|  |
| --- |
| United Financial Corporation |
| Merchant Integration Technical Manual |
| გაერთიანებული საფინანსო კორპორაცია |

Contents

[**მერჩანტის რეგისტრაცია საპროცესინგოში (UFC)** 3](#_Toc497473338)

[**კავშირის შემოწმება** 4](#_Toc497473339)

[**სერთიფიკატის კონვერტაცია** 4](#_Toc497473340)

[**1.1.1 ტრანზაქციის აიდის დაგენერირება** 5](#_Toc497473341)

[**1.1.3 ტრანზაქციის სტატუსი(წარმატებული, წარუმატებელი).** 10](#_Toc497473342)

[**1.1.4 დღის დახურვის ოპერაცია** 11](#_Toc497473343)

[2**.1.1 თანხის ჩამოჭრა ანუ ტრანზაქციის რეგისტრაცია** 12](#_Toc497473344)

[**2.1.2    პრე-ავტორიზაცია.** 12](#_Toc497473345)

[**2.1.3 პრეავტორიზაციის კომიტი(ავტორიზაცია, დადასტურება)** 13](#_Toc497473346)

[**2.1. 4    ტრანზაქციის რევერსალი** 15](#_Toc497473347)

[**2.1.5 ტრანზაქციის რეფანდი** 16](#_Toc497473348)

[**2.2.1 რეგულარული გადახდები** 17](#_Toc497473349)

[**2.2.2 ბარათის რეგისტრაცია** 18](#_Toc497473350)

[**2.2.3 თანხის ჩამოჭრა(Execution)** 19](#_Toc497473351)

[**2.2.4 კრედიტული ოპერაციები(original credit)** 20](#_Toc497473352)

##  **მერჩანტის რეგისტრაცია საპროცესინგოში (UFC)**

ნებისმიერი ორაგანიზაცია, რომელსაც სურს, რომ ისარგებლოს ელექტრონული კომერციის გადახდით, მიმართავს ბანკს და შემდგომში ბანკთან გაფორმებული ხელშეკრულების საფუძველზე, ბანკი საპროცესინგოში აგზავნის მერჩანტის რეგისტრაციის დაკვეთას.

მერჩანტის სისტემაში რეგისტრაციისათვის და სერთიფიკატის დაგენერირებისათვის საპროცესინგოსათვის (UFC) საჭიროა შემდეგი სახის ინფორმაცია:

1. შესაბამისი ვებ-ჰოსტინგის ან სერვერის გამომავალი IP მისამართი საიდანაც უნდა მოხდეს მომართვა საპროცესინგოს სერვერთან, რათა მოხდეს საპროცესინგოს მხარეს შესაბამისი IP მისამართზე დაშვება.
2. OK URL - სადაც მომხმარებელი გადამისამართდება გადახდის შესრულების შემდგომ.
3. FAIL URL - მისამართი სადაც მომხმარებელი გადამისამართდება *მხოლოდ* ***სისტემური შეცდომის*** *დროს*, სხვა დანარჩენ შემთხვევაში მომხმარებელი გადამისამართდება OK URL-ზე.

მოთხოვნილი ინფორმაციის მიწოდების შემდეგ, UFC(საპროცესინგო) არეგისტრირებს შესაბამის მერჩანტს სისტემაში და მერჩანტის მიერ მითითებულ ელ-ფოსტაზე აგზავნის სერთიფიკატს p12 ფორმატში, რომლის კონვერტაცია უნდა მოხდეს pem ფორმატში მერჩანტის მხარეს და რომლის მეშვეობით მერჩანტს შეეძლება კავშირზე გასვლა UFC (საპროცესინგოს) სერვერთან და ოპერაციების შესრულება.

* ხშირად დასმული კითხვა: **რატომ არ ვაგზავნით სერთიფიკატს PEM ფორმატში.**

მიზეზი არის უსაფრთხოება, ანუ P12 ფორმატს აქვს ჩვენს მიერ (UFC) მინიჭებული პაროლი, რომელიც უნდა შეიცვალოს და შეცვლა კი უნდა განახორციელოს მხოლოდ მერჩანტმა და საბოლოო პაროლიც მან უნდა დაადოს PEM სერთიფიკატს. (PCI DSS მოთხოვნა: https://en.wikipedia.org/wiki/Payment\_Card\_Industry\_Data\_Security\_Standard)

 მერჩანტის მხარეს ინფორმაციის ცვლილების შემთხვევაში (IP, OK URL, FAIL URL), ცვლილება უნდა მოხდეს ასევე საპროცესინგოს მხარეს, ამისათვის აუცილებელია ახალი პარამეტრები მიწერილ იქნას შესაბამის მეილზე (**ბანკთან შეთანხმებით**), რათა ასევე UFC\_ის მხარეს მოხდეს ინფორმაციის ცვლილება (ინფორმაციის ცვლილებისას ახალი სერთიფიკატი საჭირო არ არის).

### კავშირის შემოწმება

რეკომენდირებულია ინტეგრაცია დაიწყოს კავშირის შემოწმებით, ეს პროცედურა გაამარტივებს მერჩანტის ინტეგრაციას და მოახდენს შესაძლო გართულებების თავიდან აცილებას.

 კავშირის შემოწმება შესაძლებელია სერვერიდან TELNET-ის გამოშვებით. დადებითი პასუხის დაბრუნება ნიშნავს რომ კავშირი არსებობს, ხოლო უარყოფითი პასუხის შემთხვევაში მერჩანტმა უნდა გადაამოწმოს სერვერის გამომავალი IP და სერვერზე არსებული Firewall\_ის კონფიგურაცია 18443 პორტზე.

TELNET-ის გამოშვება უნდა მოხდეს ამ ჰოსტზე:

ecommerce.ufc.ge 18443

ტელნეტის გამოშვებაში შესაძლოა დაგეხმაროთ ჩვენს მიერ გადმოგზავნილი telnet.php ფაილი ან სერვერის/ვებ-ჰოსტინგის საპორტი.

**სერთიფიკატის კონვერტაცია** უსაფრთხოების პირობების გათვალისწინებით, სერთიფიკატის p12 დან pem ფორმატში კონვერტაცია უნდა მოხდეს მერჩანტის მხარეს, ეს შესაძლებელია განხორციელდეს Openssl ბიბლიოთების მეშვეობით, რომელსაც არის UFC\_ის მიერ გადმოგზავნილ არქივში ან შესაძლებელია მისი ინტერნეტში მოძიება.

 სერთიფიკატის კონვერტაციისათვის ბიბლიოთეკის მეშვეობით უნდა გაეშვას შემდეგი ბრძანება:

openssl pkcs12 -in \*.p12 -out \*.pem

სადაც \*.p12 შემავალი, დასაკონვერტირებელი ფაილია ხოლო \*.pem დაკონვერტირებული უკვე შესაბამის დირექტორიაში(თუ გზას არ მიუთითებთ pem ფორმატის სერთიფიკატი შეიქმნება Openssl გამშვები ფაილის საქაღალდეში).

კონვერტაციისას ბრძანების გაშვების შემდგომ პროგრამა მოითხოვს სერთიფიკატის პაროლს, რომელიც გადმოგზავნილი უნდა იყოს UFC\_მიერ სერთიფიკატის გადმოგზავნის პარალელურად.



ამ მაგალითში pass phrase-ად შეყვანილია იგივე პაროლი, რომელიც გადმოგზავნილია UFC\_ის მიერ

**ტრანზაქციის შესრულების ზოგადი აღწერა**

 ტრანზაქციის შესასრულებლად პირველ რიგში უნდა მოხდეს ტრანზაქციის აიდის დაგენერირება.

 ამისათვის უნდა მოხდეს შესაბამისი ბრძანებით და პარამეტრებით (თანხა, ვალუტა, ტრანზაქციის ტიპი და სხვა) ამ ლინკზე მომართვა [https://ecommerce.ufc.ge:18443/ecomm2/MerchantHandler](https://securepay.ufc.ge:18443/ecomm2/MerchantHandler) რის შედეგადაც საიტს უბრუნდება ტრანზაქციის აიდი, ხოლო გადახდის გვერდზე გადმოსასვლელად დაგენერირებული ტრანზაქციის აიდი უნდა დაიპოსტოს შემდეგ მისამართზე [https://ecommerce.ufc.ge/ecomm2/ClientHandler](https://securepay.ufc.ge/ecomm2/ClientHandler)

## **1.1.1 ტრანზაქციის აიდის დაგენერირება**

ტრანზაქციის აიდი წარმოადგენს 28 სიმბოლოიან string ტიპის მონაცემს, რომელიც დაგენერირებულია UFC\_ის სისტემის მიერ და რომელიც არის უნიკალური თითოეული ტრანზაქციისათვის, ტრანზაქციის აიდით შესაძლებელია ტრანზაქციის იდენთიფიცირება, ამიტომ მკაცრად რეკომენდირებულია რომ თითოეული ტრანზაქციის აიდი შენახული იქნას ასევე მერჩანტის მხარეს, შესაძლო პრობლემის შემთხვევაში მისი სამომავლო იდენთიფიკაციისათვის.

როგორც ზემოთ აღვნიშნეთ ტრანზაქციის აიდის დასაგენირებლად მერჩანტმა უნდა გამოაგზავნოს მოთხოვნა შემდეგ მისამართზე POST მეთოდით:

[https://ecommerce.ufc.ge:18443/ecomm2/MerchantHandler](https://securepay.ufc.ge:18443/ecomm2/MerchantHandler)

მოთხოვნის მაგალითი:

command=<type>&amount=<amount>&currency=<currency>&client\_ip\_addr=<ip>&description=<desc>&language=<language>&msg\_type=SMS

სადაც command პარამეტრი წარმოადგენს ტრანზაქციის ტიპს, amount პარამეტრი ტრანზაქციის თანხას, currency ტრანზაქციის ვალუტას, client\_ip\_addr მომხმარებლის IP მისამართს, description ტრანზაქციის აღწერას, ხოლო language პარამეტში გადმოცემული მნიშვნელობის მიხედვით ჩაიტვირთება გადახდის ტემპლეიტი შესაბამის ენაზე(EN ან GE)

Description - უნდა შეიცავდეს მხოლოდ ინგლისურ ანბანს, ციფრებს და სიმბოლოებს. სისტემა არ იღებს utf-8 ფორმატის ტექსტს.

მოთხოვნის მაგალითი შევსებული სახით:

(მაგალითში ნაჩვენებია თანხის ჩამოჭრის მეთოდი command v)

command=v&amount=300&currency=981&client\_ip\_addr=127.0.0.1&language=EN&description=UFCTEST&msg\_type=SMS

მოცემული მაგალითის მიხედვით მომხმარებელი გადამისამართდება ინგლისურენოვან (ქართულენოვანისათვის language პარამეტრში უნდა მიეთითოს GE მნიშვნელობა)გადახდის გვერდზე და ტრანზაქციის წარმატებით დასრულების შემთხვევაში ჩამოეჭრება 3 ლარი ბარათიდან(აღსანიშნავია რომ ტრანზაქციის თანხა მერჩანტმა უნდა გამოუშვას თეთრებში)

ტრანზაქციის აიდის დაგენერირება PHP CURL ბიბლიოთების მეშვეობით, სადაც უნდა მიეთითოს pem ფორმატში არსებული კონვერტირებული სერთიფიკატი:

<?php
 $curl = curl\_init();
 $post\_fields = "command=v&amount=500&currency=981&client\_ip\_addr=127.0.0.1&language=EN&description=UFCTEST&msg\_type=SMS";
 $submit\_url = "https://ecommerce.ufc.ge:18443/ecomm2/MerchantHandler";
Curl\_setopt($curl, CURLOPT\_SSLVERSION, 0);

 curl\_setopt($curl, CURLOPT\_POSTFIELDS, $post\_fields);

 curl\_setopt($curl, CURLOPT\_VERBOSE, '1');

 curl\_setopt($curl, CURLOPT\_SSL\_VERIFYHOST, '0');

 curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, '0');

 curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);

 curl\_setopt($curl, CURLOPT\_TIMEOUT, 120);

 curl\_setopt($curl, CURLOPT\_SSLCERT, getcwd().'/sertificate1.pem'); //სერთიფიკატის მისამართი

 curl\_setopt($curl, CURLOPT\_SSLKEYPASSWD, 'AAAaaa123456'); სერთიფიკატის პაროლი

 curl\_setopt($curl, CURLOPT\_URL, $submit\_url);

 $result = curl\_exec($curl);

 $info = curl\_getinfo($curl);

if(curl\_errno($curl))

{

 echo 'curl error:' . curl\_error($curl)."<BR>";

}

 curl\_close($curl);

echo $result; // ტრანზაქციის აიდის გამოტანა თვალსაჩინოებისათვის.

curl = curl\_init();

?>

სკრიპტის შედეგი, ტრანზაქციის აიდის დაბრუნების მაგალითი:

TRANSACTION\_ID: LUTI+KP7ikF7x0Ndg75lhIUbKuI=

ტრანზაქციის აიდის დაგენერირების შემდეგ მომხმარებლის გადახდის გვერდზე გადმოსამისამართებლად უნდა მოხდეს ტრანზაქციის აიდის დაპოსტვა შემდეგ მისამართზე.
[https://ecommerce.ufc.ge/ecomm2/ClientHandler](https://securepay.ufc.ge/ecomm2/ClientHandler)

ტრანზაქციის აიდის დაპოსტვის მაგალითი html ფორმის მეშვეობით:

<html>

<head>

<title>Merchant example post template to ECOMM</title>

<script type='text/javascript' language='javascript'>

function redirect() {

  document.returnform.submit();

}

</script>

</head>

<body onLoad='javascript:redirect()'>

<form name='returnform' action='[https://ecommerce.ufc.ge/ecomm2/ClientHandler](https://securepay.ufc.ge/ecomm2/ClientHandler)' method='POST'>

  <input type='hidden' name='trans\_id' value=’ტრანზაქციის უნიკალური იდენტიფიკატორი’>

<noscript>

    <center>Please click the submit button below.<br>

    <input type='submit' name='submit' value='Submit'></center>

</noscript>

</form>

</body>
</html>

 **1.1.2 მომხმარებლის გადამისამართება გადახდის გვერდზე.**

 აღსანიშნავია რომ ტრანზაქციის აიდის დაპოსტვის დროს უნდა მოხდეს UFC\_ის მიერ დაბრუნებული სტრინგიდან, ტრანზაქციის აიდის ამოჭრა და მხოლოდ მისი დაპოსტვა, ამისათვის PHP ში შესაძლებელია substr ფუნქციის გამოყენება.

 <meta http-equiv="content-type" content="text/html; charset=UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=8">
<?php
 $curl = curl\_init();
 $post\_fields = "command=V&amount=300&currency=981&client\_ip\_addr=192.168.0.88&description=UFCTEST&msg\_type=SMS";
 $submit\_url = "https://ecommerce.ufc.ge:18443/ecomm2/MerchantHandler";
 Curl\_setopt($curl, CURLOPT\_SSLVERSION, 1); //0
 curl\_setopt($curl, CURLOPT\_POSTFIELDS, $post\_fields);
 curl\_setopt($curl, CURLOPT\_VERBOSE, '1');
 curl\_setopt($curl, CURLOPT\_SSL\_VERIFYHOST, '0');
 curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, '0');
 curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);
 curl\_setopt($curl, CURLOPT\_TIMEOUT, 120);
 curl\_setopt($curl, CURLOPT\_SSLCERT, getcwd().'/sert.pem');
 curl\_setopt($curl, CURLOPT\_SSLKEYPASSWD, 'sert pass');
 curl\_setopt($curl, CURLOPT\_URL, $submit\_url);
 $result = curl\_exec($curl);
 $info = curl\_getinfo($curl);
if(curl\_errno($curl))
{
 echo 'curl error:' . curl\_error($curl)."<BR>";
}
 curl\_close($curl);
//echo $result;
//echo "<BR><BR>";
echo $result=substr($result,-28); //სტრინგიდან ტრანზაქციის აიდის ამოჭრა
//echo print\_r($info);
//echo "<BR><BR>";
$curl = curl\_init();
?>
<html>
<head>
<title>Merchant example post template to ECOMM</title>
<script type='text/javascript' language='javascript'>
function redirect() {
 document.returnform.submit();
}
</script>
</head>

<body onLoad='javascript:redirect()'>
<form name='returnform' action='https://ecommerce.ufc.ge/ecomm2/ClientHandler' method='POST'>
 <input type='hidden' name='trans\_id' value='<?php echo $result; ?>'>
<noscript>
<center>Please click the submit button below.<br>
 <input type='submit' name='submit' value='Submit'></center></noscript>
</form>
</body>
</html>

ტრანზაქციის აიდის არასწორად დაპოსტვის შემთხვევაში, ტრანზაქცია დარეჯექტდება WAF\_ის(web application firewall) მიერ და მომხმარებელი გავა forbidden სკრინზე.



ამ შემთხვევაში მერჩანტმა უნდა გადაამოწმოს რამდენად სწორად ხდება ტრანზაქციის აიდის დაპოსტვა clienthandler მისამართზე, ხოლო მომხმარებლის წარმატებით გადმომისამართების შემდეგ ტრანზაქციის შესასრულებლად მას შეყავს ბარათის მონაცემები და კლიკავს ღილაკს PAY.


გადახდის ღილაკზე დაკლიკვის შემდეგ, ტრანზაქციის სტატუსის მიუხედავად ე.ი ბარათის სწორი/არასწორი მონაცემების შევსების ან სხვა შემთხვევაში მომხმარებელი გადმომისამართდება OK მისამართზე, რომელიც მერჩანტმა მიუთითა რეგისტრაციის დროს, ხოლო FAIL მისამართზე მომხმარებელი გადამისამართდება მხოლოდ **სისტემური შეცდომის** დროს.

## **1.1.3 ტრანზაქციის სტატუსი(წარმატებული, წარუმატებელი).**

მომხმარებლის OK მისამართდზე დაბრუნების დროს, ამ გვერდზე UFC\_მხრიდან $\_POST["trans\_id"] მეთოდით იპოსტება ტრანზაქციის აიდი, რის მეშვეობითაც მერჩანტს აქვს შესაძლებლობა გაიგოს ტრანზაქციის სტატუსი OK მისამართიდან.

ინფორმაცია კონკრეტული ტრანზაქციის შესახებ, შესაძლებელია მიღებული იქნას შემდეგი ბრძანების მიხედვით(command c),
ტრანზაქციის სტატუსის გაგების სინტაქსი:

Command=c&trans\_id=<trans\_id>

აღსანიშნავია რომ როდესაც ხდება ტრანზაქციის აიდის დაპოსტვა trans\_id ის პარამეტრის მნიშვნელობის სახით უნდა მოხდეს ტრანზაქციის აიდის ენკოდინგი, რათა UFC\_ის მხარეს არსებულმა სისტემამ, იპოვოს შესაბამისი ტრანზაქცია. php ში სტრინგის ენკოდინგი შესაძლებელია გაკეთდეს urlencode ფუნქციის მეშვეობით.

ტრანზაქციის აიდი ენკოდინგამდე: B2R+qPonHFveOtgPMYs+J0Zb8kw=
ტრაბზაქციის აიდი ენკოდინგის შემდეგ: B2R%2BqPonHFveOtgPMYs%2BJ0Zb8kw%3D

ამის შემდეგ შესაბამის command c(სტატუსის გაგების კომანდა) ექნება ასეთი სახე:
Command=c&trans\_id= B2R%2BqPonHFveOtgPMYs%2BJ0Zb8kw%3D
ამ ტრანზაქციის წარმატების შემთხვევაში UFC დააბრუნებს პასუხს შემდეგნაირად:

RESULT: OK RESULT\_CODE: 000 RRN: 728418142503 APPROVAL\_CODE: 414576 CARD\_NUMBER: 4\*\*\*\*\*\*\*\*\*\*\*4813

სადაც RESULT\_CODE წარმოადგენს ტრანზაქციის შედეგის კოდს, წარმატებული ტრანზაქციის შემთხვევაში სისტემამ აუცილებლად უნდა დააბრუნოს „000“ სხვა დანარჩენ შემთხვევაში ტრანზაქცია წარუმატებლად ითვლება(კლიენტს არ ჩამოჭრია თანხა ბარათიდან).

ასევე წარუმატებლად ითვლება ის ტრანზაქცია, რომელზეც UFC\_ის სისტემა საერთოდ არ აბრუნებს სტრინგში ამ RESULT\_CODE კოდს.
 უფრო ვრცლად RESULT\_CODE აღწერილია დოკუმენტში, რომელსაც UFC აგზავნის მერჩანტთან ამ დოკუმენტთან ერთად.

OK მისამართიდან ტრანზაქციის სტატუსის გაგების სკრიპტი, შესაძლოა გამოიყურებოდეს ასე:
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=8">
 <?php
 echo "ORIGINAR TRANSACTION: " . $TRX\_ID2=$\_POST['trans\_id'];
 echo "</br>" . "</br>";
 echo $TRX\_ID=urlencode($TRX\_ID2); //ტრან აიდის ენკოდინგი
 ?>
<?php
 $curl = curl\_init();
 $post\_fields = "command=c&trans\_id=". $TRX\_ID . "&client\_ip\_addr=192.168.0.88";
 $submit\_url = "https://ecommerce.ufc.ge:18443/ecomm2/MerchantHandler";
 Curl\_setopt($curl, CURLOPT\_SSLVERSION, 1); //0
curl\_setopt($curl, CURLOPT\_POSTFIELDS, $post\_fields);
 curl\_setopt($curl, CURLOPT\_VERBOSE, '1');
 curl\_setopt($curl, CURLOPT\_SSL\_VERIFYHOST, '0');
 curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, '0');
 curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, 1);
 curl\_setopt($curl, CURLOPT\_TIMEOUT, 120);
curl\_setopt($curl, CURLOPT\_SSLCERT, getcwd().'/sert.pem');
 curl\_setopt($curl, CURLOPT\_SSLKEYPASSWD, 'pass');

 curl\_setopt($curl, CURLOPT\_URL, $submit\_url);
 $result = curl\_exec($curl);
 $info = curl\_getinfo($curl);
if(curl\_errno($curl))

{

 echo 'curl error:' . curl\_error($curl)."<BR>";

}
 curl\_close($curl);
//echo $result;
//echo "<BR><BR>";
echo $result; // =substr($result,-28);
//echo print\_r($info);
//echo "<BR><BR>";
 $curl = curl\_init();
?>

## **1.1.4 დღის დახურვის ოპერაცია**

დღის დახურვის შესრულების შემდეგ დღეს, მერჩანტის ანგარიშზე აისახება ყველა ის თანხა ფინანსურად, რომელიც მერჩანზე შესრულდა ამ ოპერაციამდე.

რეკომენდირებულია მერჩანტმა დღის დახურვის ოპერაცია გამოუშვას დღეში ერთხელ. როგორც წესი ამის ავტომატიზირება ხდება საიტის სერვერზე სერვერზე არსებული cron-jobs ის მეშვეობით.

დღის დახურვის ოპერაცია შესაძლებელია შემდეგი ბრძანების მეშვეობით

command=b

ხოლო წარმატების შემთხვევაში UFC სისტემა აბრუნებს 500 კოდს

RESULT: OK

RESULT\_CODE: 500 - დღის დახურვის პასუხის კოდი.

FLD\_075: 12

FLD\_076: 31

FLD\_087: 3201

FLD\_088: 10099

Field Descripton:

* FLD\_075 - Credits, Reversal Number.
* Field FLD\_076 - Debits, Number.
* FLD\_087 - Credits, Reversal Amount.
* Field FLD\_088 - Debits, Amount.

##  2**.1.1 თანხის ჩამოჭრა ანუ ტრანზაქციის რეგისტრაცია**

ეს არის ფინანსური ტრანზაქცია, ამ დროს ბართის მფლობელს თახა ეჭრება (ებლოკება) საბარათე ანგარიშიდან და ზემოთ აღწერილი დღის დახურვის შემდეგ მოხდება მერჩანტის ანგარიშზე ჩარიცხვა გარკვეული დროის განმავლობაში.

აუცილებელი პარამეტრები

-v ტრანზაქციის რეგისტრაციის კომანდა

amount თანხა (თეთრებში, ცენტებში) 100 = 1₾

currency ტრანზაქციის ვალუტა ISO NUMBER FORMAT (https://en.wikipedia.org/wiki/ISO\_4217).

client\_ip\_addr კლიენტის აიპი მისამართი.

description ტრანზაქციის აღწერა (არჩევითი)

language ტრანზაქციის ენა თუ რა ენაზე უნდა ჩაიტვირთოს გადახდის ტემპლეიტი(EN ან GE)

Http post პარამეტრები

command=v&amount=<amount>&currency=<currency>&client\_ip\_addr=<ip>&desc=<desc>&language=<language>&msg\_type=SMS

დაბრუნებული შედეგი

 TRANSACTION\_ID: bAt6JLX52DUbibbzD9gDFl5Ppr4=

## **2.1.2    პრე-ავტორიზაცია.**

პრე-ავტორიზაცია წარმოადგენს ტრანზაქციის ისეთ ტიპს, სადაც კლიენტს ებლოკება თანხა ბარათზე და არ ეჭრება, სანამ მერჩანტი არ ჩამოაჭრის, ბლოკის შენახვის დრო სტანდარტულად არის 30 დღე, თუმცა შესაძლებელია ზოგიერთ ბანკს სხვა პარამეტრი ჰქონდეს განსაზღვრული, ასე რომ ეს პარამეტრი დამოკიდებულია ბარათის მწარმოებელ ბანკზე (იშუერ ბანკი)

ეს ფუნქცია განსაკუთრებით გამოსადეგი შეიძლება იყოს სასტუმროებისათვის, სადაც მერჩანტმა არ იცის თუ ზუსტად რა თანხა უნდა ჩამოაჭრას კლიენტს ან ნებისმიერი სერვისის გაწევისას როდესაც ანგარიშსწორების თანხა შეიძლება შეიცვალოს.

მანდატორი პარამეტრები:

Command line პარამეტრები:

-a პრეავტორიზაციის კომანდა

amount ტრანზაქციის თანხა

currency ვალუტა

client\_ip\_addr კლიენტის აიპი მისამართი

description აღწერა (არჩევითი)

language ტრანზაქციის ენა თუ რა ენაზე უნდა ჩაირიტვირთოს გადახდის ტემპლეიტი(EN ან GE)

Http post პარამეტრები

command=a&amount=<amount>&currency=<currency>&client\_ip\_addr=<ip>&desc=<desc>&language=<language>&msg\_type=DMS

შედეგი:

 TRANSACTION\_ID: bAt6JLX52DUbibbzD9gDFl5Ppr4=

გასათვალისწინებელია რომ ამ კომანდის შემდეგ რეკომენდირებულია **1.1.3 ტრანზაქციის სტატუსი** პროცედურის შესრულება, რეზულტატის გასაგებად.

## **2.1.3 პრეავტორიზაციის კომიტი (ავტორიზაცია, დადასტურება)**

ბლოკში არსებული თანხის ჩამოსაჭრელად, შესაძლებელია კომიტის მეშვეობით(command =t).

კომიტში თანხა შესაძლოა იყოს განსხვავებული, პრე-ავტორიზაციისგან (command –a), თანხის სხვაობა(ჩამოსაჭრელი თანხის ნაკლებობის შემთხვევაში) უკან ჩაერიცხება ბარათის მფლობელს.

კომანდის პარამეტრები

 -t ტრანზაქციის კომიტი

trans\_id პრეავტორიზაციის ტრანზაქციის აიდი ენკოდირებული სახით, რომელი ტრანზაქციის კომიტიც უნდა განხორციელდეს.

amount ტრანზაქციის თანხა

currency ვალუტა

client\_ip\_addr კლიენტის აიპი მისამართი

description ტრანზაქციის აღწერა. (არჩევითი)

Http post პარამეტრები

command=t&trans\_id=<auth\_id>&amount=<amount>&currency=<currency>&client\_ip\_addr=<ip>&desc=<desc>&language=<language>&msg\_type=DMS

შედეგი:

 RESULT: <result>

 RESULT\_CODE: <result\_code>

 RRN: <rrn>

 APPROVAL\_CODE: <app\_code>

 CARD\_NUMBER <pan>

RESULT – transaction results: OK – წარმატებული ტრანზაცია, FAILED – წარუმატებელი

RESULT\_CODE –პასუხის კოდი თითოეულ ტრანზაქციაზე (3 digits) // გთხოვთ იხილოთ

 result\_code.docx ფაილი

RRN – ტრანზაქციის იდენტიფიკატორი ბანკის მხარეს (12 characters)

APP\_CODE – ტრანზაქციის იდენტიფიკატორი ბანკის ხმარეს (max 6 characters)

PAN – დამასკული ბართის ნომერი.

RESULT\_CODE არის მხოლოდ ინფომრაციული სახის. ველები rrn და app\_code გენერირდება მხოლოდ ტექნიკურად წარმატებული ტრანზაქციის დროს.

ტრანზაქცია წარმატებული იყო თუ წარუმატებელი შესაძლებელია მხოლოდ RESULT\_CODE მნიშვნელობით განსაზღვრა.

**როდესაც ტრანზაქცია/პრე-ავტორიზაცია წარმატებულია ბრუნდება მხოლოდ 000 მნიშვნელობა RESULT\_CODE-ში.**

**სხვა ყველა წარუმატებელია!!!**

* გარდა დღის დახურვის 500 კოდისა რომელიც დღის დახურვის წარმატებული კოდია.
* გარდა რევერსალის 400 კოდისა, რომელიც რევერსალის წარმატებული კოდია.

შედეგი:

 RESULT: OK

 RESULT\_CODE: 000

 RRN: 123456789012

 APPROVAL\_CODE: 123456
 CARD\_NUMBER: 9\*\*\*\*\*\*\*\*\*\*\*999

## **2.1. 4    ტრანზაქციის რევერსალი**

მერჩანტზე რევერსალის ფუნქციონალის გააქტიურება ხდება ავტომატურად, ამ ფუნქციონალის მეშვეობით შესაძლებელია დღის დახურვამდე თანხის მყისიერად უკან დაბრუნება. აუცილებელია მერჩანტმა ეს სერვისი ააწყოს თავის მხარეს, რათა მომავალში თავიდან აიცილოს პრობლემები თანხის დაბრუნების შესახებ. რევერსალის გამოყენების დროს გასათვალისწინებელია შემდეგი გარემოებები:

* რევერსალის შესრულება ერთ ავტორიზაცია/ტრანზაქციაზე შეიძლება შესრულდეს მხოლოდ ერთხელ.
* ნაწილობრივი რევერსალის გაკეთება შესაძლებელია, დარევერსალებული თანხა ჩაერიცხება ბარათის მფლობელს, დარჩენილი ნაწილი ჩაერიცხება მერჩანტს, დღის დახურვის შემდეგ. (თუ მოხდა ისე, რომ კლიენტს ჩამოჭრილი თანხიდან უნდა დაუბრუნდეს ორჯერ თანხა, ანუ ნაწილ ნაწილ, ამ შემთხვევაში შესძლებელია მხოლოდ ერთხელ Reversal-ი და მეორე ნაწილის დაბრუნებისთვის ბანკს მიმართოთ. სისტემა არ იძლევა ერთხელ შესრულებულ რევერსალზე, მეორეთაც გააკეთოთ.)
* ნაწილობრივი რევერსალი ვერ გაკეთდება თუ შესრულებულია command –a პრეავტორიზაცია. ამ დროს თუ საჭიროა, თანხის ნაწილობრივი დაბრუნება უნდა გაკეთდეს command –t იმ თანხაზე, რაც მერჩანტს უნდა ჩაერიცხოს, ხოლო სხვაობა ჩერიცხება ბართის მფლობელს(იხ. 2.1.3 პრეავტორიზაციის კომიტი)
* რევერსალის თანხა არ უნდა აღემატებოდეს ორიგინალ თანხას.
* რევერსალის კომანდაში ტრანზაქციის აიდის გადმოცემა ხორციელდება ენკოდირებული სახით.

მაგ: თუ დღე მერჩანტზე იხურება საღამოს 17:00 PM ზე, ტრანზაქცია შესრულდა 13:00 PM ზე ამ თახის დარევერსალება შესაძლებელი იქნება 17:00 PM-ის ჩათვლით, რადგან ტრანზაქციაზე რევერსალი ვეღარ შესრულდება დღის დახურვის შემდეგ. (ამ შემთხვევაში საჭიროა რეფანდი იხ.ქვევით 2.1.5 ტრანზაქციის რეფანდი )

კომანდის პარამეტრები:

 -r ტრანზაქციის რევერალი

trans\_id დასარევერსალებელი ტრანზაქციის ტრან\_აიდი ენკოდირებული სახით.

Http პოსტ პარამეტრები

command=r&trans\_id=<trans\_id>&amount=<amount>

შედეგი:

 RESULT: <result>

 RESULT\_CODE: <result\_code>

 RESULT – რევერსალის შედეგი:

OK – წარმატებული

FAILED – წარუმატებელი

შედეგის მაგალითი:

 RESULT: OK

 RESULT\_CODE: 400

**რევერსალის RESULT\_CODE უნდა დაბრუნდეს კოდი 400 რომ ჩაითვალოს წარმატებულად.**

**ყველა სხვა შემთხვევაში რევერსალი წარუმატებელია.**

## **2.1.5 ტრანზაქციის რეფანდი**

მერჩანტზე რეფანდ ფუნქციონალის გააქტიურება ხდება ავტომატურად, ამ ფუნქციონალის მეშვეობით შესაძლებელია კლიენტისათვის თანხის უკან დაბრუნება დღის დახურვის შემდგომ, აუცილებელია მერჩანტმა ეს სერვისი ააწყოს თავის მხარეს, რათა მომავალში თავიდან აიცილოს პრობლემები თანხის დაბრუნების შესახებ.

ფუნქციონალის გამოყენების დროს გასათვალისწინებელი გარემოებები:

* რეფანდი შეიძლება შესრულდეს მხოლოდ იმ ტრანზაქციაზე, რომლის შემდეგაც გაკეთებულია დღის დახურვა (1.1.4 დღის დახურვა)

მაგ: თუ დღე იხურება ღამის 17:00 PM ზე, ტრანზაქცია შესრულდა 13:00 PM ზე და ამ თახის დაბრუნება გახდა საჭირო 17:00 PM შემდეგ, უნდა შესრულდეს რეფანდი.

* ნაწილობრივი რეფანდის შესრულება შესაძლებელია.
* განმეორებით ნაწილობრივი რენფადის შესრულება შესაძლებელია.
* რეფანდის, რეფანდების ჯამური თანხა არ უნდა აღემატებოდეს ორიგინალი ტრანზაქციის თანხას.
* თუ კომანდაში მითითებული არ იქნება amount პარამეტრი მოხდება სრული თანხის დაბრუნება.

კომანდის პარამეტრები

 -k ტრანზაქციის რეფანდი

trans\_id - შესაბამისი ტრანზაქციის აიდი ენკოდირებული სახით, რომლის რეფანდიც ხორციელდება.

Http post პარამეტრები:

command=k&trans\_id=<trans\_id>&amount=<amount>

შედეგი:

 RESULT: <result>

 RESULT\_CODE: <result\_code>

result – რეფანდის შედეგი:

OK წარმატებული.

FAILED წარუმატებელი.

რეფანდის შემთხვევაში RESULT\_CODE უნდა დაბრუნდეს 000, სხვა შემთხვევაში რეფანდი შეიძლება ჩაითვალოს წარუმატებლად.

## **2.2.1 რეგულარული გადახდები**

მერჩანტზე რეგულარული გადახდის გააქტიურება ხდება, მხოლოდ ბანკის დასტურის შემდეგ

 რეგულარული გადახდა, არის ფუნქციონალი რომლის დროსაც მომხმარებელი მხოლოდ ერთხელ გადადის გადახდის გვერდზე და ყოველ შედგომ თანხის ჩამოჭრას ახორციელებს მერჩანტი, ქვევით მოცემული კომანდის მეშვეობით გადახდის გვერდზე გადასვლის გარეშე.

რეგულარული გადახდა არცერთ საწყის მერჩანტზე არ არის ჩართული და ხდება მხოლოდ ბანკთან შეთანხმების შემდეგ.

 რეგულარული გადახდის დროს პირველ ეტაპზე UFC-ის სისტემაში ხორციელდება ბარათის დამახსოვრება შესაბამისი(biller\_client\_id) პარამეტრით, რომელსაც აგენერირებს მერჩანტი, ხოლო ამ პარამეტრის მეშვეობით მერჩანტს შეუძლია შემდგომ დამახსოვრებულ ბარათიდან თანხის ჩამოჭრა.

## **2.2.2 ბარათის რეგისტრაცია**

ამ ფუნქციონალით ბარათის რეგისტრაცია ხორციელდება შემდეგი პარამეტრების მეშვეობით.

კომანდის პარამეტრები:

-z ტრანზაქციის რეგისტრაცია

currency ვალუტა

client\_ip\_addr კლიენტის აიპი მისამართი

description ტრანზაქციის დეტალები

recc\_pmnt\_id მერჩანტის მიერ დაგენერირებული მნიშვნელობა, რის მიხედვითაც განახორციელებს ამ ბარათიდან თანხის ჩამოჭრის ოპერაციას

expiry რეგულარული გადახდის რეგისტრაციის ვადა MMYY

(მაგ perspayee\_expiry=0822), თუ მერჩანტის მიერ რეგულარული გადახდის ვადა აღებატება ბარათის მოქმედების ვადას, სისტემა ავტომატურად მიანიჭებს ბარათის მოქმედების ვადას.

**კომანდის პარამეტრები სადაც მომხმარებელს პირველ გადახდაზე არ ეჭრება თანხა**

-p ავტორიზაციის რეგისტრაცია

currency ვალუტა
client\_ip\_addr კლიენტის აიპი მისამართი

description ტრანზაქციის დეტალები

biller\_client\_id მერჩანტის მიერ დაგენერირებული მნიშვნელობა, რის მიხედვითაც განახორციელებს ამ ბარათიდან თანხის ჩამოჭრის ოპერაციას

expiry რეგულარული გადახდის რეგისტრაციის ვადა MMYY

(მაგ perspayee\_expiry=0822), თუ მერჩანტის მიერ რეგულარული გადახდის ვადა აღებატება ბარათის მოქმედების ვადას, სისტემა ავტომატურად მიანიჭებს ბარათის მოქმედების ვადას.

Http post პარამეტრები

command=z&amount=<amount>&currency=<currency>&client\_ip\_addr=<ip>&description=<des c>&language=<language>&msg\_type=SMS&biller\_client\_id=<recc\_pmnt\_id>&perspayee\_expi ry=<expiry>&perspayee\_gen=1

command=p&amount=0&currency=<currency>&client\_ip\_addr=<ip>&desciption=<desc>&language=<language>&msg\_type=AUTH&biller\_client\_id=<recc\_pmnt\_id>&perspayee\_expiry=<expiry>&perspayee\_gen=1

კომანდის მაგალითი:
command=Z&amount=100&currency=981&client\_ip\_addr=192.168.0.88&description=UFCTEST8&msg\_type=SMS&language=GE&biller\_client\_id=956504\_ABC88&perspayee\_expiry=0820&perspayee\_gen=1

## **2.2.3 თანხის ჩამოჭრა(Execution)**

მოცემული ფუნქციის მეშვეობით მერჩანტს შეუძლია ჩამოაჭრას თანხა დამახსოვრებული ბარათის მფლობელს.

კომანდის პარამეტრები:

 -e თანხის ჩამოჭრა

amount თანხა

currency ვალუტა

client\_ip\_addr კლიენტის აიპი მისამართი

description აღწერა

biller\_client\_id მერჩანტის მიერ დაგენერირებული მნიშვნელობა, რომლის მიხედვითაც დაარეგისტრირა სასურველი ბარათი.

Http post პარამეტრები:

command=e&amount=<amount>&currency=<currency>&client\_ip\_addr=<ip>&desc=<desc>&language=<language>&biller\_client\_id=<biller\_client\_id >(&<property\_name>=<property\_value>)\*

Result:

 TRANSACTION\_ID: <trans\_id>

 RESULT\_CODE:<result\_code>

 RRN:<rrn>

 APPROVAL\_CODE:<appr\_code>

კომანდის მაგალითი:
command=e&amount=100&currency=981&client\_ip\_addr=192.168.0.88&description=test&biller\_client\_id=956504\_ABC88

## **2.2.4 კრედიტული ოპერაციები(original credit)**

მერჩანტზე კრედიტული ოპერაციების გააქტიურება ხდება, მხოლოდ ბანკის დასტურის შემდგომ. კრედიტული ოპერაციების დროს მერჩანტს შეუძლია მომხმარებელს დაუსვას თანხა ბარათზე, მის მიერ განხორციელებული წარმატებილი დებიტური ოპერაციის საფუძველზე. (მაგ: ასე ხდება გემბლინგ მერჩანტების შემთხვევაში).

კომანდის პარამეტრები:

 -g კრედიტული ოპერაცია

 -trans\_id დებიტური ტრანზაქციის აიდი ენკოდირებული სახით, რომელ ბარათზეც ხორციელდება კრედიტული ოპერაცია.

-amount თანხა

command=g&trans\_id=<trans\_id>&amount=<amount>

შედეგი**:**

 RESULT: <result>

 RESULT\_CODE:<result\_code>

 REFUND TRANS ID:<refund transaction id>

RESULT: <result>

RESULT\_CODE: <result\_code>
REFUND\_TRANS\_ID: <refund\_transaction\_id>

კომანდის მაგალითი:
command=g&trans\_id=DRaz5jlgxpLMrmvL%2Bt056LZSEZg%3D&amount=100

##  **ხშირად დასმული კითხვები.**

1. **სერთიფიკატის შეიმპორტების შემდეგ ბრუნდება მსგავსი შეცდომა curl error: unable to load client key:** ამ შეცდომას აბრუნებს სერვერი, რომელსაც სავარაუდოდ არ აქვს DES encryption\_ის მხარდაჭერა.

 ამ შემთხვევაში სერთიფიკატის კონვერტირება (p12 დან pem\_ში) საჭიროა მოხდეს შემდეგი კომანდით.

 openssl pkcs12 -in \*.p12 -out tbcpay.pem -nodes

 ამ საკითხზე დამატებით კითხვების შემთხვევაში უნდა მოხდეს მიმართვა მერჩანტის სერვერის(ჰოსტინგის) მხარდაჭერის ჯგუფთან.

1. **ტრანზაქციის აიდის დაგენერირების დროს ბრუნდება შემდეგი შეცდომა unregistered merchant. IP:** ეს შეცდომა ბრუნდება მაშინ როდესაც მოცემულ აიპზე წვდომა გახსნილია, მაგრამ არ არის გაწერილი კონკრეტულ მერჩანტზე, საჭიროა მოხდეს სერვერის გამომავალი აიპის გადამოწმება და მისი დაფიქსირება UFC\_ის საპორტთან.
2. **ფიქსირდება SSL/TLS შეცდომა:** UFC სერვერთან კომუნიკაციისას აუცილებელია რომ მერჩანტის სერვერს ჰქონდეს tls1.2 პროტოკოლის და SSLv3 მხარდაჭერა.(ვერსიების გადამოწმება ხდება სერვერის საპორტთან)