

风险决策系统接入手册

概述

本文档提供同盾风险决策系统的快速接入指南，通过与您的网站快速集成，同盾将为您提供在线实时的反欺诈服务。

风险决策系统

风险决策系统是同盾的旗舰级反欺诈云服务，你可以：

- 丰富灵活的规则模板，短时间内快速部署强大的反欺诈模型
- 风险可视化管理，指标报表、实时风险地图及各类风险监控报表
- 数据分析平台，事件中心可以让你便捷的进行历史数据分析
- 高性能高吞吐，200ms内返回风险决策结果

备注：免费用户(如生物探针)后台暂不可见，风险决策规则由同盾代为配置。

风险决策API

调用方式

HTTPS方式提供在线服务，支持GET及POST的方式，输出JSON格式。

用户需要将请求的数据进行URL编码，并作为URL查询参数（GET方式）或者HTTP消息体部分（POST方式）进行传输。

测试环境

```
https://apitest.fraudmetrix.cn/riskService
```

生产环境

```
https://api.fraudmetrix.cn/riskService
```

输入字段

字段名	类型	长度	必填	字段描述
partner_code	string	32	必填	同盾分配的合作方标示

secret_key	string	32	必填	同盾分配的API密钥
event_id	string	64	必填	事件ID
token_id	string	64	可选	获取设备信息的会话标识
ip_address	string	15	必填	IP地址
id_number	string	18	可选	身份证
account_login	string	32	可选	登录账户名
account_name	string	32	可选	用户姓名
account_email	string	32	可选	邮箱
account_phone	string	15	可选	电话号码
account_mobile	string	15	可选	手机号
account_password	string	32	可选	密码摘要
transaction_id	string	64	可选	交易流水
organization	string	200	可选	工作单位
biz_license	string	15	可选	工商注册号
org_code	string	9	可选	组织机构代码
accountaddressstreet	string	256	可选	买家的街道地址信息
accountaddresscounty	string	20	可选	买家的县或镇地址信息
accountaddresscity	string	20	可选	买家的城市地址信息
accountaddressprovince	string	15	可选	买家的省份地址信息
accountaddresscountry	string	15	可选	买家的国家地址信息
accountzipcode	string	10	可选	买家的邮编信息
payee_userid	string	10	可选	卖家或收款人ID
payee_name	string	10	可选	卖家或收款人姓名
payee_email	string	10	可选	卖家或收款人邮箱
payee_mobile	string	10	可选	卖家或收款人手机
payee_phone	string	10	可选	卖家或收款人座机
payeeidnumber	string	10	可选	卖家或收款人身份证

payeecardnumber	string	10	可选	卖家或收款人银行卡号
deliver_name	string	32	可选	收货人姓名
deliver_mobile	string	15	可选	收货人手机号
deliver_phone	string	15	可选	收货人座机号
deliveraddressstreet	string	256	可选	收货人街道地址信息
deliveraddresscounty	string	20	可选	收货人县或镇地址信息
deliveraddresscity	string	20	可选	收货人城市地址信息
deliveraddressprovince	string	15	可选	收货人省份地址信息
deliveraddresscountry	string	15	可选	收货人国家地址信息
deliverzipcode	string	10	可选	收货人邮编信息
pay_id	string	64	可选	内部支付流水号
pay_method	string	32	可选	支付方式
pay_amount	double	32	可选	支付金额
pay_currency	string	32	可选	支付货币
card_number	string	32	可选	银行卡号
cc_bin	string	6	可选	BIN卡号
eventoccurtime	date	20	可选	事件发生时间
useragentcust	string	200	可选	User-Agent
refer_cust	string	200	可选	Referer

API输出示例

JSON格式，样例如下：

```
{
  "success": true,
  "final_decision": "Review",
  "final_score": 20,
  "hit_rules": [
    {
      "decision": "Reject",
      "name": "自定义规则",
      "score": 20
    }
  ],
  "seq_id": "1395296796956-4b01511c-eb9f-4f19-b819-eac4f1d5fa24",
  "spend_time": 105,
  "reason_code":""
}
```

API输出字段说明

字段名称	字段类型	是否可空	字段描述
success	boolean	否	接口调用是否成功，失败原因参见 reason_code
reason_code	string	是	调用失败原因代码，详见代码表
final_decision	string	是	风险决策的最终结果，Accept/Review/Reject
final_score	int	是	最终的风险分数，只有在权重模式下有效
hit_rules	json	是	命中的规则列表，详见下表说明
seq_id	string	是	请求的唯一序列号
spend_time	int	否	该接口调用花费的时间

参考表: 命中规则字段(hit_rules)说明

字段名称	字段类型	是否可空	字段描述
decision	string	是	该条规则的决策结果，Accept/Review/Reject
score	int	是	该条规则的风险分数，只有在权重模式下有效
name	string	是	规则名称

参考表: 原因码(reason_code):

代码	代码描述	说明
101	参数不能为空	请检查是否遗漏必填字段
102	参数类型不正确	请检查是否存在参数类型不正确的字段
103	参数超过最大长度	请检查是否存在某些字段超出字段长度限制
404	没有对应的策略配置	请检查event_id是否合法有效或对应的策略未被禁用
405	策略执行超时	内部策略执行超时
500	内部执行错误	内部接口处理错误

示例代码

一键集成

如果您的服务端应用使用Java或者PHP开发，我们提

供 原生Java 、 HttpClient-3.x 、 HttpClient-4.2.x 、 HttpClient-4.3.x 、 PHP 的API调用示例代码

后台账号开通过后，您可以在

- 测试环境 <https://portaltest.fraudmetrix.cn/integration/api.htm>
- 生产环境 <https://portal.fraudmetrix.cn/integration/api.htm>

获取可直接复制粘贴的集成代码。

以下均为 生产环境 集成代码。

Java HTTP请求示例代码

以下示例代码展示如何通过Java调用同盾提供的HTTPS API接口。

备注:本示例代码中以风险决策API调用为例, 其中API地址、合作方码等都是无效的,请在实际接入中替换成真实有效的数据。

```
public class FraudApiInvoker {
    private static final Logger logger = LoggerFactory
.getLogger(FraudApiInvoker.class);
    private static String apiUrl = "https://api
.fraudmetrix.net/riskService";
    private static PoolingHttpClientConnectionManager connMgr;
    private static CloseableHttpClient httpClient;
    private static RequestConfig requestConfig;

    static {
```

```

// 设置连接池
connMgr = new PoolingHttpClientConnectionManager();
connMgr.setMaxTotal(40);
connMgr.setDefaultMaxPerRoute(20);

// 设置超时时间1s
Builder configBuilder = RequestConfig.custom();
configBuilder.setSocketTimeout(500);
configBuilder.setConnectTimeout(500);
configBuilder.setConnectionRequestTimeout(500);
requestConfig = configBuilder.build();
}

public static FraudApiResponse invoke(List<NameValuePair> params) throws IOException {
    httpClient = HttpClients.custom().setConnectionManager(connMgr).setDefaultRequestConfig(requestConfig).build();
    HttpPost httpPost = new HttpPost(apiUrl);
    CloseableHttpResponse response = null;
    HttpEntity entity = null;
    try {
        httpPost.setConfig(requestConfig);
        httpPost.setEntity(new UrlEncodedFormEntity(params));
        response = httpClient.execute(httpPost);
        int statusCode = response.getStatusLine().getStatusCode();
        if (statusCode != HttpStatus.SC_OK) {
            logger.warn("[FraudApiInvoker] invoke failed, response status: " + statusCode);
            return null;
        }
        entity = response.getEntity();
        if (entity == null) {
            logger.warn("[FraudApiInvoker] invoke failed, response output is null!");
            return null;
        }
        String result = EntityUtils.toString(entity, "utf-8");
        return JSON.parseObject(result, FraudApiResponse.class);
    } catch (Exception e) {
        logger.error("[FraudApiInvoker] invoke throw exception, details: ", e);
    } finally {
        if (entity != null) {
            entity.getContent().close();
        }
        if (response != null) {
            response.close();
        }
        if (httpPost != null) {
            httpPost.releaseConnection();
        }
    }
}

```

```

    }
    return null;
}

public static void main(String[] args) {
    List<NameValuePair> postData = new ArrayList<NameValuePair>();
    postData.add(new BasicNameValuePair("partner_code", "demo"));
    postData.add(new BasicNameValuePair("secret_key", "***"));
    postData.add(new BasicNameValuePair("event_id", "login"));
    postData.add(new BasicNameValuePair("account_login", "your_login_i
n_name"));
    postData.add(new BasicNameValuePair("ip_address", "your_login_i
p"));
    try {
        FraudApiResponse apiResp = invoke(postData);
        System.out.println(apiResp.isSuccess());
        System.out.println(apiResp.reason_code);
    } catch (Exception e) {
    }
}
}

```

PHP HTTP请求示例代码

以下示例代码展示如何通过php调用同盾提供的HTTPS API接口。

备注:本示例代码中以风险决策API调用为例, 其中API地址、合作方码等都是无效的,请在实际接入中替换成真实有效的数据。

```

<?php

function invoke_fraud_api(array $params) {
    $api_url = "https://api.fraudmetrix.net/riskService";
    $options = array(
        CURLOPT_POST => 1,           // 请求方式为POST
        CURLOPT_URL => $api_url,     // 请求URL
        CURLOPT_RETURNTRANSFER => 1, // 获取请求结果
        CURLOPT_TIMEOUT_MS => 500,   // 超时时间(ms)
        // -----请确保启用以下两行配置-----
        CURLOPT_SSL_VERIFYPEER => 1, // 验证证书
        CURLOPT_SSL_VERIFYHOST => 2, // 验证主机名
        // -----否则会被窃听的风险-----
        CURLOPT_POSTFIELDS => http_build_query($params) // 注入接口参数
    );
    $ch = curl_init();
    curl_setopt_array($ch, $options);
    if(!($response = curl_exec($ch))) {
        // 错误处理, 按照同盾接口格式fake调用结果
        return array(

```

```

        "success" => false,
        "reason_code" => "000:调用API时发生错误 [ ".curl_error($ch)." ]"
    );
}
curl_close($ch);
return json_decode($response, true);
}

// 准备接口参数
$data = array(
    "partner_code" => "demo",
    "secret_key" => "xxx",
    "event_id" => "login",
    "account_login" => "your_login_name",
    "ip_address" => "your_login_ip"
);
// 调用接口
$result = invoke_fraud_api($data);

// 换行符， 如果是通过http访问本页面，则换行为<br/>,
//      如果是通过命令行执行此脚本，换行符为\n
$separator = PHP_SAPI == "cli" ? "\n" : "<br/>";

// 打印调用结果
echo "接口调用结果: $separator";
print_r($result);

echo "调用成功: " . ($result["success"] ? "true" : "false") . $separator;
if($result["success"]) {
    echo "决策结果: " . $result["final_decision"] . $separator;
} else {
    echo "失败原因: " . $result["reason_code"] . $separator;
}

/* ----- 示例输出 -----
接口调用结果:
Array
(
    [final_decision] => Accept // 最终的风险决策结果
    [final_score] => 0 // 风险分数
    [policy_name] => 登录策略 // 策略名称
    [seq_id] => xxx // 请求序列号，全局唯一
    [spend_time] => 9 // 花费的时间，单位ms
    [success] => 1 // 执行是否成功，不成功时对应reason_code
)
调用成功: true
决策结果: Accept
----- 示例输出 ----- */

?>

```


