

# 통합 테스트 작성하기

## - ProductFilter

실무에 바로 적용하는 프론트엔드 테스트

이재성

# 사용 브랜치



shopping-mall-integration-test

```
$ git checkout shopping-mall-integration-test
```

# ProductFilter의 비즈니스 로직

- ✓ 카테고리 목록을 가져온 후 필드에 정보가 올바르게 렌더링
- ✓ 상품명을 수정하는 경우 setTitle이 호출
- ✓ 카테고리를 클릭할 경우, 클릭한 카테고리가 체크
- ✓ 최소·최대 가격을 수정하면 setMinPrice·setMaxPrice가 호출

# 정리

## ProductFilter

- 테스트를 위한 사전 준비
  - 컴포넌트의 초기 필터 정보를 갱신하기 위한 filter 스토어 모킹
  - 카테고리 필드의 정보를 렌더링하기 위한 API 호출을 msw로 모킹
- 테스트 작성하기
  - Setup 과정 (beforeEach)에서 스토어 모킹
  - 스파이 함수를 통한 스토어의 액션 호출 여부 확인
  - getByLabelText, getByPlaceholderText 쿼리를 통한 요소 조회
  - findBy~쿼리를 통한 카테고리 API 호출 기다리기
- 장바구니: /cart/PriceSummary
  - shopping-mall-integration-test-with-answer 브랜치