HTTP接口

请求方法

`POST`

请求路径

`{host}/apis/vision/v1/batch_annotate_images`

请求头

`Content-Type: application/json`

请求参数

参数	类型	属性	说明
requests	JSONArray	必填	支持多图片识别（目前仅支持1张）
requests.image	JSONObject	必填	
requests.image.content	string	必填	图片二进制内容base64编码
requests.features	JSONArray	必填	识别类型
requests.features.type	int	必填	33：流程图识别
requests.image_context	JSONObject	必填	图片上下文信息
requests.image_context.language_hints	JSONArray	必填	图片中语种列表，目前只允许填写一个语种
req_id	string	选填	请求ID，方便定位问题；如果不指定，系统将自动生成，并在response中返回

支持的语种表格

语种名	语种代码
简体中文	zh-cmn-Hans
英文	en

支持的图片格式

`jpg` `png` `bmp`

一些限制

消息体的限制为30M

支持的图形类别

图形器类型	支持属性
rectangle 长方形	顶点坐标
square 正方形	顶点坐标
roundrectangle 圆角矩形	外接矩形顶点坐标
parallelogram 平行四边形	外接矩形顶点坐标
oval 椭圆	圆心坐标，横向半径，纵向半径
circle 圆	圆心坐标，横向半径，纵向半径
hexagon 六边形	外接矩形顶点坐标
diamond 菱形	外接矩形顶点坐标
others 其他	

注：多边形中除了矩形和正方形，其他所有的多边形，都是提供外接矩形的四个顶点坐标根据该多边形的类型在该外接矩形中画出对应的图形

请求体示例

```
{
  "requests": [{//数组目前仅支持1个
    "image": {
      "content": "iV5CYIXXXXXXXXXXXXXXXXXXXXXXXXI="
    },
    "features": [{
      "type": 33
    }],
    "image_context": {
      "font": 1,
      "language_hints": ["zh-cmn-Hans"]
    }
  }],
  "req_id": "custom_req_id"
```

```
}
```

正确结果响应体示例

```
// 具体查看protobuffer协议描述
// 节点id和边id没有关系，都是各自独立生成
// 边中的headid和tailid对应于节点id，用于描述两个节点的连接关系
{
    "objects": [
        {
            "texts": [
                {
                    "text": "start",
                    "bounding_poly": {
                        "vertices": [
                            {
                                "x": 73,
                                "y": 24
                            },
                            {
                                "x": 103,
                                "y": 24
                            },
                            {
                                "x": 103,
                                "y": 36
                            },
                            {
                                "x": 73,
                                "y": 36
                            }
                        ]
                    }
                }
            ],
            "box": {
                "bounding_poly": {
                    "vertices": [
                        {
                            "x": 36,
                            "y": 4
                        },
                        {
                            "x": 142,
                            "y": 4
                        },
                        {
                            "x": 142,
```

```
                        "y": 53
                    },
                    {
                        "x": 36,
                        "y": 53
                    }
                ]
            },
            "shape": "diamond"
        }
    },
    {
        "id": 1,
        "texts": [
            {
                "text": "A0",
                "bounding_poly": {
                    "vertices": [
                        {
                            "x": 32,
                            "y": 118
                        },
                        {
                            "x": 50,
                            "y": 118
                        },
                        {
                            "x": 50,
                            "y": 131
                        },
                        {
                            "x": 32,
                            "y": 131
                        }
                    ]
                }
            }
        ],
        "box": {
            "shape": "oval",
            "circle": {
                "point": {
                    "x": 40,
                    "y": 128
                },
                "rx": 36,
                "ry": 26
            }
        }
```

```
        },
        {
            "id": 2,
            "texts": [
                {
                    "text": "B0",
                    "bounding_poly": {
                        "vertices": [
                            {
                                "x": 129,
                                "y": 118
                            },
                            {
                                "x": 147,
                                "y": 118
                            },
                            {
                                "x": 146,
                                "y": 131
                            },
                            {
                                "x": 129,
                                "y": 131
                            }
                        ]
                    }
                }
            ],
            "box": {
                "shape": "oval",
                "circle": {
                    "point": {
                        "x": 137,
                        "y": 125
                    },
                    "rx": 36,
                    "ry": 26
                }
            }
        },
        {
            "id": 3,
            "texts": [
                {
                    "text": "end",
                    "bounding_poly": {
                        "vertices": [
                            {
                                "x": 76,
```

```
                            "y": 215
                        },
                        {
                            "x": 102,
                            "y": 215
                        },
                        {
                            "x": 102,
                            "y": 226
                        },
                        {
                            "x": 76,
                            "y": 226
                        }
                    ]
                }
            }
        ],
        "box": {
            "bounding_poly": {
                "vertices": [
                    {
                        "x": 64,
                        "y": 197
                    },
                    {
                        "x": 112,
                        "y": 197
                    },
                    {
                        "x": 112,
                        "y": 246
                    },
                    {
                        "x": 64,
                        "y": 246
                    }
                ]
            },
            "shape": "square"
        }
    }
],
"edges": [
    {
        "headid": 1,
        "tail": {
            "arrow_style": "none"
        },
```

```
            "head": {
                "arrow_style": "normal"
            },
            "break_points": [
                {
                    "x": 39,
                    "y": 64
                }
            ]
        },
        {
            "headid": 2,
            "tail": {
                "arrow_style": "none"
            },
            "head": {
                "arrow_style": "normal"
            }
        },
        {
            "tailid": 1,
            "headid": 3,
            "tail": {
                "arrow_style": "none"
            },
            "head": {
                "arrow_style": "normal"
            }
        },
        {
            "tailid": 2,
            "headid": 3,
            "tail": {
                "arrow_style": "none"
            },
            "head": {
                "arrow_style": "normal"
            }
        }
    ]
}
```


错误结果响应体示例

```
{
  "responses": [{
    "error": {
      "code": 13,
      "message": "some error"
    }
  }],
  "req_id": "05951b7a-e587-428b-afe7-a16c7c055846"
}
```

将来可能扩充错误码，也可能修改某个错误码的含义，所以不建议对具体的错误码进行逻辑处理，目前支持的错误码：

错误码	说明
3	参数错误
13	内部错误

返回结果中相关结构protobuf定义

```
// 流程图识别
message FlowChartResult {
    repeated FlowObject objects = 1;
    repeated FlowEdge edges = 2;
}

message FlowObject {
    int32 id = 1; // 节点id，id数值独立编码和连线id无关
    repeated TextProperties texts = 2; // 文本数组（多行文字可能是多个文本对象）
    BoxProperties box = 3;
}

message FlowEdge {
    int32 id = 1; // 连线id，独立生成和节点id无关
    int32 tailid = 2; // 线段起始对象id，为节点列表中的节点id
    int32 headid = 3; // 线段到达对象id，为节点列表中的节点id
    TopPointProperties tail = 4; // 起点属性
    TopPointProperties head = 5; // 终点属性
    repeated Vertex break_points = 6; // 线段中折点的坐标
}

message TopPointProperties {
    // 箭头样式
    // none 或者 空， 表示没有箭头
    // normal 表示普通的箭头，示例如：->
    string arrow_style = 1;
```

```
}

message TextProperties {
    string text = 1; // 文字内容
    // If it is representing a region of text, the region is a bounding box.
    // The vertices are in the order of top-left, top-right, bottom-right,
    // bottom-left.
    // For example:
    //
    //         0----1
    //         |    |
    //         3----2
    BoundingPoly bounding_poly = 2; // 文字块儿顶点坐标
}

message BoxProperties {
    // If it is representing a region of text, the region is a bounding box.
    // The vertices are in the order of top-left, top-right, bottom-right,
    // bottom-left.
    // For example:
    //
    //         0----1
    //         |    |
    //         3----2
    BoundingPoly bounding_poly = 1; // 多边形顶点坐标
    // 目前支持的图形类别如下
    // rectangle 长方形
    // square 正方形
    // roundrectangle 圆角矩形
    // parallelogram 平行四边形
    // oval 椭圆
    // circle 圆
    // hexagon 六边形
    // diamond 菱形
    // others 其他
    // tips：多边形中除了矩形和正方形，其他所有的多边形，都是提供外接矩形的四个顶点坐标
    //       根据该多边形的类型在该外接矩形中画出对应的图形
    string shape = 2; // 多边形图形类别
    Circlepp circle = 3; // 圆或者椭圆描述结构
}

// 对于圆或者椭圆，单独定义的描述结构
message Circlepp {
    Vertex point = 1; // 圆心坐标
    int32 rx = 2; // 横向半径
    int32 ry = 3; // 纵向半径
}
```

```
// A vertex represents a 2D point in the image.
// NOTE: the vertex coordinates are in the same scale as the original image.
message Vertex {
  // X coordinate.
  int32 x = 1;

  // Y coordinate.
  int32 y = 2;
}

// A bounding polygon for the detected image annotation.
message BoundingPoly {
  // The bounding polygon vertices.
  repeated Vertex vertices = 1;
}
```