Documentation for Anti-Fraud Implementation

Summary

Version control

Name	Date	Description	Version
Adiq	10/13/2020	Document Creation	1.0
Adiq	08/18/2021	Adaptation of the mandatory fields; Anti-Fraud behavior when the customer also uses 3DS.	1.0.1
Adiq	04/14/2022	The explanatory texts of the Anti-Fraud and Anti-Fraud Flowcharts have been updated for better understanding.	1.0.2
Adiq	07/28/2022	Update the request Json template with the existing Anti-Fraud Code property.	1.0.3
Adiq	12/27/2022	Adjusted Payment.ProductType options in Domains table.	1.0.4

Document link - https://developers.adiq.io/download/adiq-antifraude-1.0.4-en.pdf

Link developers - https://developers.adiq.io/manual/ecommerce#pagamento

1. Introduction

This document is the ADIQ anti-fraud implementation manual for e-commerce operations as a means of payment. The manual will help the developers understand how they can implement the solution.

Target Audience

The document has the following target audience:

- Developers of ADIQ clients who intend to implement anti-fraud.

2. About Anti-fraud

In order to minimize the fraud rate without hindering the conversion rate, the means of payment industry developed anti-fraud systems. Fraud grows at the same pace as internet sales, so many online stores become bitter with considerable losses, resulting in the closure of their operations. The challenge is to allow online stores to grow in sales and their frauds to be controlled, so it is important to understand possible fraud attempts and how to minimize them.

In e-commerce, anti-fraud plays a very important role in assessing whether transactions that pass through the online store are secure or fraudulent. The anti-fraud informs the virtual store of the result of the analysis and thus the virtual store decides whether or not to accept the transaction. Check out the key features that a good anti-fraud can offer:

- Intended for credit operations;
- Detailed information about the transaction;
- Origin of the transaction, for example: IP address;
- Checks through the neural network the important transaction data;
- Uses market rules for fraud management according to the customer profile;
- Allows the web store to create its own security rules for easy management.

The service comes at additional cost.

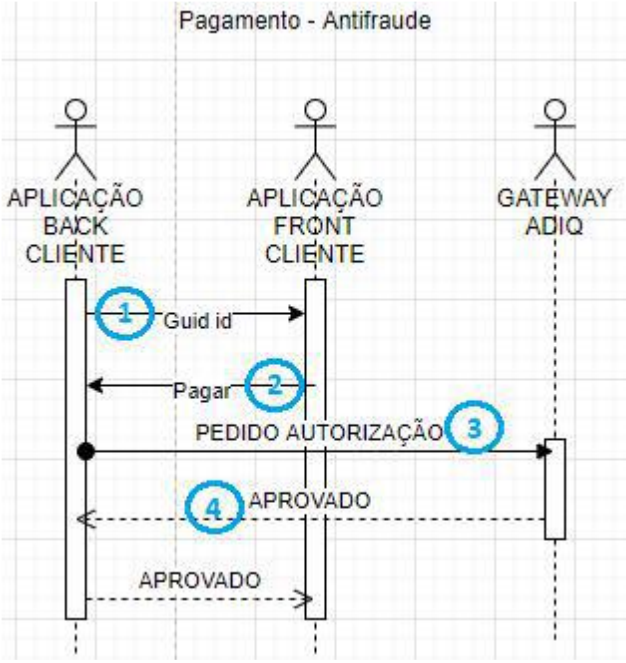
How does Anti-Fraud work in practice?

In each transaction, the anti-fraud performs more than 260 tests, check out the main ones:

- **IP Geolocation -** Identifies the IP address of the computer at the time of the transaction;
- **Device fingerprinting -** identifies the number of the computer being used;
- **Positive and negative lists -** Allows the web store to create lists with its good and bad customers;
- **Custom fields for your own data -** Allows the virtual store to register information and analysis relevant to your business model;
- **Risk detection through neural models -** allows the virtual store to use the standards established by the tool according to its branch of activity;
- **Speed Monitoring** - Allows the web store to track whether certain transaction data, such as the card number, appears in other transactions at short intervals. This can be a great indication of fraud, because in a short interval, a matter of seconds, it is unlikely that the bearer will be able to make several purchases in different locations using the same data.

Anti-fraud flowchart explained

Below we have a flowchart with a successful scenario:

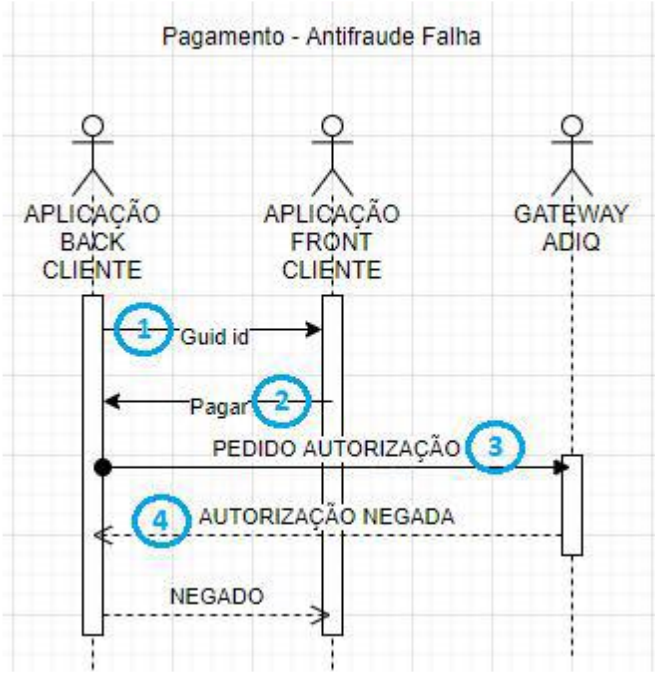


Picture breakdown:

1. Backend provides an id guid (unique identifier within 24 hours that must be generated by the user) for the front to use in capturing the fingerprint.
2. Send the "id guid" to the Backend;
3. In the “v1/payments” Gateway API, inform this data and see how to inform in the developer documentation (https://developers.adiq.io/manual/ecommerce) there is an example Json on page 13;
4. Check in the response of "v1/payments" that it will return the data of the payment executed successfully.


```
{
“paymentAuthorization”: {
“returnCode”: “00”,
“description”: “Authorized”,
“paymentid”: “02008028610304095215000000620185000000000”,
“authorizationCode”: “043711”,
“orderNumber”: “0000000001”,
“expireAt”: “2019-09-24T13:20:52.8775511-03:00”,
“amount”: 1035,
“releaseAt”: “2019-09-24T13:20:52.877545-03:00”
},
```


Below, we have a flowchart with a failure scenario:



Description of additional scenarios above:

1. Backend provides an id guid (unique identifier within 24 hours that must be generated by the user) for the front to use in capturing the fingerprint.
2. Send the "id guid" to the Backend;
3. In the “v1/payments” Gateway API, inform this data and see how to inform in the developer documentation (https://developers.adiq.io/manual/ecommerce) there is an example Json on page 13;
4. Check in the response of "v1/payments" that it will return the data of the FAILED payment.

Response Json

[{“tag” : “1003”, “description” : ”Payment was rejected by anti-fraud analysis.”}]

Anti-Fraud behavior when the customer also uses 3DS:

If the customer purchases, in addition to Anti-fraud, the 3DS service, the system behavior will be as follows:

- The 3DS service is called prior to Anti-fraud;
- Anti-fraud will only be called if the 3DS service has not been successful in credit transactions, that is, there is no need for an Anti-fraud check if the 3DS service has been successful.

3. Procedures

Anti-fraud implementation.

The implementation of anti-fraud consists of placing a JavaScript command inside the <head> and <body> tags. This command will presumably be on a page of the e-commerce website, in particular on the screen where the end customer executes the purchase, this command will be executed as soon as the page is loaded.

Requirement

- The browser must be able to run JavaScript.

Process checkpoints

- Include the JavaScript command on the payment page;
- Update codes in the back end;

Step 1 - Include the JavaScript command on the html page

Include the command below inside the <head> and <body> tags.

Add inside Header

```
<script type="text/javascript" src="https://h.online-metrix.net/fp/tags.js?org_id=1snn5n9w&session_id=adiq_br2dd470e0-698f-4ae4-bf31-71ccd33970dd">
</script>
```

Add inside Body

```
<noscript>
        <iframe style="width: 100px; height: 100px; border: 0; position:absolute top: -5000px;" src="https://h.online-metrix.net/fp/tags.js?org_id=1snn5n9w&session_id=adiq_br2dd470e0-698f-4ae4-bf31-71ccd33970dd">
        </iframe>
</noscript>
```

It must stay the same

```
<html>
    <!—HEAD —>
    <head>
        <script type="text/javascript" src="https://h.online-metrix.net/fp/tags.js?org id=1snn5n9w&session id=adiq_br2dd470e0-698f-4ae4-bf31-71ccd33970dd">
        </script>
    </head>
    <!—BODY —>
    <body>
        <noscript>
            <iframe style="width: 100px; height: 100px; border: 0; position:absolute; top: -5000px;" src="https://h.online-metrix.net/fp/tags.js?org id=1snn5n9w&session id=adiq_br2dd470e0-698f-4ae4-bf31-71ccd33970dd">
            </iframe>
        </noscript>
    </body>
</html>
```

The link in the script is as follows

```
https://h.online-metrix.net/fp/tags.js?org_id=1snn5n9w&session_id=adiq_br2dd470e0-698f-
```

```
4ae4-bf31-71ccd33970dd
```

Where:

1snn5n9w is the org_id of the approval environment:

1snn5n9w - approval

k8vif92e - production

sessionid will be the result of concatenating the adiq_br prefix, which is fixed and unchanged, with a dynamic identifier of type GUID.

Guid (globally unique identifier - GUID) is a global unique identifier, it is a 128-bit number used to identify information in computing systems, also called UUID (universally unique identifier). Although the probability of a GUID being duplicated is not zero, it is close enough to be insignificant.

2dd470e0-698f-4ae4-bf31-71ccd33970dd is an example of a GUID generated by the merchant application, that is, it must be generated on the server of the client that is implementing the Anti-fraud solution.

To learn more about generating a GUID, click on the links below:

- Generate GUID in DotNet;
- Generate GUID in Java.

This GUID code must be specified in the authorization in the sellerInfo, on codeAntiFraud field

4. Required fields in the e-commerce gateway for Anti-fraud

There is a list of fields that are required to be sent to the Gateway when we are running Anti-fraud. These fields, according to the requisitions, are:

v1/payments (POST)

Field	Description	Type	Size	Mandatory
SellerInfo.codeAntiFraud	Anti-fraud fingerprint code. If not sent, the analysis of the anti-fraud will be done without this information.	guid	36	Yes

Customer's data

All the fields above are in the Body.

Field	Description	Type	Size	Mandatory
Customer.DocumentType	Customer's identification document type (CPF, CNPJ).	int	1	No
Customer.DocumentNumber	Customer's document number without punctuation (without mask).	string	20	No
Customer.FirstName	First name of the customer.	string	60	Yes
Customer.LastName	Last name of the customer.	string	60	Yes
Customer.Email	Customer's email.	string	255	Yes
Customer.PhoneNumber	Customer's phone (without mask).	string	15	Yes
Customer.mobilePhoneNumber	Customer's cell phone (without mask).	string	25	Yes
Customer.Address	Customer’s address.	string	60	Yes
Customer.Complement	Complement of the customer's address.	string	60	No
Customer.City	Customer’s city.	string	50	Yes
Customer.State	Customer's state.	string	20	Yes
Customer.ZipCode	Customer's zip code (without mask).	string	10	Yes
Customer.IpAddress	IP address of the customer's device.	string	48	Yes
Customer.country	Country of the customer's address.	string	2	Yes

All fields above are in the Body

ShipTo fields must be completed if the product is delivered.

Field	Description	Type	Size	Mandatory

ShipTo.FirstName	First name of the customer.	string	60	No
ShipTo.LastName	Last name of the customer.	string	60	No
ShipTo.PhoneNumber	Customer's phone (without mask).	string	15	No
ShipTo.Address	Customer’s address.	string	60	No
ShipTo.Complement	Complement of the customer's address.	string	60	No
ShipTo.City	Customer’s city.	string	50	No
ShipTo.State	Customer's state.	string	20	No
ShipTo.ZipCode	Customer's zip code (without mask).	string	10	No
ShipTo.country	Country of the customer's address.	string	2	No

All the fields above are in the Body.

The LineItems fields must be completed if there are products.

Field	Description	Type	Size	Mandatory
LineItems_#_ProductCode	Product type	string	255	No
LineItems_#_ProductSKU	Product identifier in the store	string	255	No
LineItems_#_ProductName	Product name	string	255	No
LineItems_#_Quantity	Number of products being purchased	integer	10	No
LineItems_#_UnitPrice	Product price (per item)	integer	10	No

Json v1/payments Request

```
{
    “Payment”:{
      “TransactionType”: “debit”,
      “Amount”
      “CurrencyCode”: “brl”,
      “ProductType”: “debito”,
      “Installments”: 1,
      “CaptureType”: “ac”,
      “Recurrent”:false
    },
    “CardInfo”:{
      “NumberToken”:”d56109ce-6a3b-4190-beea-1c975e8cd48 6”,
      “CardholderName”:”Luiz Silveira Neto”,
      “SecurityCode”:”123”,
      “Brand”: “mastercard”,
      “ExpirationMonth”:”01”,
      “ExpirationYear”:”29”
    },
    “SellerInfo”:{
      “OrderNumber”:”1110197548565”,
      “SoftDescriptor”:”PAG*TESTE”,
      “CodeAntiFraud":"2dd470e0-698f-4ae4-bf31-71ccd33970dd”
```

```
24.       },
25.       "Customer":{
26.         "DocumentType": "cpf",
27.         "DocumentNumber":"51115672088",
28.         "FirstName":"Luiz",
29.         "LastName":"Silveira Neto",
30.         "Email":"luiz.silveira@teste.rafael.com",
31.         "PhoneNumber":"1122542454",
32.         "MobilePhoneNumber":"11987683332",
33.         "Address":"Rua Luiz Vieira, 134",
34.         "Complement":"apto. 34 - Vila Guarani",
35.         "City":"São Paulo",
36.         "State":"SP",
37.         "ZipCode":"0987 6-098",
38.         "IpAddress":"45.233.2 32.248",
39.         "Country":"BR"
40.       },
41.       "ShipTo" : {
42.         "FirstName":"Luiz Paulo",
43.         "LastName":"Cardoso",
44.         "PhoneNumber":"1122542454",
45.         "Address":"Rua Luiz Vieira, 134",
46.         "Complement":"apto. 34 - Vila Guarani",
47.         "City":"São Paulo",
48.         "State":"SP",
49.         "ZipCode":"09876098",
50.         "Country":"BR"
51.       },
52.       "Lineltems":[
53.         {
54.             "UnitPrice":"1055",
55.             "Quantity"
56.             "ProductSKU":"922111212",
57.             "ProductName":"Cadeira de plastico",
58.             "ProductCode":"235422555252"
59.         },
60.         {
61.             "UnitPrice":"935",
62.             "Quantity":"1",
63.             "ProductSKU":"455451212",
64.             "ProductName":"Guarda-chuva",
65.             "ProductCode":"23656565644"
66.         }
67.       ] ,
68.       "Sellers":[
69.
70.       ]
71.   }
```

All the properties above are in the Body.

Domains

Ownership	Contents
Payment.TransactionType	credit, debit
Payment.CurrencyCode	brl

Payment.ProductType	avista, emissor, lojista, debito
Payment.CaptureType	ac - Authorizes and captures, pa - Pre-authorizes
Payment.Recurrent	true - Recurring, false - Not recurring
Cardinfo.Brand	visa, mastercard, amex, elo, hipercard