

1. בניית API Anti-Fraud :

חלק זה מפרט את ממשק ה-API שנבנה לצורך הכנסת רשומות לטבלה במסך הדו"חות.

הקוד ב-C#, נכתב בפרויקט Winforms, ומכין חלק XML האחראי להגדרת פורמט עסקה.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.IO;
using System.Net;

namespace anti_fraud_tester
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void CheckTransactionButton_Click(object sender, EventArgs e)
        {
            //Check transaction

            string xml = @"<?xml version=""1.0"">
<Transaction xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
    <TerminalName>0963169012</TerminalName>
    <ClientIPAddress>127.0.0.1</ClientIPAddress>
    <ServerIPAddress>127.0.0.1</ServerIPAddress>
    <CardNumber>4580458045804580</CardNumber>
    <ExpDate>1208</ExpDate>
    <Amount>100</Amount>
    <CardHolderID>029442449</CardHolderID>
    <Created></Created>
    <UserField01></UserField01>
    <UserField02></UserField02>
    <UserField03></UserField03>
    <UserField04></UserField04>
    <UserField05></UserField05>
    <UserField06></UserField06>
    <UserField07></UserField07>
    <UserField08></UserField08>
    <UserField09></UserField09>
    <UserField10></UserField10>
    <UserField11></UserField11>
    <UserField12></UserField12>
    <UserField13></UserField13>
    <UserField14></UserField14>
    <UserField15></UserField15>
    <UserField16></UserField16>
```

טלרום פתרונות מיחשוב בע"מ רחוב המלאכה 5 א.ת.ח נתניה פולג ת.ד 8575

טל: 09-8922900 פקס: 09-8922929

```
<UserField17></UserField17>
<UserField18></UserField18>
<UserField19></UserField19>
<UserField20></UserField20>
</Transaction>";

byte[] xmlBytes = Encoding.UTF8.GetBytes(xml);

HttpRequest request =
WebRequest.Create("http://antifraud.mpage.co.il/api/transactions/check") as
HttpRequest;
request.Headers.Add("username", "tester");
request.Headers.Add("password", "a123456");
request.ContentLength = xmlBytes.Length;
request.Method = "POST";
request.ContentType = "text/xml";

Stream dataStream = request.GetRequestStream();
dataStream.Write(xmlBytes, 0, xmlBytes.Length);
dataStream.Close();
WebResponse response = request.GetResponse();

Stream responseStream = response.GetResponseStream();
List<byte> bytes = new List<byte>();
int b;
do
{
    b = responseStream.ReadByte();
    bytes.Add((byte)b);
} while (b > -1);

responseStream.Close();

response.Close();

string responseXml = Encoding.UTF8.GetString(bytes.ToArray());
//TODO: parse responseXml

}

private void InsertTransactionButton_Click(object sender, EventArgs e)
{
    //Insert transaction
    string xml = @"<?xml version=""1.0""?>
<Transaction xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
    <TerminalName>0963169012</TerminalName>
    <ClientIPAddress>127.0.0.1</ClientIPAddress>
    <ServerIPAddress>127.0.0.1</ServerIPAddress>
    <CardNumber>4580458045804580</CardNumber>
    <ExpDate>1208</ExpDate>
    <Amount>100</Amount>
    <Response>0</Response>
    <ActionID>1</ActionID>
    <ErrorMessage>Valid</ErrorMessage>
    <CardHolderID>029442449</CardHolderID>
    <UserField01></UserField01>
    <UserField02></UserField02>
    <UserField03></UserField03>
    <UserField04></UserField04>
```

טלרום פתרונות מיחשוב בע"מ רחוב המלאכה 5 א.ת.ח נתניה פולג תד 8575

טל: 09-8922900 פקס: 09-8922929

```
<UserField05></UserField05>
<UserField06></UserField06>
<UserField07></UserField07>
<UserField08></UserField08>
<UserField09></UserField09>
<UserField10></UserField10>
<UserField11></UserField11>
<UserField12></UserField12>
<UserField13></UserField13>
<UserField14></UserField14>
<UserField15></UserField15>
<UserField16></UserField16>
<UserField17></UserField17>
<UserField18></UserField18>
<UserField19></UserField19>
<UserField20></UserField20>
</Transaction>";

byte[] xmlBytes = Encoding.UTF8.GetBytes(xml);

HttpRequest request =
WebRequest.Create("http://antifraud.mpage.co.il/api/transactions/insert") as
HttpRequest;
request.Headers.Add("username", "tester");
request.Headers.Add("password", "a123456");
request.ContentLength = xmlBytes.Length;
request.Method = "POST";
request.ContentType = "text/xml";

Stream dataStream = request.GetRequestStream();
dataStream.Write(xmlBytes, 0, xmlBytes.Length);
dataStream.Close();

WebResponse response = request.GetResponse();

Stream responseStream = response.GetResponseStream();
List<byte> bytes = new List<byte>();
int b;
do
{
    b = responseStream.ReadByte();
    bytes.Add((byte)b);
} while (b > -1);

responseStream.Close();

response.Close();

string responseXml = Encoding.UTF8.GetString(bytes.ToArray());
//TODO: parse responseXml

}

}
```

Anti-fraud API Documentation Ver. 1.0 04/02/14 .2

7.1 General description

The anti fraud system checks transactions according to the predefined rule and the transaction history.

The API is based upon 2 actions:

1. Check the current transaction against the history and rule defined
2. Add the current transaction to the database for the historical records and statistics

For every transaction, both actions must be taken. The check action is performed before the transaction is sent to the payment gateway. The XML's Response element will be omitted at this phase.

The insert action is performed after the transaction is sent to the payment gateway, along with the response received from the gateway.

If the check result prevents the transaction to be charged, the insert action will still be performed with the XML's Response element omitted.

7.2 Check current transaction

The check URL:

<https://secure.telepay.co.il/antifraud/API/Transactions/Check>

This URL has to be called using POST method. The request header should contain the username and password, using the header keys username and password accordingly.

The content of the request is an XML representation of the transaction object, without the Response and ActionID elements. For details refer to appendix 1.

Returned HTTP status codes

200 – OK

401 – unauthorized: username/password is invalid, or the user is not permitted for the specified terminal

7.3 Insert transaction

The insert URL:

<https://secure.telepay.co.il/antifraud/API/Transactions/Insert>

This URL has to be called using POST method. The request header should contain the username and password, using the header keys username and password accordingly.

The content of the request is an XML representation of the transaction object. The response element contains the response from the check result, or the response from Shva, if the transaction was submitted to the payment gateway. The ActionID element should contain the action ID returned from the check result. For details refer to appendix 1.

Returned HTTP status codes

201 – Created

401 – unauthorized: username/password is invalid, or the user is not permitted for the specified terminal

7.4 Appendix 1 - objects structure

Transaction

```
<?xml version="1.0"?>
<Transaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <TerminalName></TerminalName>
  <ClientIPAddress></ClientIPAddress>
  <ServerIPAddress></ServerIPAddress>
  <CardNumber></CardNumber>
  <ExpDate></ExpDate>
  <Amount></Amount>
  <Response></Response>
  <ErrorMessage></ ErrorMessage >
  <ActionID></ ActionID>
  <RuleName></ RuleName>
  <CardHolderID></CardHolderID>
  <Created ></Created>
  < UserField01></ UserField01>
  < UserField02></ UserField02>
  < UserField03></ UserField03>
  < UserField04></ UserField04>
  < UserField05></ UserField05>
  < UserField06></ UserField06>
  < UserField07></ UserField07>
  < UserField08></ UserField08>
  < UserField09></ UserField09>
  < UserField10></ UserField10>
  < UserField11></ UserField11>
  < UserField12></ UserField12>
  < UserField13></ UserField13>
  < UserField14></ UserField14>
  < UserField15></ UserField15>
  < UserField16></ UserField16>
  < UserField17></ UserField17>
  < UserField18></ UserField18>
  < UserField19></ UserField19>
  < UserField20></ UserField20>
</Transaction>
```

Field name	Description
TerminalName	The name of the terminal as received from Telerom
ClientIPAddress	The IP address of the client at the time of the transaction
ServerIPAddress	The IP address of the server at the time of the transaction (if not specified will be collected automatically)
CardNumber	The card number used to perform the transaction.
ExpDate	The card expiry date
Amount	The transaction's amount
ErrorMessage	The error message received

ActionID	The action ID taken. Possible values: 0 or null – no action taken 1 – Approved (White list) 2 – Alert 3 – Rejected
RuleName	For future use
Response	The credit company's response
CardHolderID	The ID of the cardholder
Created	The transaction's creation date. If null – the current date/time is saved
UserField01	Custom fields for additional data
UserField02	
UserField03	
UserField04	
UserField05	
UserField06	
UserField07	
UserField08	
UserField09	
UserField10	
UserField11	
UserField12	
UserField13	
UserField14	
UserField15	
UserField16	
UserField17	
UserField18	
UserField19	
UserField20	

TransactionCheckResult

```
<?xml version="1.0"?>
<TransactionCheckResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Response></Response>
  <Message></Message>
  <ActionID></ActionID>
  <RuleName></RuleName>
</TransactionCheckResult>
```

Field name	Description
Response	The result's response code. A status code of 0 (zero) represents a valid transaction. For a complete list of status codes refer to appendix 2
Message	The error message of the check result
ActionID	The action ID taken. Possible values: 0 or null – no action taken 1 – Approved (White list)

	2 – Alert 3 – Rejected
RuleName	For future use

7.5 Appendix 2 - status codes

Code	Description
0	Valid transaction
1000	The transaction's amount is out of bounds
1001	Card is in black list
1002	IP address is in black list
1010	Too many transactions from the same card
1020	Too many transactions from the same IP address
1030	The user fields do not match predefined criteria
1040	The total amount per user field is beyond predefined limit
1050	The seniority is beyond predefined limit
1060	The amount is out of predefined limit for the client's seniority

7.6 Appendix 3 - Code samples

C#

```
//Check transaction

string xml = @"<?xml version=""1.0""?>
<Transaction xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
  <TerminalName>0963169012</TerminalName>
  <ClientIPAddress>127.0.0.1</ClientIPAddress>
  <ServerIPAddress>127.0.0.1</ServerIPAddress>
  <CardNumber>4580458045804580</CardNumber>
  <ExpDate>1208</ExpDate>
  <Amount>100</Amount>
  <CardHolderID>029442449</CardHolderID>
  <Created></Created>
  <UserField01></UserField01>
  <UserField02></UserField02>
  <UserField03></UserField03>
  <UserField04></UserField04>
  <UserField05></UserField05>
  <UserField06></UserField06>
  <UserField07></UserField07>
  <UserField08></UserField08>
  <UserField09></UserField09>
  <UserField10></UserField10>
  <UserField11></UserField11>
  <UserField12></UserField12>
  <UserField13></UserField13>
  <UserField14></UserField14>
  <UserField15></UserField15>
  <UserField16></UserField16>
  <UserField17></UserField17>
  <UserField18></UserField18>
  <UserField19></UserField19>
  <UserField20></UserField20>
</Transaction>";

byte[] xmlBytes = Encoding.UTF8.GetBytes(xml);

HttpRequest request =
WebRequest.Create("https://secure.telepay.co.il/AntiFraud/api/transactions/check") as HttpRequest;
request.Headers.Add("username", "test");
request.Headers.Add("password", "a123456");
request.ContentLength = xmlBytes.Length;
request.Method = "POST";
request.ContentType = "text/xml";

Stream dataStream = request.GetRequestStream();
dataStream.Write(xmlBytes, 0, xmlBytes.Length);
dataStream.Close();
WebResponse response = request.GetResponse();

Stream responseStream = response.GetResponseStream();
List<byte> bytes = new List<byte>();
int b;
do
```

טלרום פתרונות מיחשוב בע"מ רחוב המלאכה 5 א.ת.ח נתניה פולג ת.ד 8575

טל: 09-8922900 פקס: 09-8922929

```
{
    b = responseStream.ReadByte();
    bytes.Add((byte)b);
} while (b > -1);

responseStream.Close();

response.Close();

string responseXml = Encoding.UTF8.GetString(bytes.ToArray());
//TODO: parse responseXml

//Insert transaction
string xml = @"<?xml version=""1.0""?>
<Transaction xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
xmlns:xsd=""http://www.w3.org/2001/XMLSchema"">
    <TerminalName>0963169012</TerminalName>
    <ClientIPAddress>127.0.0.1</ClientIPAddress>
    <ServerIPAddress>127.0.0.1</ServerIPAddress>
    <CardNumber>4580458045804580</CardNumber>
    <ExpDate>1208</ExpDate>
    <Amount>100</Amount>
    <Response>0</Response>
    <ActionID>1</ActionID>
    <ErrorMessage>Valid</ErrorMessage>
    <CardHolderID>029442449</CardHolderID>
    <UserField01></UserField01>
    <UserField02></UserField02>
    <UserField03></UserField03>
    <UserField04></UserField04>
    <UserField05></UserField05>
    <UserField06></UserField06>
    <UserField07></UserField07>
    <UserField08></UserField08>
    <UserField09></UserField09>
    <UserField10></UserField10>
    <UserField11></UserField11>
    <UserField12></UserField12>
    <UserField13></UserField13>
    <UserField14></UserField14>
    <UserField15></UserField15>
    <UserField16></UserField16>
    <UserField17></UserField17>
    <UserField18></UserField18>
    <UserField19></UserField19>
    <UserField20></UserField20>
</Transaction>";

byte[] xmlBytes = Encoding.UTF8.GetBytes(xml);

HttpRequest request =
WebRequest.Create("https://secure.telepay.co.il/AntiFraud/api/transactions/inse
rt") as HttpRequest;
request.Headers.Add("username", "test");
request.Headers.Add("password", "a123456");
request.ContentLength = xmlBytes.Length;
request.Method = "POST";
request.ContentType = "text/xml";
```

טלרום פתרונות מיחשוב בע"מ רחוב המלאכה 5 א.ת.ח נתניה פולג ת.ד 8575

טל: 09-8922900 פקס: 09-8922929

```
Stream dataStream = request.GetRequestStream();
dataStream.Write(xmlBytes, 0, xmlBytes.Length);
dataStream.Close();

WebResponse response = request.GetResponse();

Stream responseStream = response.GetResponseStream();
List<byte> bytes = new List<byte>();
int b;
do
{
    b = responseStream.ReadByte();
    bytes.Add((byte)b);
} while (b > -1);

responseStream.Close();

response.Close();

string responseXml = Encoding.UTF8.GetString(bytes.ToArray());
//TODO: parse responseXml
```