1. 얼굴 비교
   1. 파일 리스트를 불러온다.
      1. 기존에 만든 함수를 활용하여 리스트를 빠르게 불러온다.
   2. 얼굴비교
      1. 빠르게 작업하기위해 이미지는 따로 입력하지 않고, RekoServiceImpl에서 이름으로 직접 입력한다.
      2. Controller에서 iRekoServ.faceCopare()로 들어가준다.
      3. public boolean faceCompare(){
         1. AmazonS3 s3 = getS3(); // s3가져오기
         2. URL imgUrl1 = s3.getUrl(BUCKETNAME, “이미지이름1”); //이미지 URL 가져오기
         3. URL imgUrl2 = s3.getUrl(BUCKETNAME, “이름2”);
         4. List<CompareFacesMatch> faceLst = CompareFace(getRekognition(), imgUrl1, imgUrl2);//얼굴 비교
      4. private List<CompareFacesMatch> CompareFace(AmazonRekognition rekognition, URL imgUrl1, Uimgrl2){

//이미 정보 얻기

* + - 1. Image source = new Image().withBytes(getImageByteBuffer(imgUrl1));
      2. 이미지 target = ------------------------------(imgUrl2));
      3. Float similarityThreshold = 90F;

//얼굴비교

* + - 1. List<CompareFacesMatch> faces = getCompareFaceResult(rekognition, source, target, similarityThreshold);
      2. // 리스트에 비교 결과가 나온다

//얼굴 비교하는 함수

* + 1. private List<CompareFacesMatch> getCompareFaceResult(AmazonRekognition rekognition, Image source, Image target,Float similarityThreshold) {
       1. CompareFacesRequest req = new CompareFacesRequest().withSourceImage(source).withTargetImage(target).withSimilarityThreshold(similarityThreshold);

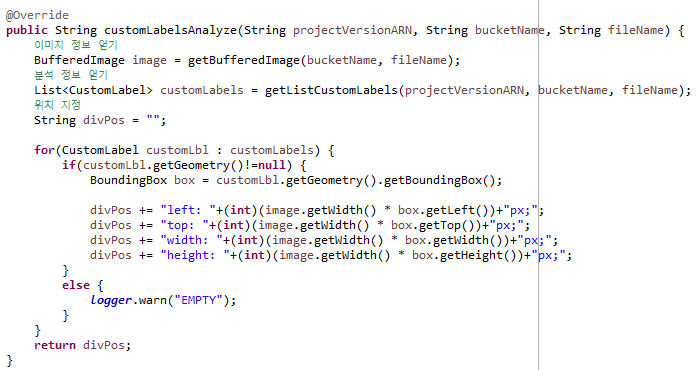
* + - 1. CompareFacesResult result = rekognition.compareFaces(req);

return result.getFaceMatches();}

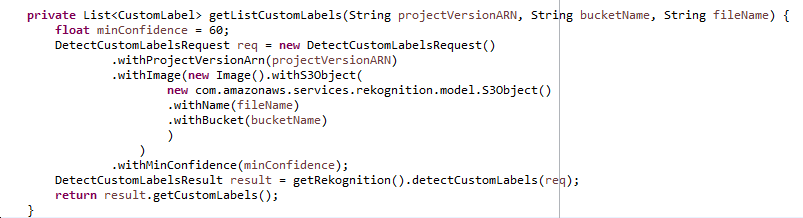
* 1. 마지막에 Controller에선 boolean값으로 결과가 온다. 그것을 jsp파일에 보내준다.
     1. jsp에선 <c:if test=”${isMatch == true}”></c:if>로 받아 결과를 맞춰서 출력해준다.

<Rekognition 사용자정의 모델>

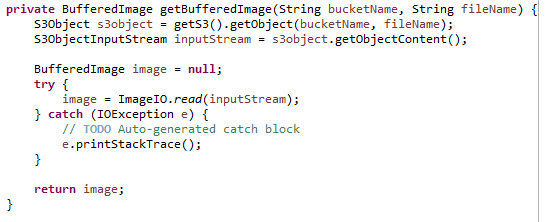
1. 먼저 Rekognition에서 사용자정의 모델로 들어가 시작한다. (리전확인 필수)
2. 버킷을 생성하라고하면 생성한다.
3. 프로젝트를 만들어준다. 간단함
4. 데이터세트를 만들어준다.
   1. 이미지를 업로드하고, label을 추가하고, 이미지를 클릭하여 Asign하여 라벨을 추가해주고, drawbind? 로 이미지에서도 더 자세하게 인식할 부분을 잘라준다.
   2. 저장한 후, 모델 만들기로 모델을 생성한다.
5. JAVA에서 모델 이용하기
   1. 모델 시작하기
      1. 모델의 ARN을 controller로 보내준다.
      2. iRekoServ.modelStart(modelARN));을 생성하고,
      3. public String modelStart(String modelARN){
         1. int minInterfaceUnits = 1; // 최소 인터페이스 유닛?
         2. StartProjectVersionRequest req = new StartProjectVersionRequest().withMinInferenceUnits(minInterfaceUnits).withProjectVersionArn(modelARN); //start 리퀘스트
         3. StartProjectVersionResult result = getRekognition().startProjectVersion(req) // 결과
         4. return result.getStatus();
   2. 모델 종료하기
      1. 모델은 시작하면 요금이 청구되기 때문에 사용하고 바로 종료시켜줘야 한다.
      2. 모델 시작메소드에서 start를 stop으로 바꾸면된다.
   3. 이미지 분석
      1. controller로 ARN, bucketName, fileName을 보내준다.
      2. iRekoServ.customLabelsAnalyze(ARN, bucketName, fileName)메소드만들어줌



* + - 1. BufferedImage로 이미지정보를 받고,
      2. List<custumLabel> customLabels로 분적정보를 받는다.
      3. 그리고, divPos로 jsp에서 이미지에 분석한 결과를 사각형 도형으로 보여주기 위해 그 style정보를 담아 리턴한다.



* + - 1. S3에서 bucketName과 fileName으로 이미지를 받고, 그것으로 DetectCustomLabelsRequest를 보낸다.
      2. getRekognition().detextCustomLabels(req)로 결과를 받고 result.getCustomLabels()로 결과를 받아준다.



* + - 1. bucketName과 fileName으로 S3에서 파일을 불러온다.
      2. 그것을 s3object.getObjectContent()로 stream으로 받아준다.

사용자 정의 Rekognition은 요금이 청구될수 있기 때문에 모두 삭제해준다.