

[KAKAO PAY] Data Team Homework

카카오페이 데이터팀 과제입니다.

Problem

핀테크 흐름과 함께 FDS(Fraud detection system)("이상 금융거래 탐지 시스템") 도입을 통한 금융 모니터링을 하고 있다.

FDS를 단순히 커다란 규모의 전산 시스템, 보안 시스템으로 여길 수도 있지만 사실 가장 정성적인 개념으로 IT 기술적인 부분 뿐만 아니라 모니터링 요원의 분석과 연계 기관이 유기적으로 연결되고 정보와 데이터가 엮인 시스템이다.

아래는 카드사의 부정방지 시스템의 활용 예이다.

카드사 부정방지시스템 직원들, 잠 못드는 24時

“주부 고객이 룸살롱에서 결제를? 바로 전화걸죠”

시간대별로 자주 발생하는 카드 부정 사용 사례

◆2~3교대로 24시간 카드 부정사용 감시

◆낮 12시에 서울 시청에서 쓴 카드가 오후 3시 베트남에서도 결제

◆카드 사고의 절반 이상이 주말에 발생

평소 이용패턴 벗어난 결제
도난카드로 의심하고 확인

서울서 결제 후 바로 해외결제
위조카드 가능성 거의 100%

승인내역 SMS 적극 이용을



이미지 출처 : 금융보안연구원 FDS 세미나 신한 카드사 발표 자료 "異常 거래 탐지시스템(FDS) 적용 사례" 발췌

FDS 의 시스템 중 일부분인 Rule Engine 모듈을 포함한 Restful Service 를 구현하고자 한다. 이 서비스를 활용하여, 부정방지 시스템 모니터링 직원들은 고객의 사용자 ID로 검색을 하고 그 사용자가 특정 룰에 해당하는지를 알 수 있다.

카카오 페이의 머니 서비스를 기준으로 아래 요구사항을 참고해서 Rule Engine 및 Rest API 를 구현하라.

요구 사항

- Gradle 기반 프로젝트
- 여러 개의 Rule 을 추가 및 유지보수를 고려한 Flexible 한 룰 엔진 구현
- Rest API 구현 (Rule engine 을 활용한 Rest API 구현)

Rest API Spec.

고객의 사용자 ID를 입력 후에 사용자가 Rule 엔진의 조건에 의해서 모니터링 대상인지를 판별한다.

(룰은 아래의 RuleA, RuleB, RuleC 가 존재한다.)

API 기본 정보

메서드	요청 URL	출력 포맷
GET	~/v1/fraud/{user_id}	JSON

요청 변수 (Request parameter)

요청 변수	타입	필수 유무	설명
user_id	long	Y	사용자 ID

출력 결과

필드	타입	설명
user_id	long	요청 변수의 사용자 ID
is_fraud	boolean	룰에 의한 이상 거래 가능성 여부
rule	string	해당되는 룰 이름 (여러개 일 경우는 여러개 표기(콤마로 구분))

Rule set

RuleA.

- 카카오머니 서비스 계좌 개설을 하고 1시간 이내, 20만원 충전 후 잔액이 1000원 이하가 되는 경우

RuleB.

- 카카오머니 서비스 계좌 개설을 하고 7일 이내, 카카오머니 받기로 10만원 이상 금액을 5회 이상 하는 경우

RuleC.

- 2시간 이내, 카카오머니 받기로 5만원 이상 금액을 3회 이상 하는 경우

사용자 로그 유형

- * 카카오머니 서비스 계좌 개설 로그
 - 발생 일시
 - 사용자 ID
 - 카카오머니 계좌번호

- * 카카오머니 송금 로그
 - 발생 일시
 - 사용자 ID
 - 송금 카카오머니 계좌번호
 - 송금 이체전 카카오머니 계좌 잔액
 - 수신 카카오머니 계좌번호
 - 수신 사용자 ID
 - 송금 금액

- * 카카오머니 충전 로그 (은행계좌에서 머니 충전)
 - 발생일시
 - 사용자 ID
 - 카카오머니 계좌번호
 - 충전 금액
 - 은행 계좌 (은행 계좌에서 머니 계좌로 이체)

- * 카카오머니 받기 로그
 - 발생 일시
 - 사용자 ID
 - 받기 카카오 머니 계좌 번호
 - 받기 전 카카오머니 계좌 잔액
 - 보낸 카카오머니 계좌번호
 - 보낸 사용자 ID
 - 받기금액

Constraints

- Rule 엔진 영역은 POJO 형태로 구현 (lombok, guava, Spring 기능 일절 사용하지 말것)
- User log(Datasource 영역) 구현은 필수가 아님 (사용자 로그가 있다고 가정하고 구현)
- Rest API 영역은 Spring 또는 Spring-boot 기능을 활용할 것
- (참고사항: Rest API 의 사용자 로그 영역은 mock-up data 를 활용할 것)
- MVC Pattern 및 Design Pattern 을 잘 활용하면 가산점 부여
- Rule engine 의 경우, Prototype 수준이 아닌 Production 수준으로 구현
 - 오류없이 동작 (Unit test 코드 추가할 것)
 - Thread-Safety 준수
 - 성능 및 공간 최적화

Evaluation

- Gradle 프로젝트 및 Spring 을 활용한 Rest API 구성 여부
- Rule Engine: OOP 기반 클래스 설계가 잘 되었는지 여부 (interface 포함)
- Rule Engine: 논리적 Rule 단계를 구조화가 잘 되었는지 여부

제출방법

- 실행 방법 문서 & 코드 설명 문서 포함
- Gradle 프로젝트와 함께 전체를 zip 으로 압축해서 받은 메일 주소로 제출할 것