1. **Try**
   1. **실시간 분석 (Real time Analysis)**
      1. **성능 도전 : 응답속도 0.5초**
   2. **기술적 + 구조적 => 경험 / 학습**
2. **Done**
   1. **API**
   2. **Entry Point : 영화 검색 API 요청…**
3. **To Do**
   1. **결과 : 음악(노래)을 추천**
   2. **알고리즘 (실 시간 : 사용자의 요청 시점)**
      1. **음악의 검색 / 수집 기능 : 검색어 ‘제목’**
         1. **API URL 5개 : 병렬/동시 비동기 호출 5개 REST API 호출**
         2. **메모리에 Merge(병합)**
      2. **비 정형 데이터 보관**
      3. **필요로 하는 변수(입력값) 정제 과정**
         1. **복수검색결과여부(many\_search\_result\_yn = Y/N)**
         2. **Map<String,Object> parameter = new HashMap<>();**
         3. **parameter.put(‘many\_search\_result\_yn’,’y’)**
      4. **Decider(의사결정자)**
         1. **앙상블 모델**
      5. **규칙 (Rule)** 
         1. **엔진 : MH\_music\_rule\_engine.jar**
            1. **RuleEngineInterface**
            2. **interface : fire(Map<String,Object> parameter)**
            3. **if(parameter.get(‘many\_search\_result\_yn’) == ‘y’) recommend ‘this.music’**
         2. **정해진 규칙**
            1. **경험을통해 알고 있는 규칙**
            2. **ex) 가장 복수로 검색된 노래, 복수 검색 노래 중에서 영화 검색 조건에 AND에 모두 검색되는 노래.**
            3. **수집정보 저장 → 누적… 사이즈 10000건 이상.. 정도….**
            4. **movie\_musice\_table**

**insert…..**

* + - 1. **ML 규칙**
         1. **데이터가 일정 부분 누적되어야 함.**
         2. **데이터가 있어야 함.**
         3. **배치/실시간배치 → 모형 산출해 내야 함.**

1. **Concept**
   1. **모듈( Module) (응집도:Cohesion , 결합도:Coupling)**

1. 영화 제목 검색

- 제작년도, 제목, 감독, 배우, 나라

2. Flow

1) 사용자의 영화 선택

2) 영화 정보 API를 통해서 검색

3) 검색 결과를 음악 추천 API에 전달

4) 영화

- 제작년도

- 제목

- 감독

- 배우

- 나라

3. 음악(노래) 검색

\* 제약사항

- spotify는 한국에서 서비스가 안되기 때문에 VPN을 해외로 돌려 이용

- 애플뮤직은 한국 음악이 적음

- shazam은 공식 api가 없음

- 빌보드는 hot100 등 차트 관련 api

\* API 후보 / 크롤링(웹사이트 검색.....)

- SpotifyApi ( https://github.com/svkaka/SpotifyAlbumSearch )

- Itunes API ( https://itunes.apple.com/search?key=value )

- ManiaDB ( http://www.maniadb.com/api/ )

- billclintonswag.com/

- www.last.fm/api

1. 3개 정도의 음악 검색 API를 정의함.(수집)

2. Spring Boot 서버 개발을

1) Spring Boot Initializer

- spring boot rest server depencency

- Spring Web.

2) localhost:8080/hello --> Welcome 응답 주는 REST 서버 개발

3. 3개의 URL에 동시에 요청 하는 Client를 개발

1) Java 개발