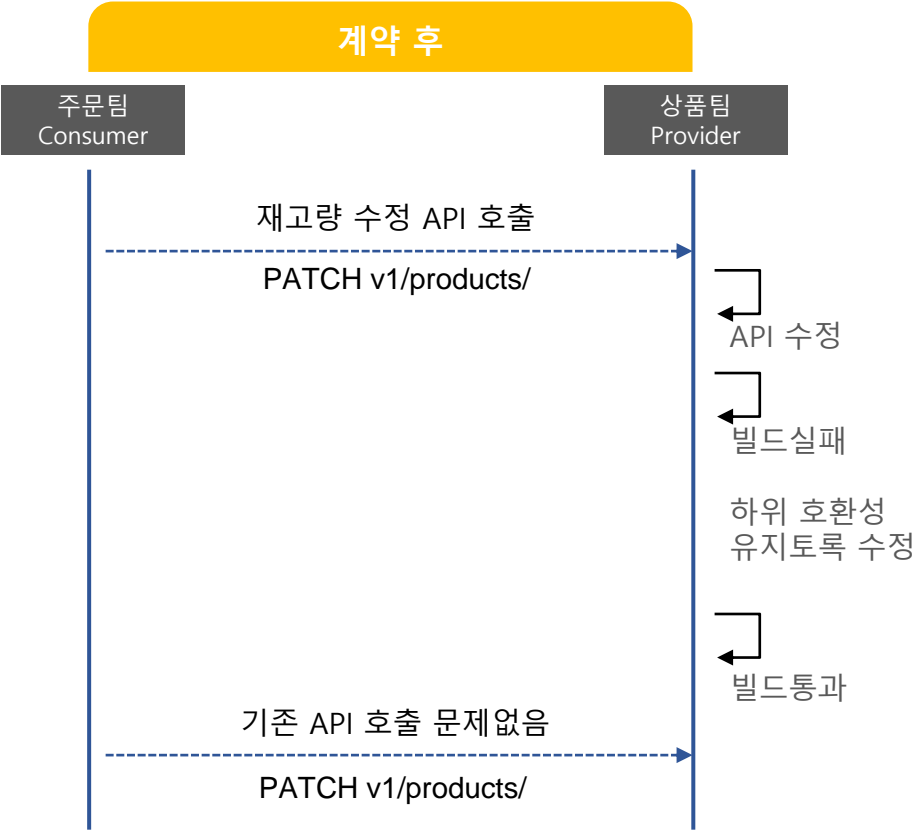
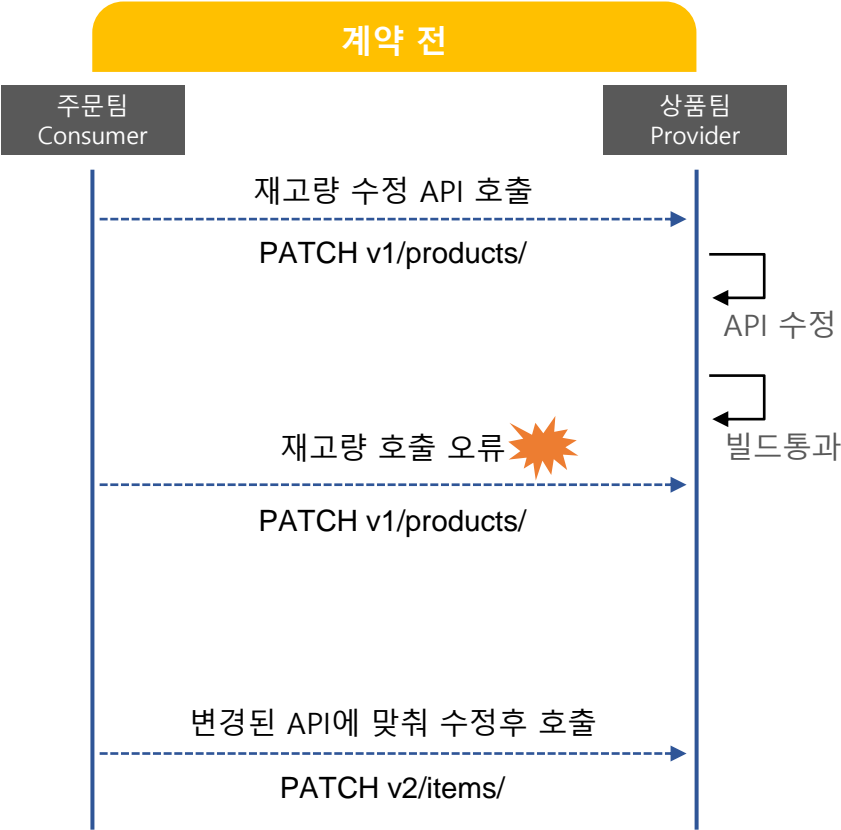


Contract Test



상품팀에서 API를 일방적으로 변경


소스 위치

products / src / main / java / com / example / template / ProductController.java

```
13
14 @GetMapping("/product/{productId}")
15 Product productStockCheck(@PathVariable(value = "productId") Long productId) {
16     return this.productService.getProductById(productId);
17 }
18 }
19
```

← /item/{productId} 로 변경

주문팀의 서비스 장애 발생



12 STREET

한 일반 상품

상품 정보

TV

가격 : 10000

재고수량 : 10

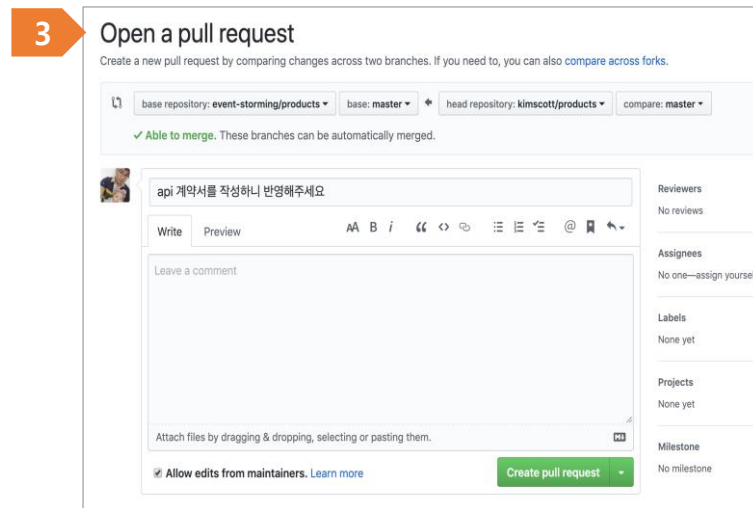
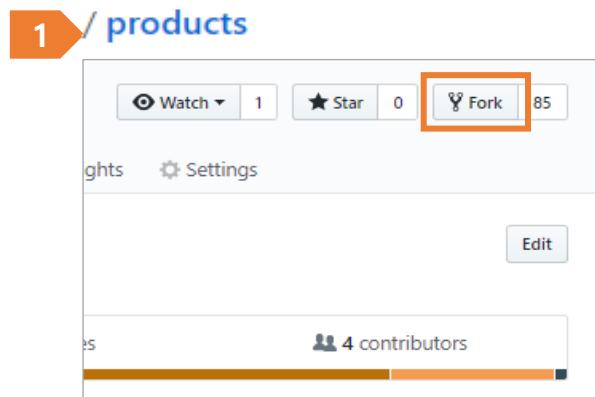
localhost:8080 내용:

Could not commit JPA transaction; nested exception is javax.persistence.RollbackException: Error while committing the transaction

확인

계약체결(1/2) - 주문팀에서 계약서 작성 후, 상품팀에 체결 요청

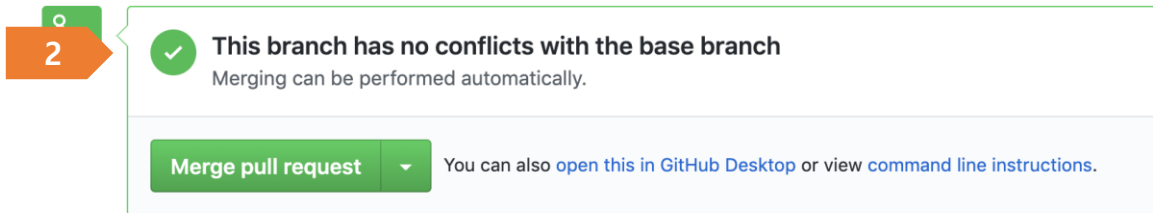
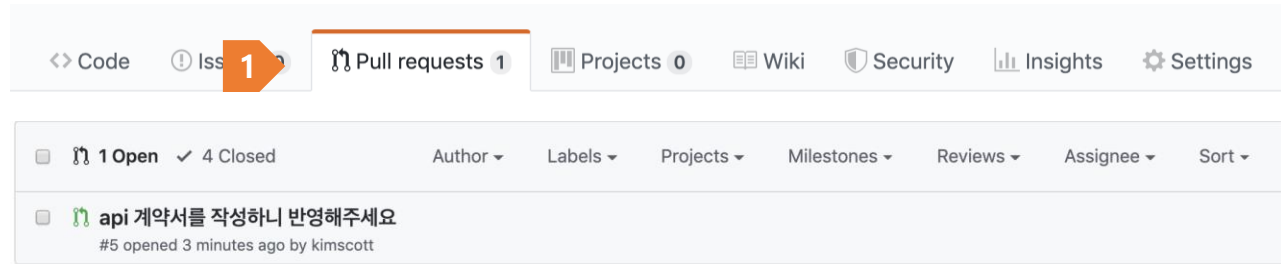
1. 상품팀 소스 복사 (포크 생성)
2. 주문팀이 계약서 생성
3. 주문팀에서 생성한 계약서 (productGet.groovy) 파일을 상품팀에 Pull request 요청함



api 계약서를 작성하니 반영해주세요

계약체결(2/2) - 상품팀에서 계약서 수락

- 상품팀 : Pull Request 메뉴 선택
- 상품팀은 해당 계약서를 accept 하여 반영함
- 상품 서비스는 이제부터는 계약서를 안지켰을때 아예 배포가 안됨



계약체결 후, 상품팀은 계약 위반으로 배포 실패함

CloudBuild 에서 mvn package 단계 실패

빌드 정보

상태	❗ 빌드 실패
빌드 ID	6db60b82-3b99-4aef-9870-ce85dce23ccd
이미지	—
트리거	master 브랜치에 푸시 (products)
소스	GitHub event-storming/products 🔗
Git 커밋	9997fc8cebf7e444d94b790378bb58dfc324362c 🔗
환경 변수	CLOUDSDK_COMPUTE_ZONE=asia-northeast1-a CLOUDSDK_CONTAINER_CLUSTER=standard-cluster-1
시작 시간	2019년 10월 29일 오후 3시 42분 59초 UTC+9
결린 시간	1분 54초

⚠ 로그가 너무 커서 이 페이지에 완전히 표시할 수 없습니다. 빌드 단계의 로그가 누락되거나 완료되지 않을 수 있습니다.

빌드 단계 모두 펼치기

❗ build

gcr.io/cloud-builders/mvn — clean package

1분 46초 ▾

실패 LOG

```
Step #0 - "build": 2019-10-29 06:44:51.547 DEBUG 69 --- [           main] o.s.c.c.v.m.stream.StreamStubMessages : Picked channel name is [event-out]
Step #0 - "build": [ERROR] Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 4.976 s <<< FAILURE! - in com.example.template.MessagingTest
Step #0 - "build": [ERROR] validate_productChanged(com.example.template.MessagingTest) Time elapsed: 0.221 s <<< ERROR!
Step #0 - "build": com.jayway.jsonpath.PathNotFoundException: No results for path: $('[productName']
Step #0 - "build":         at com.example.template.MessagingTest.validate_productChanged(MessagingTest.java:43)
Step #0 - "build":
```

상품팀에서는 하위 호환성을 유지하며, 추가 API 제공

products / src / main / java / com / example / template / ProductController.java

```
@GetMapping("/item/{productId}")
Product productStockCheck(@PathVariable(value = "productId") Long productId) {
    return this.productService.getProductById(productId);
}

@GetMapping("/product/{productId}")
Product productStockCheck1(@PathVariable(value = "productId") Long productId) {
    return this.productService.getProductById(productId);
}
```

기존의 하위 호환성 API 유지