

스윗트래커 과제안내

요구사항

1. 작업물에 관련해서는 DB를 제외하곤 아무런 제약이 없습니다. 외부 라이브러리등을 이용해서 자유롭게 개발하시면 됩니다.
2. 시간은 과제를 전달받은 날로부터 일주일내로 전달해주시면 됩니다.
3. 반드시 네이티브 앱으로 작성이 되어야 합니다.
4. 반드시 내부 DB는 iOS는 코어데이터, Android는 ROOM을 이용해서 구성하셔야 합니다.
5. 바인딩되는 데이터는 아래의 url을 통해서 아래 json 데이터를 서버에서 받아오는 형태로 구현이 되어야 합니다.
http://img.sweettracker.net/image/mobile_test/mobile.json
6. 받아온 데이터는 내부 DB에 저장 후 다시 꺼내와서 화면을 구성해주세요.
7. UI는 iOS : <https://zpl.io/V11e5LE>, Android : <https://zpl.io/V0mMjOK> 를 통해서 확인하시면 됩니다.
각자 지원한 OS로 개발을 진행해주시면 됩니다.
 - id : sweettracker.test@gmail.com
 - pw : sweettracker!@
8. Android의 경우 Kotlin, iOS는 Swift로 작성해주세요.

JSON data

요청 URL : http://img.sweettracker.net/image/mobile_test/mobile.json

데이터 맵

```
1  {
2    "parcelCompanyCode" : String (택배사코드),
3    "parcelCompanyName" : String (택배사이름),
4    "parcelInvoice" : String (송장번호),
5    "parcelLevel" : Integer (택배상태),
6    "parcelDeliverTime" : String (배달예정시간),
7    "purchaseItemImg" : String (상품이미지),
8    "purchaseItemName" : String (상품명),
9    "purchaseItemDate" : String (상품구매일자),
10   "trackingDetail" (배송 상세): [
11     {
12       "time" : String (시간),
13       "where" : String (위치),
14       "status" : String (상태)
15     }
16   ]
}
```

parcelLevel의 경우 0 -> 송장만 등록이 된 경우 , 1 -> 집하, 2-> 배송중, 3 -> 배달출발, 4 -> 배달완료 로 매핑을 하여 사용하시면 됩니다.

과제 테스트 후

과제가 완료 되면 소스파일을 압축 후 recruit@sweettracker.co.kr 으로 전달해주세요. 소스파일 전달받은 후 영업일 기준 2~4일 이내 결과를 통보해드립니다. 과제 테스트에 통과하시는 분에 한해서, 면접을 진행하게 됩니다.