

ai 서비스 인터넷 서버 (스프링 BOOT)	java, servlet, jsp, spring html, javascript, css, jquery
ai 서비스 구현 사례	naver cloud platform api
iot ai 스피커 구현 사례	ai 스피커 적재 컴퓨터(라즈베리파이 os) – kt server 연동

2. 암호 변경(필수)->기억하기
3. 오른쪽 상단 콘솔 클릭하기
4. 가운데 위 콘솔 시작하기
5. 왼편 메뉴 products\$services
6. ai service 클릭
7. 화면 하단 스크롤
8. +application등록 클릭
9. 각 clova 항목 서비스옆 서비스설명/요금안내 클릭 개발가이드 클릭
읽어보자
10. 해당 서비스 항목 선택
- 11.

< Application 1

Application 이름 설정

AI · Naver Service 를 이용할 Application 을 등록하세요

Application 이름

kdigital

Service 선택

Application에서 이용할 Service 를 선택하세요

☐ CLOVA☐ CLOVA Speech Recognition (CSR)☒ CLOVA Face Recognition (CFR)☐ CLOVA Voice - Premium☐ CLOVA Sentiment☐ CLOVA Summary

서비스 설명/요금안내

개발 가이드

서비스 설명/요금안내

개발 가이드

서비스 설명/요금안내

개발 가이드

서비스 설명/요금안내

개발 가이드

서비스 설명/요금안내

개발 가이드

위 항목 외에도 필요한 항목들 체크한다.

아래로 스크롤한다

서비스 환경 등록

Application에서 이용할 서비스 환경을 등록하세요

Web 서비스 URL(최대 10개)

http://localhost:9001

+추가



X삭제

Android 앱 패키지 이름(최대 10개)

안드로이드 앱 패키지 이름을 입력하세요.예) com.example.m

+추가

iOS Bundle ID(최대 10개)

iOS앱의 Bundle ID를 입력하세요.예) com.naver.example.MyN

+추가

취소


등록

웹서비스 url에 아무 url 입력 - 추가 버튼 - 등록 버튼 클릭한다.





Application 1

등록한 Application 정보를 확인하고 관리합니다.

+ Application 등록

개발 가이드 상품 더 알아보기 새로고침 

삭제

<input type="checkbox"/> App 이름	서비스구분	당일 사용량	당월 사용량	한도 및 알림 설정	등록일
<input type="checkbox"/> kdigital 인증 정보  변경	 CLOVA Face Recognition (CFR) 	0% <div></div>	0/50,000 회 0% <div></div>	0/300,000 회 한도 및 알림 설정 	2021-09-24

화면을 확인한다

인증 정보를 클릭한다.

client id와 client secret 값은 ai makers kit의 kt 인증 api key와 같이 네이버 서버의 ai 서버와 통신하기 위한 인증 데이터이다.

활용할 AI Services

1. CLOVA Speech	음성 메모, 영상 자막 생성, 통화 녹취록 관리 등의 서비스를 만들 때 활용할 수 있는 음성인식 서비스
2. CLOVA Voice	입력된 텍스트를 RESTful API 방식으로 전달하면 서버에서 인식하는 API
3. CLOVA Speech Synthesis(CSS)	입력한 텍스트를 자연스러운 목소리로 재생해주는 음성 합성 API
4. CLOVA Premium Voice(CPV)	CLOVA의 인공지능 기술로 더 사람같은, 고품질의 합성음 제공
5. CLOVA Speech Recogniton(CSR)	사람의 목소리를 텍스트로 바꿔주어 다양한 음성 인식 서비스에 활용
6. CLOVA OCR	인쇄물 상의 글자와 이미지를 디지털 데이터로 자동으로 추출하는 기술

7. CLOVA Face Recognition(CFR)	이미지 속의 얼굴을 감지하고 인식하여 얻은 다양한 정보를 제공
8. Pose Estimation	이미지 속의 사람을 감지하고 몇명이 어떤 포즈를 취하고 있는지에 대한 좌표 정보
9. Object Detection	이미지 내 사람 및 자동차 등 객체의 타입과 위치를 감지하여 정보를 제공
10.CLOVA Chatbot	ai 상담
11. API Gateway	안정적인 API 호출을 돕는 다양한 관리 기능과 모니터링 대시보드 제공

CLOVA Face Recognition(CFR) 개념 이해

CLOVA Face Recognition(CFR) 실습

1. 네이버 ai 서비스 사이트 로그인

얼굴인식-id, 암호,

개발자 가이드+ 자바코드 연동

1명	<code>{"info":{"size":{"width":264,"height":200},"faceCount":1},"faces":[{"celebrity":{"value":"송혜교","confidence":0.631658}}]}</code>
2명	<code>{"info":{"size":{"width":223,"height":226},"faceCount":2},"faces":[{"celebrity":{"value":"송중기","confidence":0.879385}},{"celebrity":{"value":"이연희","confidence":0.173051}}]}</code>
사	<code>{"info":{"size":{"width":259,"height":195},"faceCount":0},"faces":[]}</code>

물	
2mb초과	error!!!!!! responseCode= 400 java.io.IOException: Server returned HTTP response code: 400 for URL:
json	<pre>{ "info":{"size":{"width":264,"height":200}, "faceCount":1}, "faces":[{"celebrity":{"value":"송혜교","confidence":0.631658}}] }</pre> <p>info변수 info.size.width-이미지사진가로크기 info.size.height-이미지사진세로크기 info.faceCount-얼굴 수 faces변수 faces[0]. celebrity – 유명인이름 faces[0]. confidence-뵈은정도(0-1)</p>

<오류>

400	2mb 초과 파일크기 / 파라미터명 image 아니다
404	apiurl 확인
401	client key, client secret 확인
FileNotFoundException	파일경로명이나 파일명 확인

2. 스프링부트 실행

0.mynaverai라는 이름의 실습용 스프링부트 프로젝트를 생성한다.

- 프로젝트name : mynaverai

- 자바버전 8
- 패키지명 edu.spring.naverai
- 항목은 spring web, oracle driver, mybatis framework, jdbc api 를 선택한다.

1. images폴더의 이미지파일들만 압축 공유.
2. naver ai cfr 콘솔 개발자 가이드에서 키, 암호, url 확인한다.
3. naver ai cfr 콘솔 개발자 가이드에서 자바 샘플 예제 복사하여 현재 프로젝트의 crfr 패키지 생성 - APIExamFace.java 파일 만들어 붙여넣는다.
4. 이미지파일명 부분 수정, 키, 암호 추가한다.
5. run as - java application으로 실행한다

<오류코드>

Bad Request error!!!!!! responseCode= 400	2mb 초과 파일인지 확인 image 파라미터명 확인
Unauthorized error!!!!!! responseCode= 401	인증 키와 암호 확인 후 수정 naver console에서 해당 application의 인증 정보 확인
Not Found error!!!!!! responseCode= 404	apiurl 확인
java.io.FileNotFoundException:	이미지파일의 경로 및 파일명 확인

<결과분석>

<pre> {"info": {"size":{"width":264,"height":200},"faceCount":1}, "faces":[{"celebrity":{"value":"송혜교","confidence":0.7217 62}}] } </pre>	<p>info:전체결과객체변수</p> <p>info.size: 이미지파일크기변수</p> <p>info.size.width:가로크기</p> <p>info.size.height:세로크기</p> <p>info.faceCount:발견된 얼굴 수</p> <p>faces:얼굴정보저장객체변수</p> <p>faces[0].celebrity: 닮은 유명인정보변수</p> <p>faces[0].celebrity.value:유명인이름</p>
---	--

	faces[0].celebrity.confidence:낮은확률(최대 1)
--	--

7. 2명 이상의 사람 얼굴이 보이는 이미지사진으로 변경 테스트

String imgFile = "C:/Users/user/Desktop/images/bangtan7.jpg";

```
{
  "info": {
    "size": {
      "width": 640,
      "height": 407,
      "faceCount": 7
    },
    "faces": [
      {
        "celebrity": {
          "value": "유겸",
          "confidence": 0.01
        }
      },
      {
        "celebrity": {
          "value": "엄현경",
          "confidence": 0.0940699
        }
      },
      {
        "celebrity": {
          "value": "유진",
          "confidence": 0.0625277
        }
      },
      {
        "celebrity": {
          "value": "현승민",
          "confidence": 0.154775
        }
      },
      {
        "celebrity": {
          "value": "정은지",
          "confidence": 0.34685
        }
      },
      {
        "celebrity": {
          "value": "김태형",
          "confidence": 0.01
        }
      },
      {
        "celebrity": {
          "value": "이일민",
          "confidence": 0.0777358
        }
      }
    ]
  }
}
```

6. mvc 구조로 변경하여 브라우저에서 이미지 전달받아서 테스트

6-0. pom.xml 에 jsp 사용 위한 dependency 추가

```
<!-- for jsp -->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>
<!-- for jsp jstl -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
```

```
</dependency>
<!--for tomcat restart -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <optional>true</optional>
    </dependency>
```

6-1.application.properties

```
server.port=9001
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
spring.devtools.livereload.enabled=true
```

6-2.WebConfig.java 생성

```
@Configuration
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/naverimages/**")
            .addResourceLocations("file:///C:/Users/user/Desktop/images/");
    }
}
```

6-3.MyNaveraiapplication.java 수정

```
@SpringBootApplication
@ComponentScan
@ComponentScan(basePackageClasses = NaverFaceController.class)
public class MynaveraiApplication {

    public static void main(String[] args) {
        SpringApplication.run(MynaveraiApplication.class, args);
    }
}
```



```

    }
}

```

6-4. mvc(cfr 패키지)

NaverFaceController	"faceinput" "face"
NaverService	edu.spring.naverai패키지 인터페이스 public interface NaverService { String test(String image); }
NaverFaceService	NaverService 상속 test 메소드 내에 APIExamFace.java의 main 메소드 내용을 복사 키, 암호, url 그대로. 이미지파일명 매개변수로 전달받도록 수정 . 리턴값은 네이버ai서버의 응답으로 리턴
faceinput.jsp	images폴더의 파일들을 보여주는 뷰 이 중 파일 1개 클릭하면 파일명만 전송
face.jsp	클릭한 이미지의 파일명 받아서 유명한 닳은꼴 분석 결과 보여주는 뷰
WebConfig.java	edu.spring.naverai패키지 이미지파일들을 저장한 외부 경로(현재프로젝트외부)를 스프링부트에서 url로 매핑
pom.xml	json 데이터 parsing을 위한 라이브러리 다운로드 <dependency> <groupId>org.json</groupId> <artifactId>json</artifactId> <version>20201115</version> </dependency>
	JSONArray

	JSONObject
--	------------

<실습 결과 확인하기>

kimbab.jpg -> 이미지 파일 2mb 초과 오류

사물..->...

1명 사진->...

여러명 사진->...

사진에 몇명 있는지 출력

사진 크기 출력

NaverFaceController	/face – String image 메소드 완료 /faceinput 메소드 바탕화면Wimages 폴더 모든 이미지 파일 이름을 모델로 생성 faceinput.jsp 뷰로 전달
/cfr/faceinput.jsp	song.jpg 링크-<a> 이미지 출력 - ==> 링크 클릭하면 /face 이동하면서 image 파라미터에는 파일명 전달 bangtan7.jpg
face.jsp	완료

6-5. 얼굴 인식 서비스

입력받은 이미지로부터 얼굴을 감지하고 입력된 이미지에서 얼마나 많은 얼굴이 감지되었고 각 얼굴이 어디에 어떤 크기로 위치하며 어떤 모습을 하고 있는지 반환하는 REST API 입니다. 이미지에서 다음과 같은 정보를 분석합니다.

- 감지된 얼굴의 수
- 감지된 각 얼굴을 분석한 정보
 - 감지된 각 얼굴의 좌표 및 크기
 - 감지된 각 얼굴의 눈, 코, 입의 좌표
 - 감지된 얼굴의 추정 성별 및 추정치
 - 감지된 얼굴의 추정 나이 및 추정치
 - 감지된 얼굴에서 분석된 감정
 - 감지된 얼굴의 방향

요청

POST <https://naveropenapi.apigw.ntruss.com/vision/v1/face>

NaverFaceService2	NaverService 상속 test 메소드 내에 APIExamFace.java의 main 메소드 내용을 복사 키, 암호, url 그대로. 이미지파일명 매개변수로 전달받도록 수정 . 리턴값은 네이버ai서버의 응답으로 리턴
faceinput2.jsp	images폴더의 파일들을 보여주는 뷰 이 중 파일 1개 클릭하면 파일명만 전송 html 태그의 url 부분만 변경
face2.jsp	클릭한 이미지의 파일명 받아서 얼굴 위치 눈코입 위치 보여주는 뷰
얼굴인식시 jquery-3.2.1.min.js 파일 복사하여 static폴더에 넣기 canvas 태그 사용	
<pre> { "info":{"size":{"width":264,"height":200},"faceCount":1}, "faces":[{ "roi":{"x":80,"y":45,"width":61,"height":61}, </pre>	

```
"landmark":null,
"gender":{"value":"female","confidence":0.999912},
"age":{"value":"24~28","confidence":0.856547},
"emotion":{"value":"neutral","confidence":0.999967},
"pose":{"value":"right_face","confidence":0.61542}
}
]
}
```

```
{
"info":{"size":{"width":264,"height":200}
,"faceCount":1},
"faces":[
{"roi":{"x":80,"y":45,"width":61,"height":61},
"landmark":null,
"gender":{"value":"female","confidence":0.999912},
"age":{"value":"24~28","confidence":0.856547},
"emotion":{"value":"neutral","confidence":0.999967},
"pose":{"value":"right_face","confidence":0.61542}}
]
}
```

