

基于人脸识别的登录与注册_任务卡

任务名称：基于人脸识别的一键登录与注册

任务目标：通过java开发人脸识别方式的项目的登录与注册模块。

任务概述：某大厂程序员小王，早上 10：30 来到工作岗位，打开电脑准备编写代码，板凳还没暖热就被技术经理喊到了小黑屋。

小王进入这家企业已经 3 月有余了，在这三个多月里，目睹了很多应届生同事 因为 BUG 问题被领导训斥甚至辞退，而同样是今年应届生的小王，因为之前拜了阿里的一个大牛为师，工作的过程中一直有师父可以请教，所以在工作中收获了很多的掌声。面对小黑屋中表情严肃的技术经理，小王脑海中出现了一段话“难道 我终于迎来了人生中的第一个重大 BUG？唉，莫装逼，果然遭雷劈啊！”。

还好事不是小王想的那样，原来是技术经理鉴于小王 一直以来的优秀表现，准备让小王完成已有项目的一个登录注册功能的迭代需求（原有项目只有用户名密码登录注册），现在准备添加一个人脸识别登录与注册的入口，同时为小王配备1名前端开发小伙伴负责页面的编写。第一版要求先完成人脸登录注册的基本功能，后续再完整接入项目中。

1、概念

1.1 前沿生态-图像识别技术

图像识别技术是人工智能的一个重要领域。它是指对图像进行对象识别，以识别各种不同模式的目标和对像的技术。

图像识别技术是立体视觉、运动分析、数据融合等实用技术的基础，在导航、地图与地形配准、自然资源分析、天气预报、环境监测、生理病变研究等许多领域重要的应用价值。

1.2 人脸 face_token

人脸（Face）在人脸识别技术中特指图像中发现的人脸，当对一张图片进行人脸检测时，会将检测到的人脸记录下来，包括人脸在图片中的位置，用一个系统标识face_token来表示。

注意：对同一张图片进行多次人脸检测，对同一个人脸会得到不同的face_token。

1.3 人脸集合 faceSet

人脸集合（FaceSet）是用来存储检测到人脸的存储对象。一个FaceSet内face_token是不重复的。

1.4 人脸搜索

计算机检测到图片中一个人脸之后，通过人脸判断人身份的过程被称为人脸搜索。

人脸搜索是指采集用户新的人脸，在多个已知身份用户的人脸集合中进行搜索，找出新的人脸属于哪一个已知身份用户。人脸搜索需要调用Search API。

2 Web API

2.1 注册账号

首先，您需要有一个账号。

- 如果您已经注册了账号，请忽略这一步。
- 如果您还没有注册，可以在 [Face++ 官网](#) 点击页面右上角的“注册”，在注册页面根据引导，完成注册。注册时可选择使用短信验证或是邮箱验证。

中 | EN

控制台

注册

欢迎使用Face++用户控制台

短信注册

邮箱注册

请输入用户名

+86 请输入手机号

验证码

获取验证码

请输入密码

☐ 我已阅读并同意 [《服务协议》](#) 和 [隐私政策](#)

注册

 微信注册

已有账号？ [立即登录](#)



注册成功后，您需要选择开发者类型并补充对应的信息：

欢迎使用Face++用户控制台

* 开发者类型：

☒ 个人开发者 ☐ 公司

* 真实姓名：

行业：

请选择

地区：

请选择

开始体验



2.2 登录账号

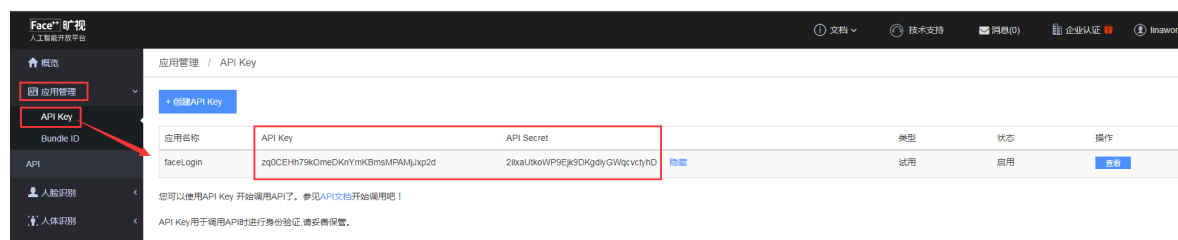
输入刚刚注册的账号密码进行登录。



2.3 查看自己的API key

要调用 API，需要先创建一个 API Key（API 密钥），它是使用 API 和 SDK 的凭证。

登录成功进入到如下页面：按照红色框提示查看自己的密钥（注册成功之后都有一个创建的应用，应用名称可以改为自定义的名称，API KEY 和 API SECRET 是自动生成，不可以更改，我们直接使用就可以）：



注：免费API key 可依据免费规则调用API，如果您希望使用付费版服务，可以点击图中的蓝色按钮创建新的应用（这里不再介绍）。

2.4 完成

好了，所有需要的操作都已经完成，我们现在可以跟着 [API 文档](#)，进行接入了。

3 API接口调用

3.1 控制台在线演示



3.2 postman调用API接口

参考文档设置postman参数调用api接口

文档中心 / Detect API

调用方法

POST

请求体格式

multipart/form-data

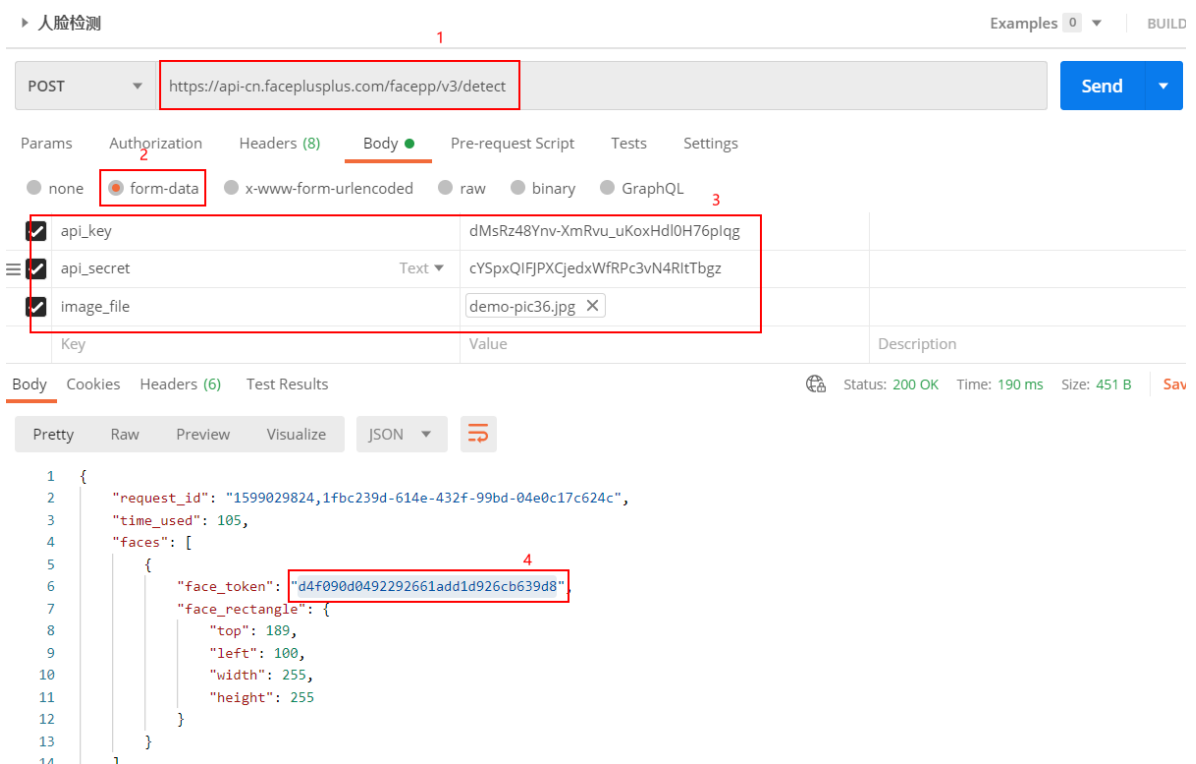
权限

所有 API Key 都可以调用本 API。其中 calculate_all 和 face_rectangle 参数只有正式 API Key 才能使用，试用 API Key 无法使用。

请求参数

是否必选	参数名	类型	参数说明
必选	api_key	String	调用此API的API Key
必选	api_secret	String	调用此API的API Secret
必选（三选一）	image_url	String	图片的 URL。 注：在下载图片时可能由于网络等原因导致下载图片时间过长，建议使用 image_file 或 image_base64 参数直接上传图片。

3.2.1 人脸检测



3.2.1 人脸搜索

POSTsearch-人脸搜索

https://api-cn.faceplusplus.com/facepp/v3/search

SendSave

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTests

noneform-datax-www-form-urlencodedrawbinaryGraphQL BETA

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> api_key	zqQCEHh79kOmeDKnYmKBmsMPAMJxp2d	
<input checked="" type="checkbox"/> api_secret	2lixUtkoWP9Ejg9DKgdyGWqcvcyhD	
<input checked="" type="checkbox"/> outer_id	myFaceSet	
<input checked="" type="checkbox"/> face_token	60d14cddd806069be8401d0b7dfc6b0	
Key	Value	Description

BodyCookiesHeaders (4)Test Results

Status: 200 OKTime: 999 msSize: 307 BSaveDownload

PrettyRawPreviewJSON

```
1 {
2   "request_id": "1599738700,47b94c33-e7dd-4299-8bf2-c9fb3a10ed2f",
3   "time_used": 574,
4   "thresholds": {
5     "face_val": 80.332,
6     "face_sim": 73.975,
7     "face_sim": 69.101
8   },
9   "results": [
10    {
11      "confidence": 38.918,
12      "user_id": "",
13      "face_token": "2808582a6c0af80d3c04fac71180cf43"
14    }
15  ]
16 }
```

3.2.3 创建faceset

创建faceset

Examples0BUILD

POSThttps://api-cn.faceplusplus.com/facepp/v3/faceset/create

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> api_key	dMsRz48Ynv-XmRvu_uKoxHdl0H76plqg	
<input checked="" type="checkbox"/> api_secret	cYSpxQIFJPXCjedxWfRPc3vN4RItTbgz	
<input checked="" type="checkbox"/> outer_id	travelLoginFaceSet	
Key	Value	Description

3.2.4 添加face到faceset

添加face到faceset

Examples0BUILD

POSThttps://api-cn.faceplusplus.com/facepp/v3/faceset/addface

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> api_key	dMsRz48Ynv-XmRvu_uKoxHdl0H76plqg	
<input checked="" type="checkbox"/> api_secret	cYSpxQIFJPXCjedxWfRPc3vN4RItTbgz	
<input checked="" type="checkbox"/> outer_id	travelLoginFaceSet	

Response

3.2.5 根据outer_id得到faceset

POST https://api-cn.faceplusplus.com/facepp/v3/faceset/getdetail Send ▾

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

	KEY	VALUE	DESCRIPTION
☰	<input checked="" type="checkbox"/> api_key	dMsRz48Ynv-XmRvu_uKoxHdl0H76plqg	
	<input checked="" type="checkbox"/> api_secret	cYSpxQIFJPXCjedxWfRPc3vN4RitTbgz	
	<input checked="" type="checkbox"/> outer_id	travelLoginFaceSet	
	Key	Value	Description

4、登录注册功能

4.1 功能分析

4.4.1 步骤分析

```
/*
 * 注册的步骤：
 * 1、获取图片--上传到服务器指定文件夹
 * 2、判断是否包含人脸 detect API
 * 包含：
 * 在faceset中查找是否有相似度高的人脸：search API
 * 没有--表示没有注册过：添加人脸标识-facetaken到faceset,返回
true;
 * 有--表示注册过：删除照片；返回false;
 * 不包含：删除上传的图片，返回false
 */
/*
 * 登录的步骤：
 * 1、获取图片--上传到服务器指定文件夹
 * 2、判断是否包含人脸 detect API
 * 包含：
 * 在faceset中查找是否有相似度高的人脸：search API
 * 有：删除上传的照片；返回true
 * 没有：删除上传的照片；返回false
 * 不包含：删除上传的图片，返回false
 */
```

4.4.2 servlet

1、概念：

Servlet（**Server Applet**）是**Java Servlet**的简称，称为小服务程序或服务连接器，用**Java**编写的服务器端程序，具有独立于平台和协议的特性，主要功能在于交互式地浏览和生成数据，生成动态**web**内容。

2、servlet任务

- 1）读取客户端（浏览器）发送的数据----（接受请求）
- 2）处理数据并生成结果----（处理请求）
- 3）发送**HTTP**响应到客户端（浏览器）----（给出响应）

4.2 代码实现

4.2.1 创建一个web项目

详细创建过程参考：前置课程视频：web项目的创建与tomcat的配置

4.2.2 配置文件

```
API_KEY=zq0CEHh79kOmeDKnYmKBmsMPAMjJxp2d
API_SECRET=2ilxaUtkoWP9Ejk9DKgdiyGwqcvctyhD
DISPLAY_NAME=linaFaceSet
OUTER_ID=linaFaceSet
```

4.2.3 servlet代码

```
package com.wln.face.servlet;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.wln.util.FaceUtil;
import com.wln.util.ImageUtils;
@WebServlet("/faceController")
public class FaceController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /*
     * 1、接受客户端的请求 request
     * 2、处理请求
     * 3、返回数据给客户端 response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //步骤：
        //1、获取用户上传的图片--上传到指定的文件夹中upimg
        File file=ImageUtils.uploadImg(request, "imgData", "upimg");
        boolean res=false;
        boolean delFlag=true;
        try {
            //2、判断是否包含人脸信息： detect
            String faceToken=FaceUtil.detect(file);
            if(faceToken!=null){// 2.1 包含：
                // 在faceset中查找是否有相似度高的人脸信息： search
                res=FaceUtil.search(faceToken);//有：登录成功；删除照片 没有：登录失败；删除照片
            }
            //判定用户请求的类型
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        String type=request.getParameter("type");
        if("register".equals(type)) { //如果是注册
            if(res) { //有：已经注册过；删除照片
                res=false;
            } else { //没有：可以注册，添加facetoken到faceset中；保留照片
                res=FaceUtil.addFace(faceToken);
                delFlag=false; //不删除照片
            }
        }
        } else { // 2.2 不包含：登录失败；删除照片
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        //删除照片
        if(delFlag) {
            file.delete();
        }
        //返回数据给客户端 response
        PrintWriter pw=response.getWriter();
        String msg="{\"success\":\""+res+"}";
        pw.write(msg);
        pw.close();
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    doGet(request, response);
}
}

```

4.2.4 登录页面代码

前端资料--都在提供给学生资料文件中

4.3 测试
