[테스트] CDC 기반 Contract test

contract test 사용법을 익힌다.

Instruction

---

contract test 실습

contract test 시나리오

- 서비스 정상 작동 확인
- 상품서비스(8085)와 주문서비스(8081)를 기동한다.

```
cd orders
mvn spring-boot:run
cd products
mvn spring-boot:run
```

- 주문을 한다.

```
http http://localhost:8081/orders productId=2 quantity=3
customerId=1@uengine.org
```

- 계약(Contract) 위반 사항 만들기
    - 주문서비스에서 주문을 할때, 상품서비스의 api 를 호출한다.
        - Order.java 파일(45 행)의 restTemplate.getForEntity 확인
        - http://상품서비스/product/productId
- 상품서비스에서 해당 api 를 item 으로 변경한다.
- 상품서비스의 ProductController.java 확인
- 15 행에서 @GetMapping("/product/{productId}") 을
- @GetMapping("/item/{productId}") 으로 변경
- 상품서비스를 재시작 하고 주문해 본다.

```
http http://localhost:8081/orders productId=2 quantity=3
customerId=1@uengine.org
```

- 404 에러 발생!!

CDC(Consumer Driven Contract) 계약 체결

- Consumer 가 참조하는 코드를 Provider 일방적인 수정방지를 위한 Contract 적용
- Consumer 인 주문 서비스 개발자가 주도적으로 계약서를 작성(CDC)한다.
- order 서비스의 최상위 root 에 productGet.groovy 파일 참고

- **productGet.groovy 파일을 복사하여서, product 서비스의 test/resources/contracts/rest 폴더에 복사를 한다.**
- **실제로는 Git 환경에서 PR(Pull Request)을 요청하고 이를 상품팀이 수락한다.**
- **(contracts/rest 폴더는 없기때문에 새로 만들어야 합니다.)**
- **(contracts/rest 폴더를 만드는 이유는 productGet.groovy 파일에 package contracts.rest 라고 선언을 하였기 때문입니다.)**
- **계약에 의해서 product 서비스에서 Test, or Package 실행단계에서 에러가 발생한다.**
- **product 서비스의 package 명령을 호출한다.**

```
cd products
mvn package
```

- **test fail 에러 발생!!**

Consumer 와 체결한 계약**(Contract)**을 위반하여 상품팀에서는 빌드단계에서부터 실패하게 된다.

- **계약 위반을 해결하기 위하여 product 서비스는 기존의 /product 라는 api 를 유지해야한다...**
  - **product 서비스의 ProductController.java 에서 @GetMapping("/product/{productId}") 를 다시 생성한다.**
  - **product 서비스의 package 명령을 호출하여 봅니다.**

```
cd products
mvn package
```

- **테스트 성공 및 jar 파일 생성 완료!!**

주문서비스에서 테스트

- **주문서비스는 상품서비스에서 정상적으로 테스트를 적용하여 배포중인지 테스트를 할 수 있다.**
- **주문서비스가 상품서비스의 api 를 테스트 하기 위해서는 상품서비스에서 stub 파일을 제공해 주어야 한다.**
  - **상품 서비스에서 mvn install 을 하여 stub 파일을 Local(.m2 folder)에 생성한다.**

```
cd products
mvn install
```

- **주문서비스에서는 만들어진 stub 파일(Mock Server)을 바라보며 테스트를 진행한다.**
  - **order 서비스의 test/java/com.example.template/ProductContractTest.java 파일참고**
  - **@AutoConfigureStubRunner 에서 주문서비스의 stub 을 바라본다.**
  - **TestRestTemplate 으로 "/product/1" api 를 호출하여 결과값을 비교한다.**

CheckPoints

1. 주문서비스의 productGet.groovy 파일을 products 서비스에 계약을 걸고, 테스트를 성공시키시오.