 자신 앞으로 이동해 온 상품을 시스템 지시에 따라 상자에 담고 바로 이어 포장 단계로 넘길 수 있도록 설계 되어 생산성 향상 및 작업 환경을 개선한 자동화 시스템





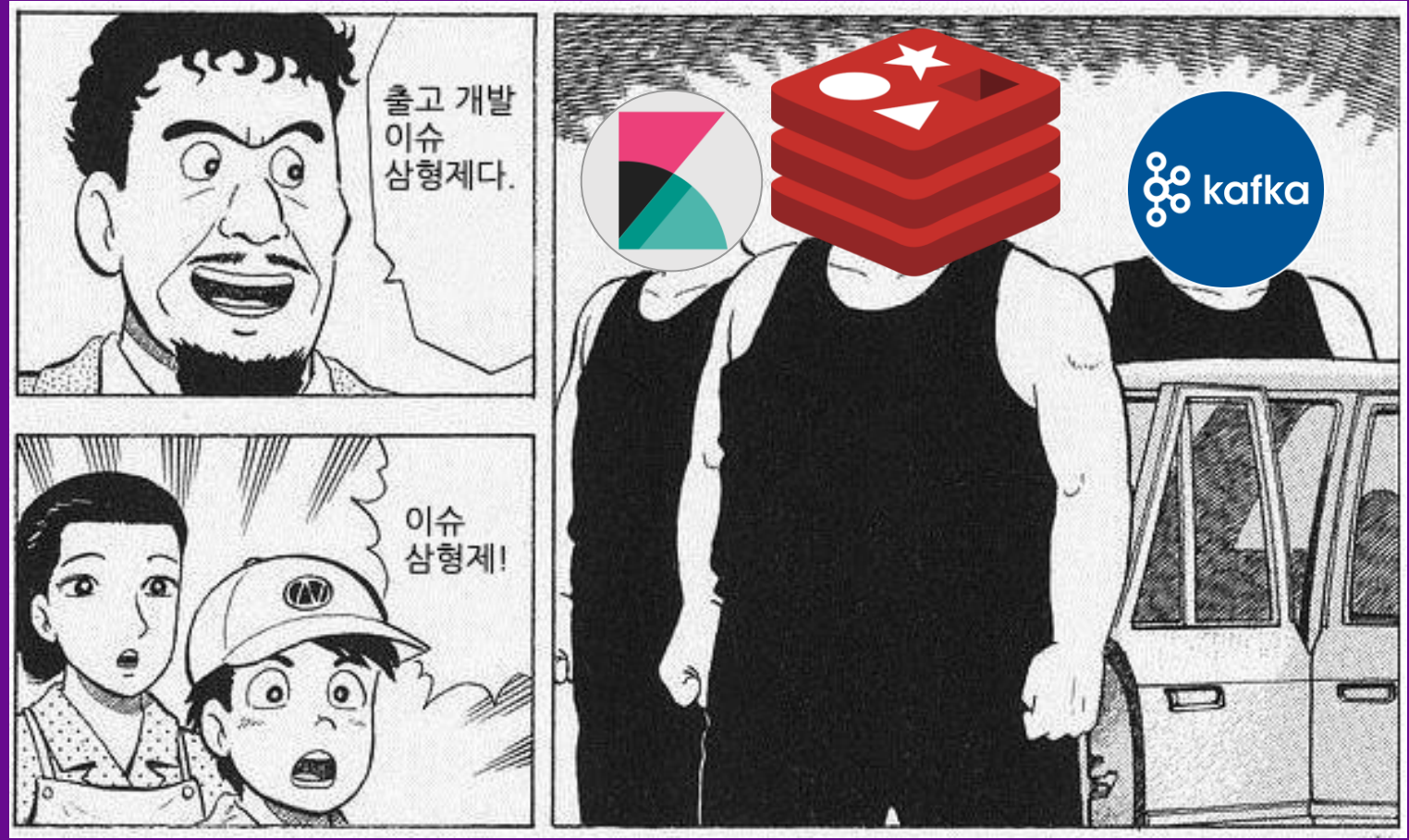
세번째 이야기-최준영 님
컬리 김포CC PDA Picking 기술 스택

 김포 PDA Picking에 사용된 모든 기술 스택을 소개합니다!



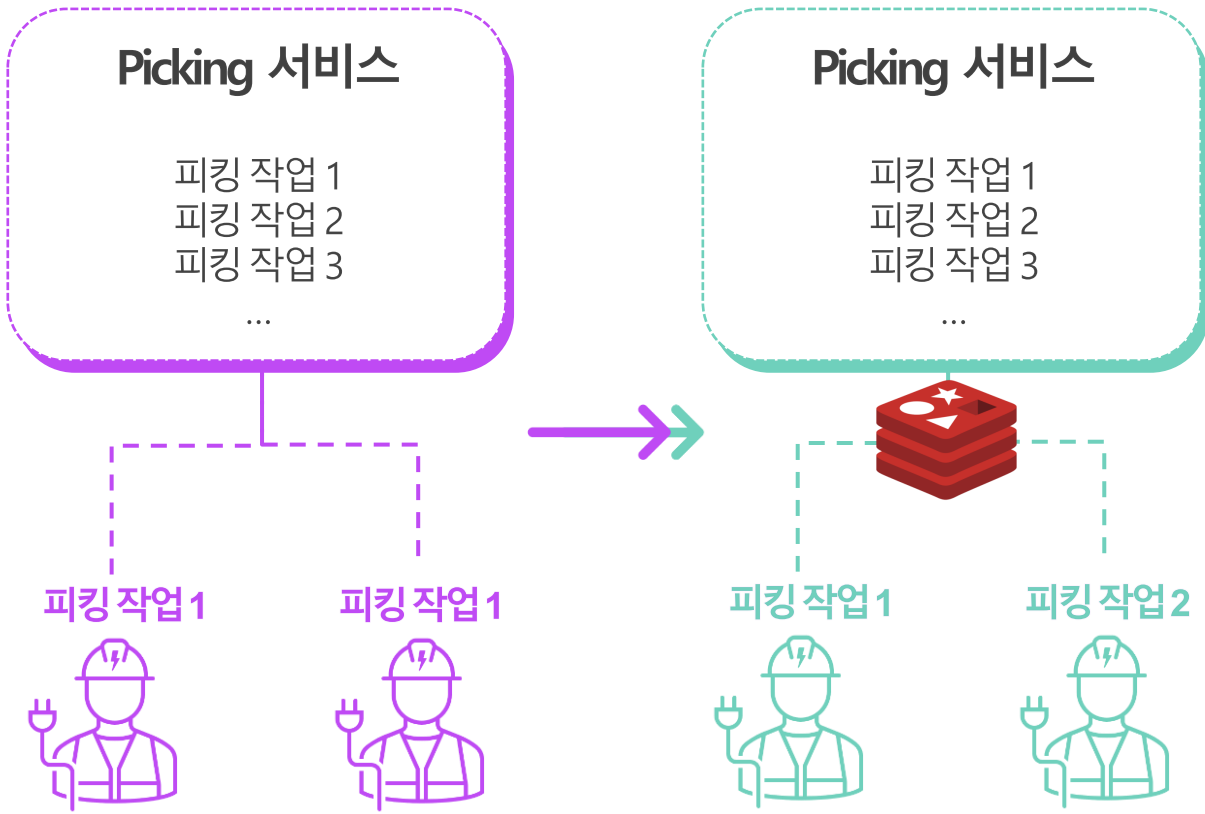


세번째 이야기-최준영 님
개발 이슈에 대해 소개합니다





세번째 이야기-최준영 님
이슈1 (동시성 제어, Redis Distributed Lock)



```
protected void initRedisDistributedLock() {
    this.redisClient = SpringUtils.getBean( name: "redisson");
    this.lockKey = lockKey + this.distributedKey;
    this.redisDistributedLock();
}

protected void redisDistributedLock() {
    log.info("redisDistributedLock !! {}", this.lockKey);
    this.rLock = redissonClient.getLock(this.lockKey);
    this.rLock.lock( leaseTime: 10, TimeUnit.SECONDS);
}

protected void redisDistributedUnLock() {
    log.info("redisDistributedUnLock !! {}", this.lockKey);
    this.rLock.unlock();
}
```

```
> keys PDA-CC02:picking_batch_last_
qps:*
1) "PDA-CC02:picking_batch_last_qps:GGM1"
2) "PDA-CC02:picking_batch_last_qps:GGH1"
```





세번째 이야기-최준영 님 이슈2 (로그 확인, Kibana)

The screenshot shows the Datadog interface for a specific trace. At the top, it identifies the trace as 'logistics-outbound-picking-cc02' with a POST request to '/graphql'. A flame graph below shows the execution timeline with a highlighted span of 4.49 ms. The log viewer at the bottom shows the following log entries:

```

Feb 10 19:06:04.965 logistics-outbound-picking-cc02 | logistics-outbound-picking-cc02-AS-152-9
===== USER : GP043 REQUEST : com.kurly.ipd.outboundpicking.core.gimpo.adapter.in.PickingTaskMutationResolver(mark
toteId=T100016079, picker=GP043, warehouseKey=GM01, documentDate=2022-02-10, requestShippingDate=2022-02-11, processType=MULTI
0079, skuCode=MK0000097524, skuAlternativeCode=8801115135104, skuName=[서울우유] 말기 우유 200ml 3개입, skuSubstituteName=[유통] [서울
Feb 10 19:06:04.967 logistics-outbound-picking-cc02 | logistics-outbound-picking-cc02-AS-152-9
==> Preparing: select /*+ index(a PICKING_MASTER_IDX_08) */ a.PICKING_MASTER_ID AS pickingMasterId ,a.WAREHOUSE_KEY AS wareh
,a.PROCESS_SUB_TYPE AS processSubType ,a.GROUP_NO AS groupNumber ,a.WORK_BATCH_NO AS workBatchNumber ,a.PICKING_BATCH_NO AS p
,a.SKU_SUB_NAME AS skuSubstituteName ,a.PICKING_ZONE_ID AS pickingZoneId ,a.LOCATION_KEY AS locationKey ,a.LOCATION_NAME AS l
Feb 10 19:06:04.968 logistics-outbound-picking-cc02 | logistics-outbound-picking-cc02-AS-152-9
==> Parameters: GM-092381-002-0079(String), GM-092381-002(String), GM-092381(String)
Feb 10 19:06:04.972 logistics-outbound-picking-cc02 | logistics-outbound-picking-cc02-AS-152-9

```

The screenshot shows Kibana search results for the query 'pickingBatchNumber: 'GL-091883-001-0012''. It displays 49 hits. A bar chart shows the distribution of hits over time. Below the chart is a table of log entries with columns for Time, picker, actionType, and keyCode.

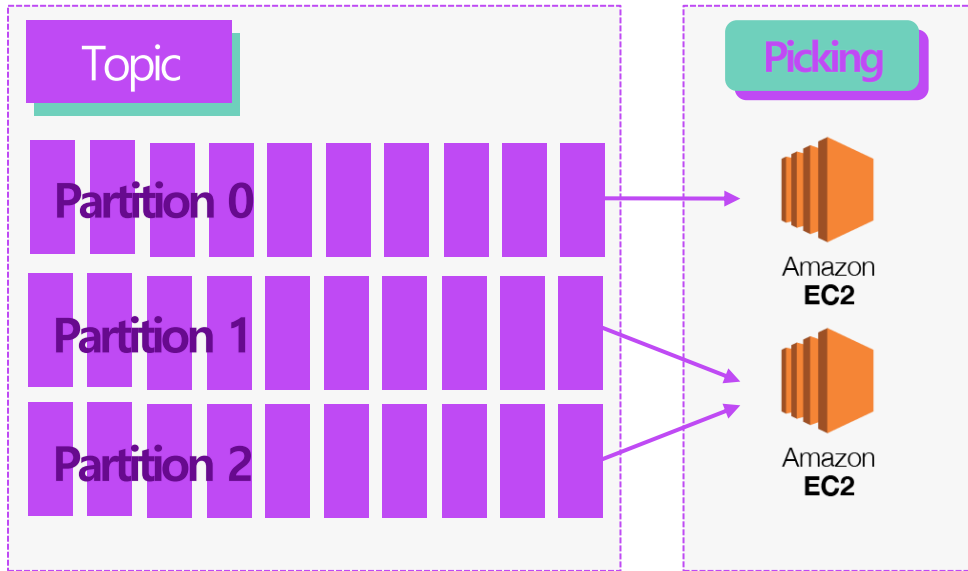
Time	picker	actionType	keyCode
> Feb 10, 2022 @ 19:25:41.645	GP640	BATCH_COMPLETE	GL-092385-001-0014
> Feb 10, 2022 @ 19:25:41.510	GP640	SKU_SCAN	8809660652350
> Feb 10, 2022 @ 19:25:41.048	GP640	SKU_SCAN	8809660652350
> Feb 10, 2022 @ 19:25:33.264	GP640	LOCATION_SCAN	13SC0209103
> Feb 10, 2022 @ 19:24:56.353	GP640	SKU_SCAN	8801114149355
> Feb 10, 2022 @ 19:24:51.822	GP640	LOCATION_SCAN	13SC0209101
> Feb 10, 2022 @ 19:24:20.735	GP640	SKU_SCAN	8809663130664
> Feb 10, 2022 @ 19:24:20.067	GP640	SKU_SCAN	8809663130664
> Feb 10, 2022 @ 19:24:08.256	GP640	LOCATION_SCAN	13SC0101203
> Feb 10, 2022 @ 19:23:44.829	GP640	SKU_SCAN	8809536620230



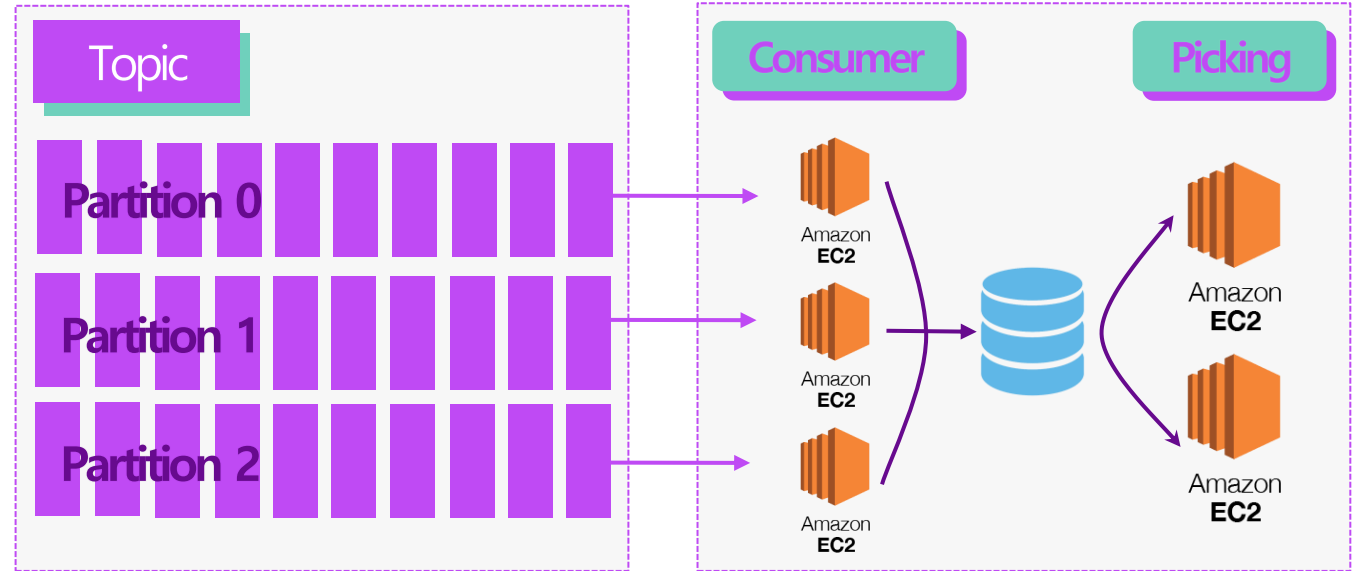


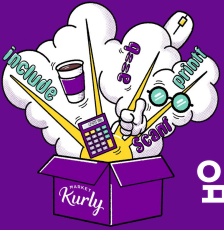
세번째 이야기-최준영 님
이슈3 (컨슈머 분리, Kafka Consumer)

As-Is



To-Be





세번째 이야기-최준영 님
앞으로의 방향성



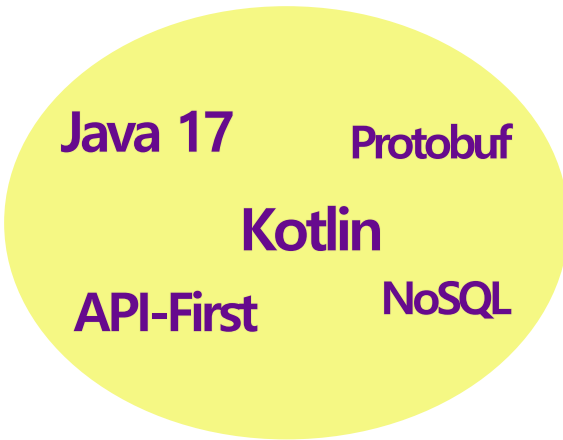
출처 : <https://www.youtube.com/watch?v=YtztXnXeAb4>

김포 DAS 도입

*DAS(Digital Assorting System):
회사별 각 오더유형별 주문 품목을 한번에 피킹(총괄 피킹)하고 피킹된 상품은 DAS존에 비치된 빈 바구니에 각 고객별 주문상품을 재분류하는 시스템



신규 센터 오픈



신규 기술 도입





세번째 이야기-최준영 님
팀 문화를 소개합니다.

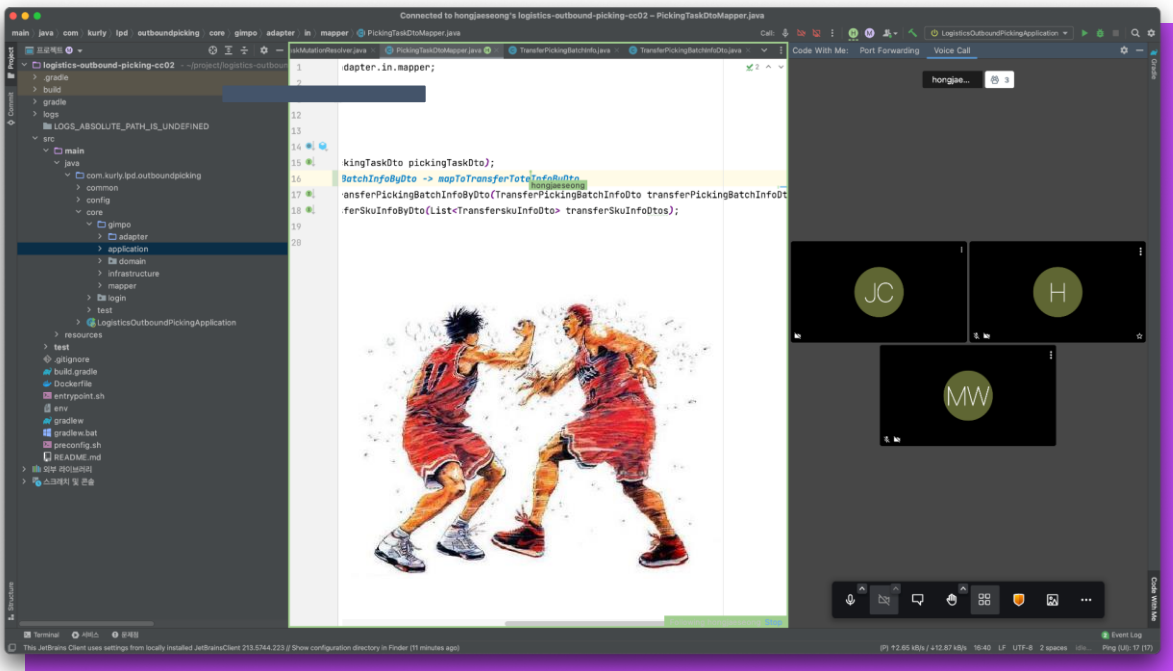




세번째 이야기-최준영 님
페어 프로그래밍과 Code With Me를 소개합니다.



저희 팀의 특별한 개발 문화를 소개합니다.



페어 프로그래밍

Code With Me

- ✓ 서로의 지식을 공유하며 개발
- ✓ 같은 목표에 집중하여 더 빠른 결정
- ✓ 실시간 코드 리뷰 가능
- ✓ 드라이버는 작은 목표, 네비게이터는 더 넓은 시야





세번째 이야기-최준영 님
저희 팀은 문서에 진심입니다.

저희 팀의 특별한 개발 문화를 소개합니다.

진지 문서화

기본 전략

앞서 말했듯이 모든 배포는 QA 검수가 이뤄져야 배포가 가능하게끔 구조 변경을 해야함.
그러므로 티켓이 발생될 때마다 개발자는 feature branch를 생성하고, 개발 완료 시 QA 업무 요청을 진행하여 QA branch를 생성한다. 만약 개발 병위가 비슷하지만 티켓이 나뉘어있다면 QA와 조율하여 한 QA branch에 여러 feature branch를 병합하여 QA를 진행한다.

master: 현재 운영 Product에 배포된 내용과 동일한 내용의 브랜치
stage: 차기 운영 Product에 배포되기 전으로, 담당 내일 배포가 되어도 문제가 없는 브랜치
qa: QA 검수를 받기 위한 브랜치
feature: 신규 개발을 위한 브랜치

1. Jira에 티켓이 생성되면, 해당 내용에 대한 feature branch를 생성한다
2. 개발 완료 후 QA 알람이 확정되면 QA branch를 생성하고 소스를 병합한다
3. QA 검수가 완료될 동안, 개발자들은 QA branch에서 각자 개발할 소스를 병합한다
4. 만약 동시에 다른 QA가 진행될 경우, 새로운 QA branch를 생성하고 2~3번과 동일한 형태로 진행한다
5. QA가 완료될 경우 stage branch에 소스를 병합한다
6. stage branch 브랜치에서 master branch 브랜치로 소스를 병합하고 이미지를 생성하여 운영 Product에 배포한다

개발자간 지켜야 할 규칙 - 이름

이름이 branch 규칙은 어느정도 정리가 되지만, 협업하기엔 아직 못가지 정리하고 넘어가야 할 내용들이 있다. 대표적으로 branch 명칭 규칙, PR 규칙에 대해 다뤄보고자 한다.

1. branch명

branch type과 Jira 티켓명으로 branch명을 생성하되, 해당 branch에 대한 내용을 확인하려면 Jira에 접속하여 해당 티켓 번호를 입력하고 조회해야 했다. 그래서 간단한 요약을 작성하여 한눈에 branch 내용을 파악하기 용이하게 작성하기로 했다.

사내 협업 툴(컨플루언스) 활용

Outbound-스쿼드 B

MARKET Kurly
출고서비스개발 Squad B

무슨 일을 하나요?

저희 팀은 마켓컬리에서 발생하는 유통의 송고 서비스 중 유통 피킹 시스템과 신규 WMS 고도화를 담당합니다. 고객님께서 주문하신 물건을 최대한 빠르고 정확하게 피킹할 수 있게 지원하고, 현 WMS 시스템의 3.0 버전을 위한 신규 기술 도입을 연구하고 있습니다. 구체적으로는 다음과 같은 일을 합니다.

- 하루 수십건의 주문에서 더 빠르게 물품을 담을 수 있도록 최적화된 데이터 원장 작업자가 PDAR 작업할 수 있도록 개발하고 운영합니다.
- 현장 관리자가 관리, 제어, 통계 모니터링 할 수 있는 어드민 시스템을 개발하여 사용합니다.
- 김포 QPS 운영을 담당하는 WCS와 kafka로 연계하여 피킹 정보를 송수신합니다.
- 팀원들이 신규 WMS 시스템에 필요한 기술들을 각자 유기적으로 학습시키고 스태디하여 다가올 개발에 대비합니다.
- 개발자들은 두명이 작업을 이루는 페어 프로그래밍을 추구하여 더욱 견고한 시스템을 완성할 수 있도록 노력합니다.
- 모든 코드는 코드리뷰를 진행하고, 코드리뷰가 끝나면 QA를 시작합니다.
- 프로젝트가 시작되면 기획/운영/개발이 한팀으로 움직이기로 했습니다.
- 팀원 전체가 문서의 중요성을 알고 대부분의 내용을 문서화하여 기록합니다.

진행 사항

- 김지미김주희
- 김포 PDA 고도화 논의 시작

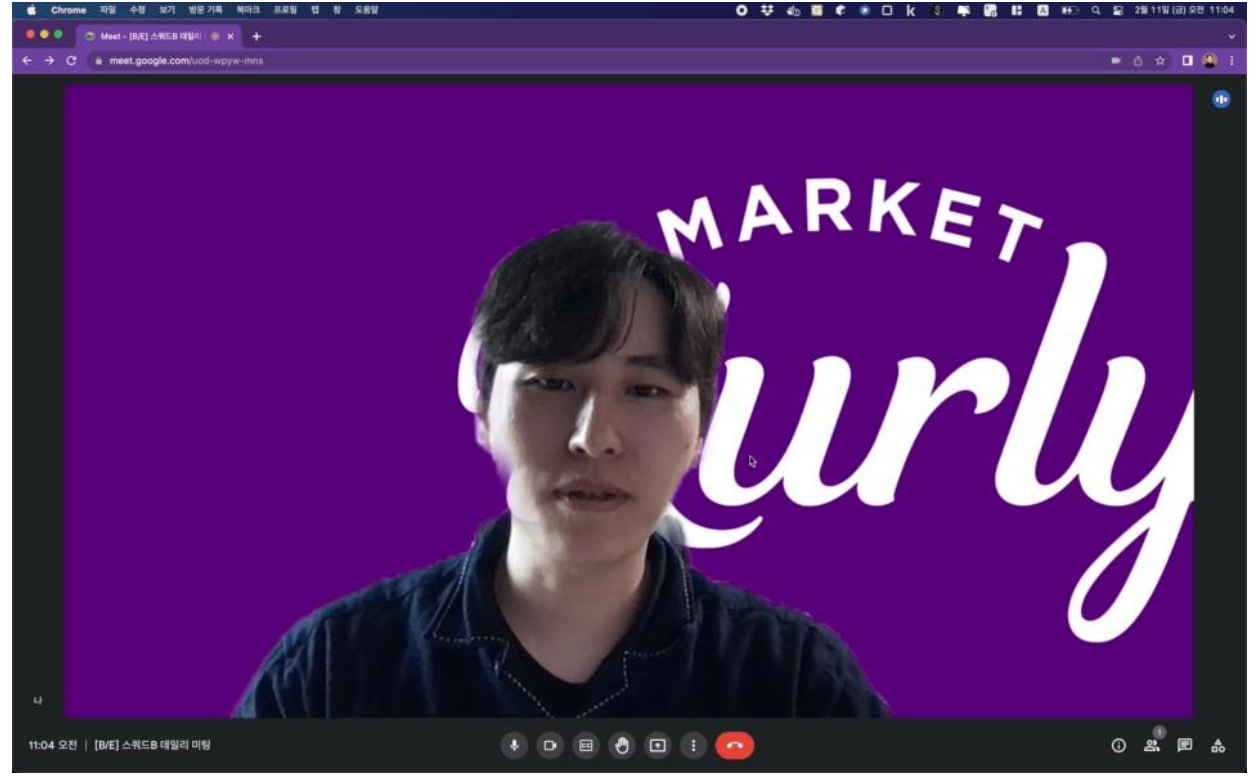




세번째 이야기-최준영 님
'버스터 콜' 들어보셨나요?



저희 팀의 특별한 개발 문화를 소개합니다.



※보안정보 및 개인정보 입력 금지※

2021.11.10 호출자 [오대호]

LOD우민식(아가) 만들
LOD오대호(이카) 11월 11, 2021에 마지막 업데이트 · 12 페이지를 조회한 사람 11명

2021.11.10 버스터콜

@LOD오대호 님의 호출

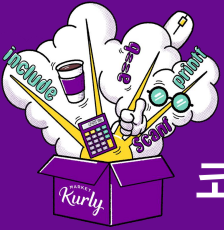
참석자

- @LOD홍재성
- @LOD최준영**
- @LOD최우림
- @LOD우민식

주제 : 김포 리팩토링 스팩을 어떻게 할까요?

결론

1. 변경점이 있을경우 본인의 스코프 내에서 변경을 한다.
 - a. 전체 코드 레벨에서 변경이 필요한부분은 TODO로 코멘트를 남겨두고 코드리뷰시 팀원이 해당 TODO에 따봉을 달아서 인지하도록한다.
 - b. 현재 일정내 가능하면 진행하되, 불가하다면 Todo로 남겨둔다.
2. mybatis -> jdbcTemplate 전환하지 않는다. 통계, 배치 쿼리 외 비즈니스 로직은 가능한 jpa 로 이관한다.
 - a. 추가피킹 지시
3. 추후 개선 과제는 TODO 에 남기도록 한다. [관리자 API 서비스 분리](#)
4. 유효성 검사는 에러포맷 일관화와 추후 graphql deprecated 를 고려하여 graphql 스키마 에러를 활용하지 않고 spring, java code 레벨에서 처리한다.



세번째 이야기 - 최준영 님
코드리뷰도 물론 하고 있습니다.

저희 팀의 특별한 개발 문화를 소개합니다.

```

Feature/lot 436 picker list api #69
MinSiK-Woo merged 4 commits into thefarmersfront:stage from kurly:feature/LOT-436-picker on 23 Dec 2021

daeho-oh0 reviewed on 23 Dec 2021
src/main/java/com/kurly/lpd/outboundpicking/core/gmpo/application/UserService
e.java
31 + pickerGrantDtoList.add(PickerGrantDto.builder().userId(a.getUserId()).isNew(
32 + Collections.emptyList().isEmpty().getPickerGrantAuthorizationEntities());
33 + Collections.emptyList().isEmpty().getPickerGrantAuthorizationEntities());
34 + });

daeho-oh0 on 23 Dec 2021
ListsPickerGrantDto pickerGrantDtoList = userMasterEntityList.stream().collect(Collectors.toList());
Collections.emptyList().isEmpty().getPickerGrantAuthorizationEntities(); 이 같은 문제를 꼭 살펴보는 것이 중요해요? 둘 다 별 차이가 없을 것 같긴 하지만 별
도의 목적 없이 적혀 있다면 이는 경우 지는 스텝들을 더 자주 쓰는 것 같아요!

MinSiK-Woo on 23 Dec 2021
... 코틀린인데 forEach를 자주 써서 습관적으로 그런거 같아요...; 일단 둘의 차이점에서는 좀더 알아보도록 하겠습니다.

daeho-oh0 reviewed on 23 Dec 2021
src/main/java/com/kurly/lpd/outboundpicking/core/gmpo/application/UserService
e.java
34 + });
35 + pickerGrantDtoList.sort(Comparator.comparing(PickerGrantDto::getUserId));
37 + pickerGrantDtoList.sort(Comparator.comparing(PickerGrantDto::isNew).reversed());

daeho-oh0 on 23 Dec 2021
PickerGrantDto 객체로 맵핑 후 userId 와 isNew 값으로 오름차순 정렬하는 코드로 보입니다.
위에서 스텝들을 사용 한다면 map() 후 sorted() 해서 정렬하면 코드도 깔끔할 것 같습니다. (스텝들을 안 써도 list 정렬하
면서 제이닝 해서 처리할 수 있네요.)
찾아보니 여러개의 조건으로 정렬할 때 제이닝에서 처리하는 방법들이 많네요.
반영하시다면 참고하시면 좋을 것 같습니다.

MinSiK-Woo on 23 Dec 2021
감사합니다
    
```

```

LOUT-415 피킹존 역설 로드 개선 #50
woorim-choi merged 9 commits into thefarmersfront:stage from woorim-choi:feature/LOT-415-loc on 14 Dec 2021

daeho-oh0 on 10 Dec 2021
Collections.emptyList().isEmpty().getPickerGrantAuthorizationEntities(); 와 같은 코드로 될일 수 있을까요?

woorim-choi on 10 Dec 2021
Collection이 Null safe한 empty 메소드가 있었으면 반영하겠습니다.

daeho-oh0 approved these changes on 10 Dec 2021
daeho-oh0 left a comment
src/main/java/com/kurly/lpd/outboundpicking/common/excel/up/LoadPickingZoneMasterExcel.java
37 -37,8 ~39,8 @@
38 |         throw new IllegalArgumentException("유호하지 않은 센터입니다.");
39 |     }
40 | }
42 + HashMap<String, Integer> locationMap = new HashMap<>();

daeho-oh0 on 10 Dec 2021
Map<String, Integer> locationMap = new HashMap<>(); 와 같이 HashMap 대신 Map 으로 받는데 더 좋지 않을까 싶습
니다.

woorim-choi on 10 Dec 2021
리터럴로 넘기거나 반환하는 객체가 아니어서 Map Interface로 선언하는것은 의미가 없다고 생각합니다.

MinSiK-Woo approved these changes on 13 Dec 2021
MinSiK-Woo left a comment
LGTM

woorim-choi merged commit cc2a511 into thefarmersfront:stage on 14 Dec 2021
    
```

Feature → QA, stage

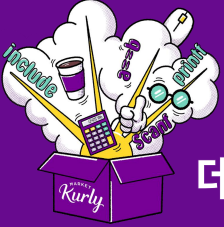


stage → master



must





세번째 이야기-최준영 님
데일리 스크럼, 장애 회고 문화도 있습니다.



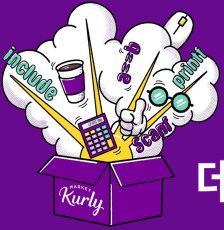
저희 팀의 특별한 개발 문화를 소개합니다.



데일리 스크럼



장애 회고



세번째 이야기-최준영 님
데일리 스크럼, 장애 회고 문화도 있습니다.



실제 팀원들이 남긴 회고록 - 김포 Picking 서비스 도입을 하며 느낀 점 - 을 공개합니다.



오대호 님

회사의 빠른 성장속도에 따라 다양한 기술적 이슈들을 마주하였고,
이를 해결해 나가는 과정에서 다양한 경험과 성장을 이룰 수 있었다.
최고의 복지는 좋은 동료라는 말을 실감했다.
물심양면으로 지원을 아끼지 않는 동료들과 함께 하지 않았다면 지금과 같이 잘 해낼 수 없었을 것이라 생각한다.
코로나가 종식되면 오프라인에서 더 많은 이야기를 나누고 싶다.

혼자 하는 것보다 여럿이서 서로 소통하며 의견교환을 하는 것이
프로젝트를 견고하게 만드는 가장 좋은 방법이라는 걸 느꼈다.
리팩토링을 진행하며 이전에 작성했던 소스를 객관적으로 평가하고 좀 더 좋게 개선하며 얻는 것이 많았다.
새로운 개발 방식을 적용 시키는데 있어 어려움이 많았다.
이후 개발 시에는 어떻게 하면 좀 더 실무에 맞는 방식이 될지 생각하면서 개발을 해야겠다.



홍재성 님





세번째 이야기-최준영 님
컬리 합류 독려 메시지



저희 팀에 합류하신다면, 아래의 경험을 얻으실 수 있어요!



수평적인 문화, 누구든 개발에 의견을 낼 수 있고 의견을 낸 내용이 실제로 반영될 수 있어요.

앞서 설명 드린 개발 문화, 기술 스택을 실제로 함께 경험해 볼 수 있어요.

빠르게 성장하는 도중에 많은 변화와 시스템 개선을 진행하는 것을 경험할 수 있어요.

이미 고도화 되어 있는 시스템이 아니기에, 많은 기술들을 고민해보고 도입해 볼 수 있어요.

다양한 버전의 프로젝트를 직접 개발해 볼 수 있어요.

'신규 개발'에 정말 진심입니다!





컬리 개발자성장기

네번째 이야기의 주인공

저는

“남편 같은 개발자”

김성준 입니다.



> 김성준님 | 딜리버리프로덕트 플랫폼개발

[지식 공유, 서비스 라이브러리를 통해
폭풍 성장 가능한 컬리 개발 문화]

“컬리에는 (누구나, 언제든지 원하면) 열린 세미나를 통해
서로의 지식을 공유할 수 있어요.

그리고 우리 개발자들의 편의를 높이기 위해 서비스들을
끊임없이 라이브러리화하는 개발 문화도 자리잡고 있죠.

컬리만의 특별한 개발 문화를 소개합니다.”

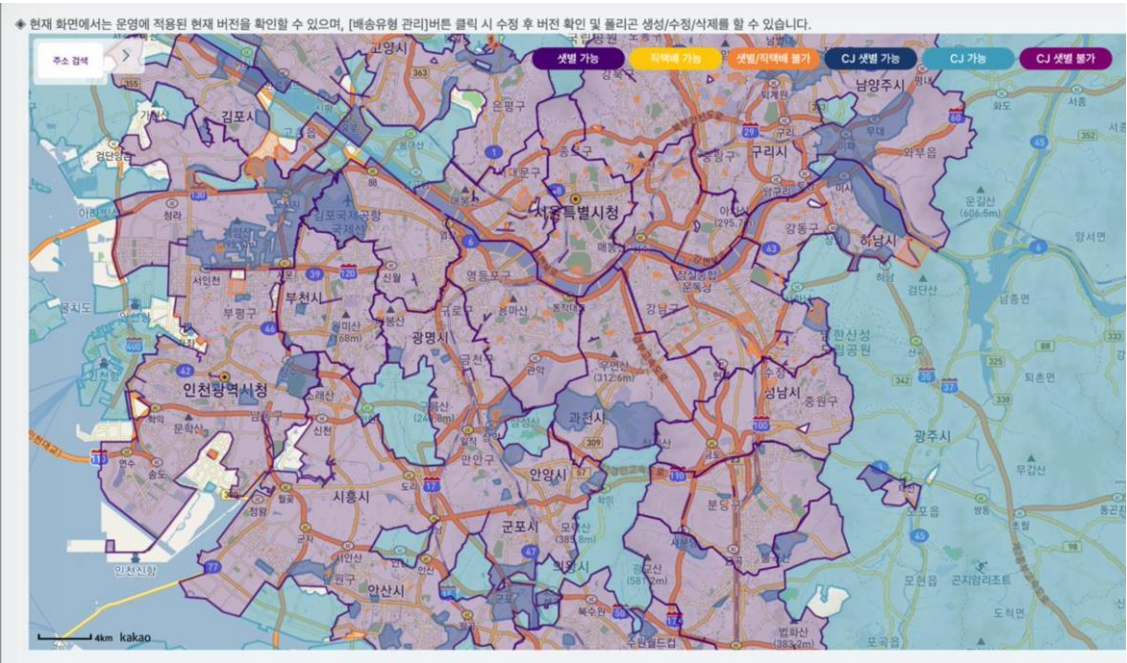




네번째 이야기-김성준 님 딜리버리 플랫폼 팀 소개



딜리버리 플랫폼 팀은 어떤 팀일까요?



컬리몰의 주문을
배송 지역, 배송 기사, 비용 등을 고려하여
최적의 차량 및 배송 경로 탐색!





네번째 이야기-김성준 님
개발자로서 성장하는 방법

개인적인 노력



타인과의 교류



이 두가지의 노력이 **함께 병행되어야** 개발자로서 **독보적으로 성장할 수** 있죠!



그렇지만, 타인의 노하우와 현장의 경험 습득을 위한 '타인과의 교류'는 혼자서 어찌할 수 없는 것이 현실!





네번째 이야기-김성준 님
컬리의 개발 지식 공유 문화에 대해



"git 전문가 000님! 저희도 전수해주세요!"

"물론입니다! 지금 바로 모이시죠!"



"제가 읽은 <오늘의 000> 내용이 너무 좋아서, 서평 공유합니다!"

컬리에서는 최신 트렌드를 주제로 한 세미나 이외에도
특정 주제에 대한 팀원들의 요청이나 본인이 얻은 개발 지식과 실패담에 대한 다양
한 주제의 세미나들이 규모를 막론하고 자주 진행됩니다!




도메인 지식 외적으로도 정말 많은 것을 배울 수 있었죠.






네번째 이야기-김성준 님 개발자들을 위한 개발자로서의 성장

 개발자들을 위한 개발자. 무슨 의미인가요?

➤ 다른 개발자들을 위해 라이브러리화, SDK, 개발 가이드 등을 진행하는 개발자입니다.

컬리에서는 자연스러운 생각들

 어떻게 하면 다른 팀원들의 손을 조금 덜어 줄 수 있을까?

 어떻게 하면 소스의 관리 포인트를 최소화 할 수 있을까?





네번째 이야기-김성준 님
직접 체험한 최고의 사내 라이브러리



배송 정보들을 실시간으로 처리하기 위해
주문, 배송 업체, 물류 센터 등 여러 곳에서 연동이 필요하네!

앗! 지금 당장은 결과가 우선적으로 필요한데,
일일이 구현을 해야 하나?

실시간 배송 정보 프로젝트 개발 중

아니 이미 구현되어 있잖아?!

WOW!



네이버 국회라이브러리 예시로 이해를 돕기 위한 사진입니다.





네번째 이야기-김성준 님

이제는 모듈화 된 많은 기능들을 모두 라이브러리화하자!



서비스 간의 연관성이 워낙 깊다 보니
함께하는 각 팀원분들의 리소스가 과하게 투입되어야 하네!



실시간 배송 정보 프로젝트 개발 중

라이브러리화

시간 및 개발 효율 증대를 위해,
각 서비스들에 필요한 많은 기능들을 모듈화 해보자!

➤ 개인의 개발 시간 축소 & 본인의 개발 부분에 집중 가능





네번째 이야기-김성준 님
컬리 합류 독려 메시지



어서 빨리 합류하세요!

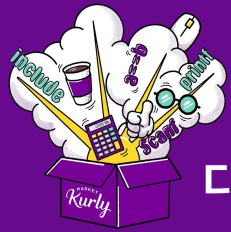


컬리의 개발 문화를 함께 경험



개인적인 성장과 값진 경험





컬리 개발자성장기

다섯번째 이야기의 주인공

저는

“경계선 의식 없이 일하는 것을 즐기며 문샷 프로젝트에 흥미를 가지고, 현실 문제를 기술로 풀어본 경험에 성취감을 느끼는 개발자”

최대복 입니다.



> 최대복님 | 데이터 플랫폼 ML 엔지니어

[물류에서 ML로 최적화 문제를 풀어 보기]

“물류 데이터를 통해 최적화 문제를 풀기 위한 고민의 과정과 ML을 기반으로 솔루션을 찾기 위한 Deepdive한 과정을 공유합니다. 또한 다른 팀원들과 협업하는 방식에 대해서도 함께 소개하고 싶습니다.”





다섯번째 이야기-최대복 님
컬리에서 ML로 풀고자 하는 많은 문제들...

마케팅(Marketing)

- 추천
- 검색
- Dynamic Pricing
- ...

풀필먼트 (Fullfillment)

- 작업 할당
- 스케줄링
- ...


공급망 (Supply Chain)

- 수요 예측
- ...





다섯번째 이야기-최대복 님
컬리에서 ML로 풀고자 하는 많은 문제들...

 각각 문제에 따라서 적합한 모델이 다르죠!

마케팅(Marketing)

- 추천
: Field-aware Factorization machine, Graph neural network, user embedding
- 검색
: Word2Vector, NLP
- Dynamic Pricing
: reinforcement learning

풀필먼트 (Fullfillment)

- 작업 할당
: Genetic algorithm, Meta Heuristic
- 스케줄링
: Linear programming

공급망 (Supply Chain)

- 수요 예측
: prophet, Lstm





다섯번째 이야기-최대복 님

컬리 풀필먼트 (Fullfillment) 문제에서 경험한 문제들



풀필먼트에서는 어디까지 최적화 할 수 있을까? (Constraint, Boundary condition)

✓ **Boundary Condition**

: 실제로 어떤 부분을 Handling 할 수 있을까? 어디까지 Handling 할 수 있을까?



환경이 변할 때 최적화 모델 성능이 유지될까? (Robust & Causality)

✓ 물류가 많이 들어오는 시간, 혹은 날짜에 최적 성능을 보장하는 알고리즘이 따라서 달라질 수 있을까?

✓ 그렇다면, 외부 환경과 알고리즘 효율의 Causality를 어떻게 알 수 있을까?



풀필먼트 최적화 시스템에서 테스트와 배포는 어떻게 하지? (풀필먼트 시스템에서의 MLOps)

✓ 실제 물류 데이터를 받아서 테스트 하는 환경 구축

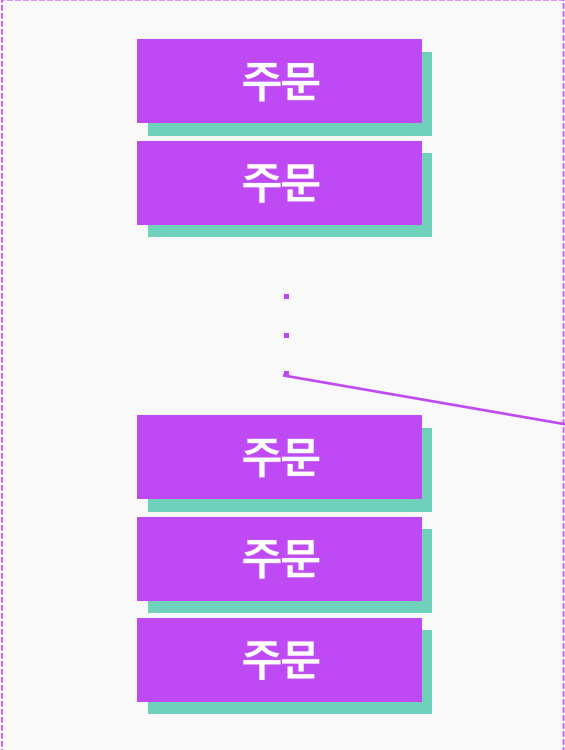
✓ 실제 변경된 최적화 방식이 좋은 쪽으로 변할지, 아니면 더 악화될 지 예상할 수 있을까?



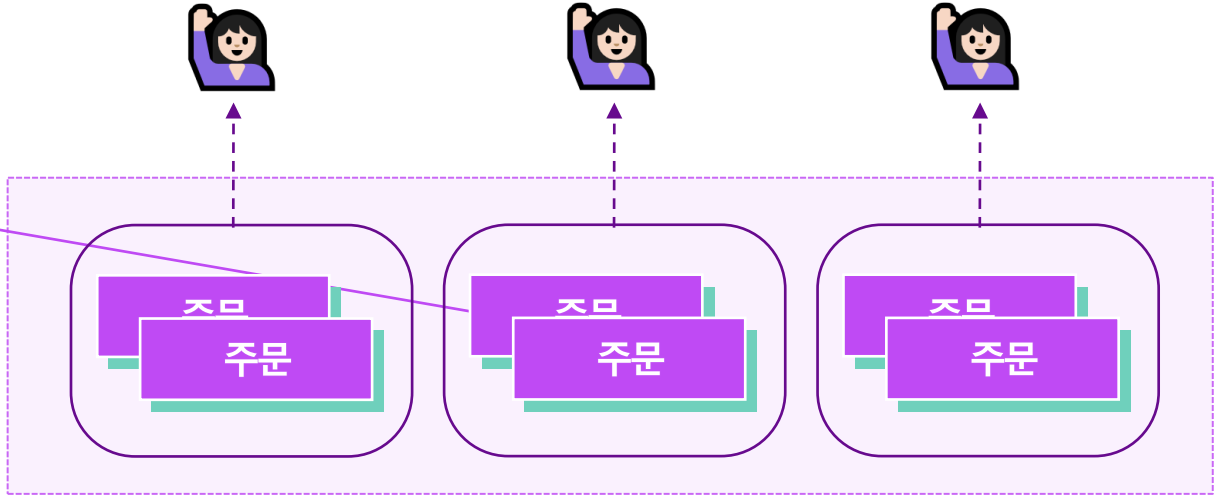
다섯번째 이야기-최대복 님
실제로 커리 풀필먼트 (Fullfillment) 문제에서 풀어야 할 문제



극도로 단순화 해본 커리 풀필먼트 시스템의 문제는,



» 많은 양의 주문이 인입되었을 때,
적절하게 주문을 묶고(Clustering), 순서를 결정한 뒤(Ordering),
어떤 작업자에게 할당(Matching) 할 지 풀어야 합니다.

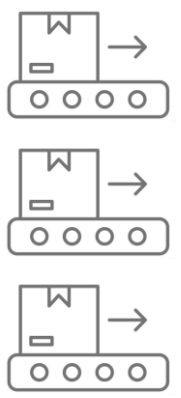




다섯번째 이야기-최대복 님
풀필먼트(Fulfillment)에서의 Constraint



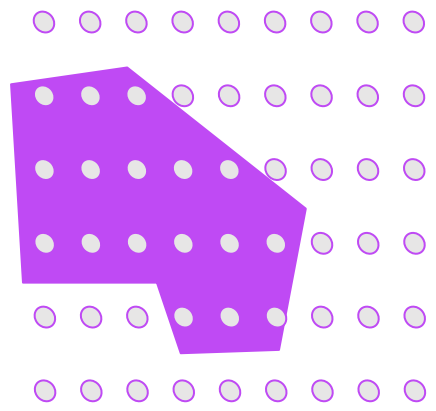
작업자가 한번에 나를 수 있는 물건의 수



물류 센터에서 병렬 처리할 수 있는 작업의 수



최적화 요청 처리 되어야 할 최소한의 시간 (with NP-hard)

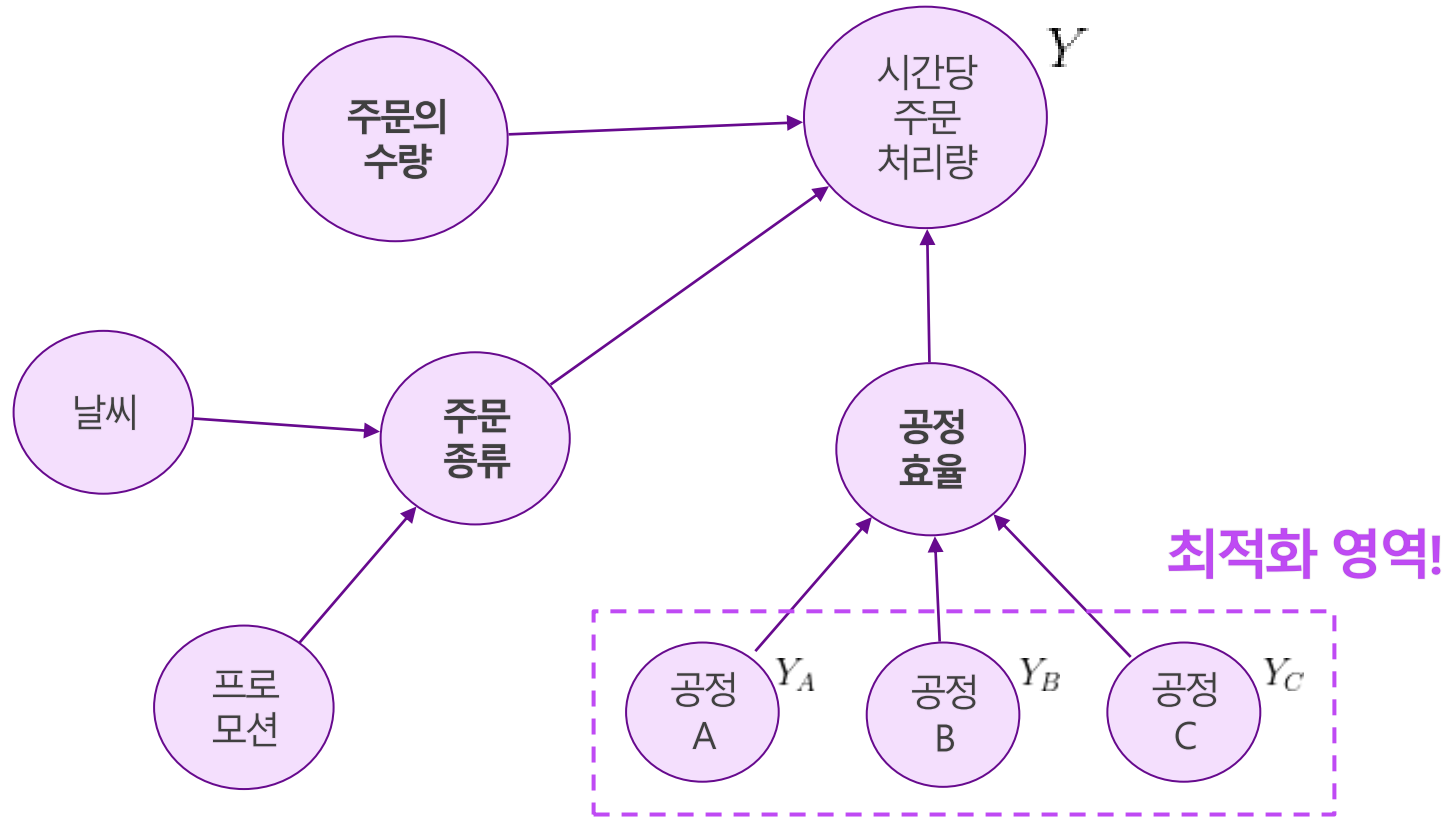


제한(Constraint)이 있는 정수






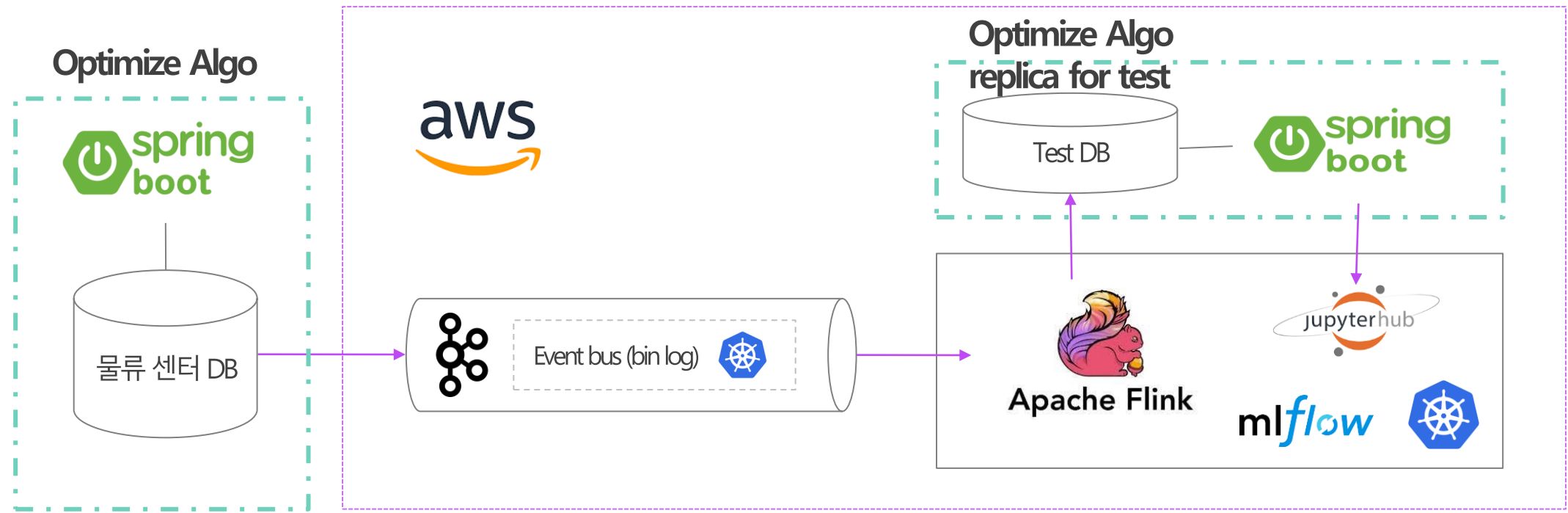
다섯번째 이야기-최대복 님
풀필먼트(Fulfillment)에서의 Causality





다섯번째 이야기-최대복 님
컬리의 테크 Spec을 소개합니다.

 컬리의 최적화된 로직은 풀필먼트(물류) 센터 IDC에 구축되어 있습니다!





다섯번째 이야기-최대복 님
앞으로의 개발 방향

실험 환경은 구축 완료



이제 테스트와 배포를 해보자!



기존 방식보다 향상된 것을 어떻게 알 수 있을까?
(기존 방식보다 향상되지 않았다면 배포X)

- ✓ Causality에 기반한 풀필먼트 최적화 성능 평가하기
- ✓ 컬리의 각 풀필먼트 공정에 적합한 Evaluation metric



풀필먼트(물류) 작업 관리자가 요청하는 시간 내 최적 결과 도출

- ✓ 서비스가 요구하는 response time 맞추기
- ✓ 모든 주문은 개별적인 하나의 작업으로 할당 되어야 함 (Entity integrity)





다섯번째 이야기-최대복 님
컬리 합류 독려 메시지



컬리에 합류하신다면, 아래의 경험을 얻으실 수 있어요!

컬리의 뛰어난 엔지니어들과 함께 일할 수 있어요.

경계선 의식 없이 공통의 문제를 갖고 데이터 플랫폼 팀원들이 함께 문제에 참여해요.

고도화 된 풀필먼트 시스템 위에서 최적화를 적용해볼 수 있는 기회예요.

- 데이터로 풀필먼트시스템의 효율성을 측정할 수 있는 경험
- 새로운 알고리즘을 실험해보고 이를 바탕으로 새로운 실험을 설계하는 경험

End to End 관점에서 최적화를 고민하고 적용해 볼 수 있는 환경이에요.

- 컬리 앱의 주문 첫 시작부터, 배송까지 존재하는 수많은 최적화문제를 고민

그 외에도 다양한 컬리가 풀고 있는 문제들을 해결할 수 있어요.

- 아이템 추천 (Item Recommendation)
- 수요 예측 (Demand Forecasting)



MARKET KURLY DEVELOPER MEET UP

FUN 하지만 뻔하지 않은, 컬리의 개발자 성장기

