

단건 배달 관리 서비스

강예준, 김원웅

https://youtu.be/0MM1nm6DJ_I

주제

네트워크 구조

체인코드

주제

- 단건 배달 관리 서비스
 - 배민1, 쿠팡이츠

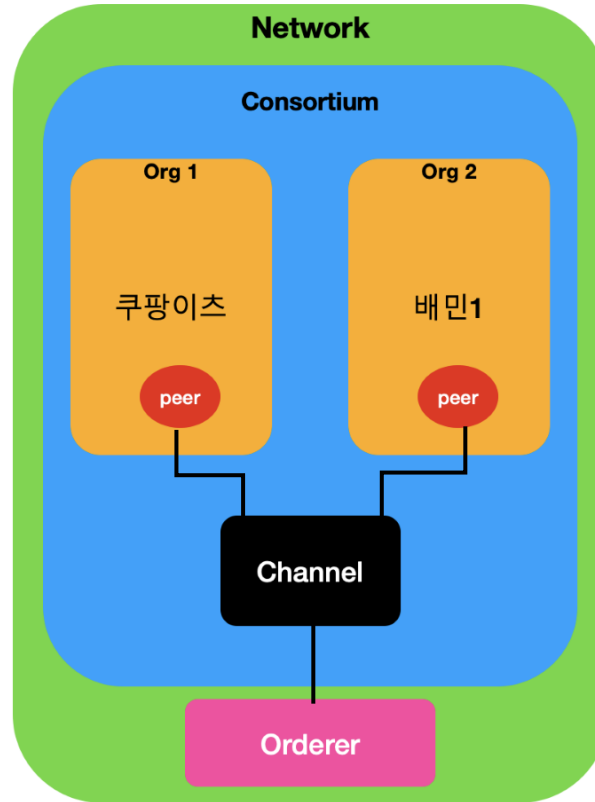
5일 업계에 따르면 배달의민족 단건배달 서비스 '배민1'을 운영하는 우아한형제들 고객센터에는 최근 단건배달 관련 소비자 민원이 접수됐다. 소비자 A씨는 배민1으로 서울 영등포구 여의도 소재 치킨집에서 음식을 주문했는데, 앱을 통해 배달원 이동 경로를 확인해보니 배달원이 배송 경로를 이탈해 다른 지역을 돌고 있었다고 주장했다.

배달원이 단건배달 서비스인 배민1과 쿠팡이츠로부터 각각 주문을 받은 뒤 한꺼번에 배달할 경우, 실질적으론 묶음배달이지만 각 플랫폼에선 단건배달로 잡히는 '맹점'이 있단 얘기다. 배달 앱 고객센터에도 관련 사례에 대한 민원이 지속 접수되는 것으로 파악된다.

이미경, "빨리 온다길래 '단건배달' 주문했는데...'한강 건너 빙빙' 왜?", 「한경 경제」 2022.01.05

“모바일 기기의 고유번호인 **IMEI**를 통해 해결”

네트워크 구조



체인코드

• 배달 & 배달원 클래스

TS delivery-man.ts
TS delivery-manage.ts
TS delivery.ts

```
export class DeliveryMan {  
  IMEI: string;  
  phone: string;  
  inDelivery: boolean;  
  
  public constructor(IMEI: string, phone: string) {  
    this.IMEI = IMEI;  
    this.phone = phone;  
    this.inDelivery = false;  
  }  
}
```

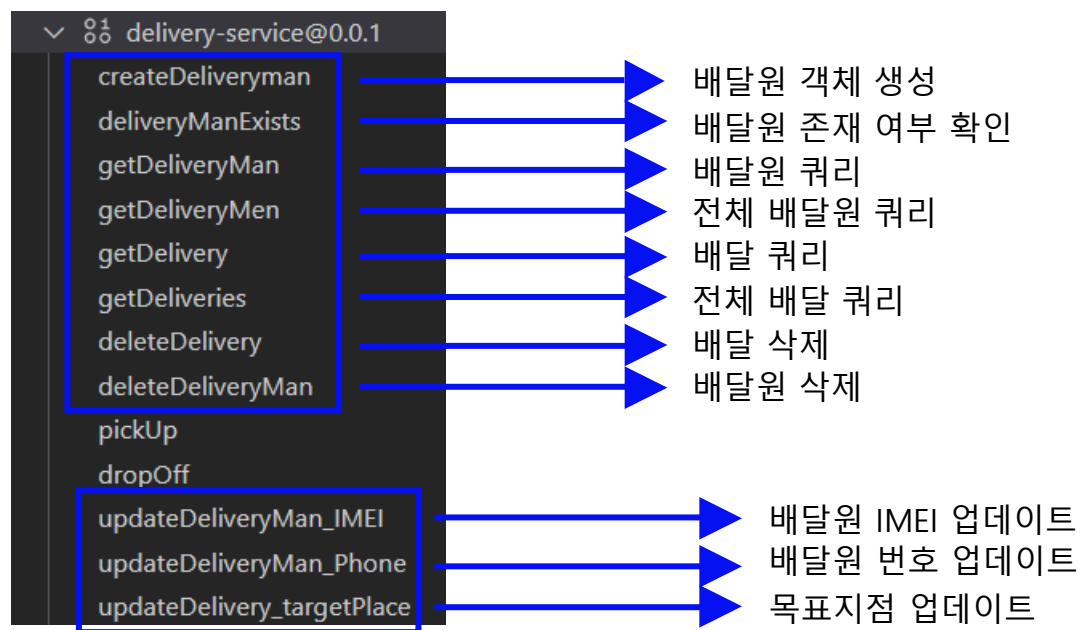
배달원 클래스

```
export class Delivery {  
  key? : string;  
  IMEI: string;  
  target_place: string;  
  in_time: number;  
  out_time: number;  
  isDone: boolean;  
}
```

배달 클래스

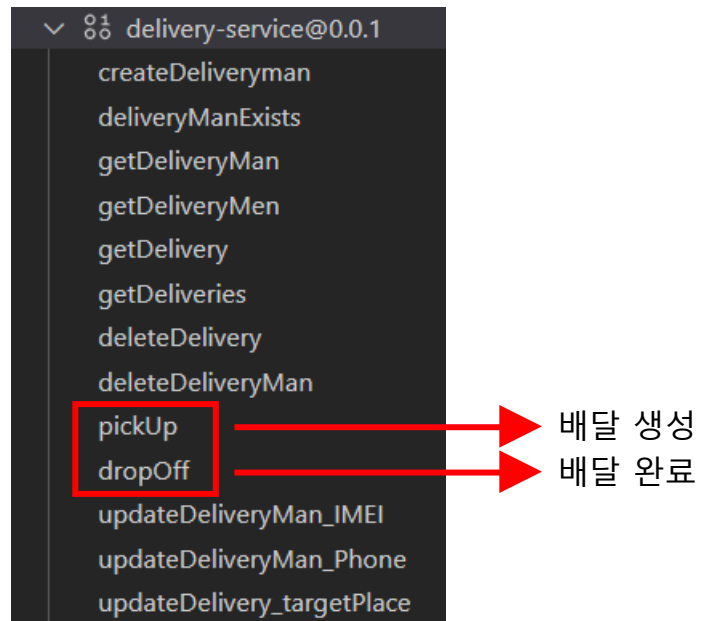
체인코드

• 체인코드 (CRUD)



체인코드

- 메인 체인코드



체인코드

- pickUp

```
@Transaction()
async pickUp(ctx: Context, IMEI: string, target_place: string) {
    // 기존의 배달원 state 불러오기
    const deliveryMan = this.getDeliveryMan(ctx, IMEI);

    // 이미 배달이 진행 중이라면
    if ( (await deliveryMan).inDelivery ) {
        console.log('It is impossible to deliver multiple items at once');
        return;
    }

    // 배달 객체 생성
    try {
        const delivery: Delivery = {
            key: this.keyGen(IMEI),
            IMEI: IMEI,
            target_place: target_place,
            isDone: false,
            in_time: ctx.stub.getTxTimestamp().seconds.low,
            out_time: -1
        };

        // 배달원 상태 변경
        (await deliveryMan).inDelivery = true;
        await ctx.stub.putState((await deliveryMan).IMEI, stateValue(await deliveryMan));

        //배달 객체 world state에 입력
        await ctx.stub.putState(delivery.key, stateValue(delivery));
    }

    catch(err) {
        console.log(err);
    }
}
```

```
// 키 생성
private keyGen(IMEI: string) {
    return `D-${IMEI}`;
}
```


체인코드

- dropOff

```
// 배달 완료
@Transaction()
async dropOff(ctx: Context, IMEI: string) {
  try {
    // delivery key 생성
    const key = this.keyGen(IMEI);

    // delivery 객체 불러오기
    const byteItem = await ctx.stub.getState(key);
    const delivery = toItem(byteItem);

    // deliveryMan 객체 불러오기
    const deliveryMan = this.getDeliveryMan(ctx, IMEI);

    // 배달 완료 시간
    delivery.out_time = ctx.stub.getTxTimestamp().seconds.low;

    // 배달 완료 시간 출력
    const delivery_time = delivery.out_time - delivery.in_time;
    const time_hour = Math.floor(delivery_time / 60 / 60);
    const time_min = Math.floor(delivery_time / 60 % 60);
    const time_sec = Math.floor(delivery_time % 60);
    console.log(`the delivery took ${time_hour}h ${time_min}m ${time_sec}s`);

    // inDelivery -> false
    // isDone -> true
    (await deliveryMan).inDelivery = false;
    delivery.isDone = true;

    // delivery, deliveryMan state 업데이트
    await ctx.stub.putState(IMEI, stateValue(await deliveryMan));
    await ctx.stub.putState(delivery.key, stateValue(delivery));
  }

  catch(err) {
    console.log(err);
  }
}
```