

# eWallet API integration guide

version 5.1  
8/31/2015

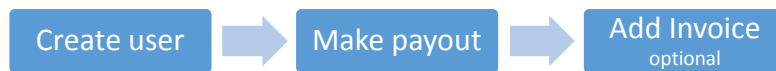


International Payout Systems, Inc. (IPS) eWallet API Integration Guide contains information proprietary to IPS, and is intended only to be used in conjunction with IPS products and services.

This Guide contains information protected by copyright. No part of this manual may be photocopied or reproduced in any form without prior written consent from IPS. Information contained in this manual is subject to change without notice.

Making Global Business Happen

## Typical workflow:



## Main eWallet API Functions

- Create user: **eWallet\_CreateUser**
- Make payout: **eWallet\_Load**
- Add invoice: **eWallet\_AddCheckoutItems**
- Auto login to eWallet: **eWallet\_RequestUserAutoLogin**

## eWallet API

eWallet API can be accessed via **{JSON}** or Key-Value Pairs (KVP) format:

<b>TEST JSON</b>	<a href="https://testewallet.com/eWalletWS/ws_JsonAdapter.aspx">https://testewallet.com/eWalletWS/ws_JsonAdapter.aspx</a>
<b>PRODUCTION JSON</b>	<a href="https://i-payout.net/eWalletWS/ws_JsonAdapter.aspx">https://i-payout.net/eWalletWS/ws_JsonAdapter.aspx</a>
<b>TEST KVP</b>	<a href="https://testewallet.com/eWalletWS/ws_Adapter.aspx">https://testewallet.com/eWalletWS/ws_Adapter.aspx</a>
<b>PRODUCTION KVP</b>	<a href="https://i-payout.net/eWalletWS/ws_Adapter.aspx">https://i-payout.net/eWalletWS/ws_Adapter.aspx</a>

For a full API document, PHP and C# examples please go to <https://www.i-payout.com/api>

All functions in eWallet API return a **Response** structure. You always have to look at **m\_code** and **m\_text** values. When **m\_code** = 0 -> it means success. If **m\_code** < 0 it means there was a problem, read **m\_text** for a description.

### JSON Request Format:

```
{"fn":"function name","parameter_name":"parameter_value"}
```

- **fn**: name of a function, like: eWallet\_CreateUser
- **parameter\_name and parameter\_value**: *input fields* per function \*

### Key Value Pairs (KVP) Request Format:

```
?fn=functionName&parameters
```

- **functionName**: name of a function, like: eWallet\_CreateUser
- **parameters**: *input fields* per function in a key-value pair format \*

**Format to make an array**: [field1=value1%26field2=value2][field1=value3%26field2=value4]. If value contains character '=', you will need to double URL encrypt it.

### Key Value Pairs Response Format:

Each member of a **Response** structure is URL-encoded. In order to parse a response correctly, you have to get a "response" value of a HTTP response variable and URL-decode it.

NOTE: **m\_code** member of a **Response** structure in JSON returns a numeric code like "-2", while in KVP it returns a text value, like "CUSTOMER\_NOT\_FOUND". Please always read **m\_text** member for a complete description of an error.

\* All required parameters must be submitted; all other missed parameters will be treated as NULL, empty, false or 0.

## eWallet\_CreateUser

Use this function to register a user in the eWallet system.

This function is a short version of **eWallet\_RegisterUser**.

### Input Fields:

Request Field	Data Type	Required	Comment
MerchantGUID	String	Yes	API Merchant ID (contact IPS to receive it)
MerchantPassword	String	Yes	API Merchant password (contact IPS to receive it)
UserName	String[100]	Yes	User name to register. It can be any name, email, ID, and so on
FirstName	String[50]	Yes	First name of a user. If Company name is entered, it should be a first name of an owner of this company
LastName	String[50]	Yes	Last name of a user. If Company name is entered, it should be a last name of an owner of this company
CompanyName	String[50]	No	Company name if eWallet account holder is a business. Additional documents might be required, like registration documents, proof of ownership of this company, and tax id of this company
Address1	String[100]	No	<b>Home</b> address of this user. If Company name is entered, it should be a home address of an owner of this company
Address2	String[100]	No	Use Address2 field for Apt# or leave it empty
City	String[30]	No	City name
State	String[20]	No	For US users: state should be as 2 character format, like <b>FL, TX</b> For non-US users: state might be an empty string
ZipCode	String[20]	No	For US users: ZIP code must be 5- or 9-digit format (12345-6789) For non US users: ZIP code can be in any format
Country2xFormat	String[2]	No	Country should be in 2 character format, like <b>US, CA, GB, RU</b>
EmailAddress	String[100]	Yes	Upon registration and after any important eWallet activity, user will receive an email into this address (please make sure that this address is correct and user can receive emails into it)
DateOfBirth	String[10]	Yes	User's date of birth in format: <b>YYYY-MM-DD</b> . Should be 18 or older. If date of birth is not known, please set it to: <b>1900-01-01</b>
WebsitePassword	String[20]	No	Login password to eWallet. It can be the same password as for back office. If a password is not specified, a random password will be generated and email will be sent to this user. Password cannot contain spaces or the symbols "<" and ">"

### Return:

Response Field	Data Type	Comment
m_Code	Int (JSON) / string (KVP)	0 or NO_ERROR -> success
m_Text	String	Description of m_code
TransactionRefID	Int64	User's ID in eWallet platform

### API call example:

<b>REQUEST</b> {"fn":"eWallet_CreateUser","MerchantGUID":"12345678-e2f9-4a42-bef1-258eb99c1b45","MerchantPassword":"1a#@Sdr","UserName": "User1", "FirstName":"John","LastName":"Doe","CompanyName":"","Address1":"2500 E Hallandale Beach","Address2":"Suite 800", "City":"Hallandale beach","State":"FL","ZipCode":"33009","Country2xFormat":"US","EmailAddress":"test@user.com","DateOfBirth":"1980-12-31","WebsitePassword":""}
<b>RESPONSE (SUCCESS)</b> {"response":{"m_Code":0,"m_Text":"OK","LogTransactionID":0,"TransactionRefID":230004,"ACHTransactionID":0,"ProcessorTransactionRefNumber":"","CustomerFeeAmount":0.0,"CurrencyCode":null}}
<b>RESPONSE (FAILURE)</b> {"response":{"m_Code":-1,"m_Text":"Country should be in 2 letter format, like US","LogTransactionID":0,"TransactionRefID":0,"ACHTransactionID":0,"ProcessorTransactionRefNumber":"","CustomerFeeAmount":0.0,"CurrencyCode":null}}

### Remarks:

It is recommended to use "auto-login to eWallet" feature after registration (**eWallet\_RequestUserAutoLogin**).

## eWallet\_AddCheckoutItems

Use this function to add an invoice from a shopping cart to eWallet account for a checkout.

### Input Fields:

Request Field	Type	Required	Comment																											
MerchantGUID	String	Yes	API Merchant ID																											
MerchantPassword	String	Yes	API Merchant password																											
UserName	String	Yes	User name to whom to assign an invoice																											
arrItems (array of <b>CheckoutItem</b> )	array	Yes	<p><b>CheckoutItem</b> structure:</p> <table border="1"> <thead> <tr> <th>Field</th> <th>Type</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>Amount</td> <td>Decimal</td> <td>amount to pay. Format: <b>xxx.yy</b> (use period as a decimal separator)</td> </tr> <tr> <td>CurrencyCode</td> <td>String[3]</td> <td>currency code, like USD</td> </tr> <tr> <td>ItemDescription</td> <td>String[170]</td> <td>description of the item. Please use a <u>complete</u> description, like: "12 bottle pack of Solar Energy Drink" or "Monthly subscription for</td> </tr> <tr> <td>MerchantReferenceID</td> <td>String[70]</td> <td>reference ID in your system. It will be used in a notification call back using <i>Merchant Notify URL</i></td> </tr> <tr> <td>UserReturnURL</td> <td>String[250]</td> <td>if a payment was done instantly (via credit card or eWallet balance) on eWallet website, user can click on this URL after payment is completed</td> </tr> <tr> <td>MustComplete</td> <td>Boolean</td> <td>flag to set if this payment can be canceled manually by user. (<i>recommended value: false</i>)</td> </tr> <tr> <td>IsSubscription</td> <td>Boolean</td> <td>if <b>true</b> this checkout item will generate a monthly invoice. The first recurring invoice will begin 1 month after this invoice has been successfully paid and funds are settled</td> </tr> <tr> <td>SubscriptionAmount</td> <td>Decimal</td> <td>monthly payment amount. Format: <b>xxx.yy</b></td> </tr> </tbody> </table>	Field	Type	Comment	Amount	Decimal	amount to pay. Format: <b>xxx.yy</b> (use period as a decimal separator)	CurrencyCode	String[3]	currency code, like USD	ItemDescription	String[170]	description of the item. Please use a <u>complete</u> description, like: "12 bottle pack of Solar Energy Drink" or "Monthly subscription for	MerchantReferenceID	String[70]	reference ID in your system. It will be used in a notification call back using <i>Merchant Notify URL</i>	UserReturnURL	String[250]	if a payment was done instantly (via credit card or eWallet balance) on eWallet website, user can click on this URL after payment is completed	MustComplete	Boolean	flag to set if this payment can be canceled manually by user. ( <i>recommended value: false</i> )	IsSubscription	Boolean	if <b>true</b> this checkout item will generate a monthly invoice. The first recurring invoice will begin 1 month after this invoice has been successfully paid and funds are settled	SubscriptionAmount	Decimal	monthly payment amount. Format: <b>xxx.yy</b>
Field	Type	Comment																												
Amount	Decimal	amount to pay. Format: <b>xxx.yy</b> (use period as a decimal separator)																												
CurrencyCode	String[3]	currency code, like USD																												
ItemDescription	String[170]	description of the item. Please use a <u>complete</u> description, like: "12 bottle pack of Solar Energy Drink" or "Monthly subscription for																												
MerchantReferenceID	String[70]	reference ID in your system. It will be used in a notification call back using <i>Merchant Notify URL</i>																												
UserReturnURL	String[250]	if a payment was done instantly (via credit card or eWallet balance) on eWallet website, user can click on this URL after payment is completed																												
MustComplete	Boolean	flag to set if this payment can be canceled manually by user. ( <i>recommended value: false</i> )																												
IsSubscription	Boolean	if <b>true</b> this checkout item will generate a monthly invoice. The first recurring invoice will begin 1 month after this invoice has been successfully paid and funds are settled																												
SubscriptionAmount	Decimal	monthly payment amount. Format: <b>xxx.yy</b>																												
AutoChargeAccount	Bool	Yes	<p><b>false (recommended)</b> – User has to go to eWallet, confirm, select a payment method and enter a security PIN.  <b>To avoid unauthorized charges it is recommended to use FALSE at all times so a user can review an invoice and enter his/her Security PIN.</b></p> <p><b>true</b> – auto charge eWallet account if there is:            1) enough balance on eWallet OR            2) secondary payment account is set: verified bank account (US only) or credit card (subject to amount limitation per merchant setup, and if a processor can process credit cards without CVV2).  <b>Please contact IPS to enable auto charge account if required</b></p>																											

### Return:

Response Field	Data Type	Comment
m_Code	Int (JSON) / string (KVP)	0 or NO_ERROR -> success
m_Text	String	Description of m_code
arrItemsResponse	Response []	Array of <b>Response</b> objects for each submitted invoice in arrItems

### Remarks:

This function will try to:

- Add an invoice to pay into eWallet account
- If **AutoChargeAccount** =
  - True*: instantly debit from eWallet. If there is insufficient funds then function will debit from a secondary payment account (if set). System will send an email to a user about this transaction.
  - False*: System will send an email to a user to complete this transaction (recommended).
- If function returned in **arrItemsResponse** m\_Code =
  - NO\_ERROR (0) – transaction was processed/settled and completed successfully (AutoChargeAccount = true)
  - PROCESSING (-58) – transaction was submitted without user interaction and it will take some time to complete this payment. For example, if a secondary account is a bank account then it might take 3-5 business days to complete this payment. (AutoChargeAccount = true)
  - PENDING (-57) – user interaction is required. In this case, the system will send a notification email to the user to complete this transaction. (AutoChargeAccount = false)

Upon a completion or user's cancelation of this payment (if **MustComplete** flag is *False*), the eWallet notification system will call a page on your server (*Merchant Notify URL*) with the following parameters:

1. act=**PaymentToMerchant**
2. status\_id=status of this invoice (can be only SETTLED (*invoice is paid*) or VOIDED (*invoice is canceled*))
3. status\_desc=status description
4. trnx\_id=MerchantReferenceID
5. log\_id=IPS TransactionID (numeric) of a payment, same as Response.LogTransactionID
6. hash=SHA1 hash of: trnx\_id + log\_id + MerchantGUID + MerchantPassword

#### Hash calculation

For example, if you received a notification with the following parameters:

trnx\_id = 54321, log\_id = 8790

Your MerchantGUID = 6d099995-a2f9-4a01-bef1-258eb99c1a77, MerchantPassword = psd123

You would combine all parameters into a string like: 5432187906d099995-a2f9-4a01-bef1-258eb99c1a77psd123

and then compute SHA1 hash of that string, which would generate a hash as

**F225DA02326628D9444468AB4868A7386D43C912**

If your computed hash matches with a received one then this notification is valid and you can process this payment.

You can use <http://www.sha1-online.com> to compare hash values during testing.

*Merchant Notify URL* will be called as a POST request. In a response, your script must write into output **OK** (only 2 characters, do not add any HTML tags) signaling that you processed this notification. If "OK" is not received back, the eWallet notification system will try to call your page 4 more times with intervals about 1 min apart.

To set up your *Merchant Notify URL* you need to login into eWallet Management Console and set it up in "System Overview" page in the **eWallet Setup** section.

If **IsSubscription** = True **eWallet\_AddCheckoutItems** will call internally **eWallet\_AddUserRecurringPayin** function. The subscription will activate when a payment for this invoice is settled (it might take several days to settle after payment is done, for example a payment via a wire or a bank account). When payment settles, a subscription's start day will be a *settlement day* and following subscription invoices will be posted on user's eWallet every month relative to this settlement day. For notifications on subscription payment please see Remarks in **eWallet\_AddUserRecurringPayin** function. **If initial payment is not required for the subscription, please use function eWallet\_AddUserRecurringPayIn**

To test notifications from **eWallet\_AddCheckoutItems**, you can load eWallet account by using **eWallet\_Load**.

#### API call example

##### REQUEST

```
{ "fn": "eWallet_AddCheckoutItems", "MerchantGUID": "12345678-e2f9-4a01-bef1-258eb99c1b77", "MerchantPassword": "12#@Sdr", "UserName": "ip123", "arrItems": [{"Amount": 10.50, "CurrencyCode": "USD", "ItemDescription": "lotions and potions", "MerchantReferenceID": "1234567890", "UserReturnURL": "", "MustComplete": false, "Status": 0, "IsSubscription": false, "SubscriptionAmount": 0.0, "SubscriptionDaysToStart": 0}, {"AutoChargeAccount": false}]
```

##### RESPONSE (SUCCESS)

```
{ "response": {"m_Code": 0, "m_Text": "OK", "LogTransactionID": 0, "TransactionRefID": 0, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": null}, "arrItemsResponse": [{"m_Code": -57, "m_Text": "", "LogTransactionID": 183078, "TransactionRefID": 91506, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": "USD"}]}
```

**m\_code = -57** in *arrItemsResponse* means that an invoice was submitted successfully and user has to login to pay.

##### RESPONSE (FAILURE)

```
{ "response": {"m_Code": -2, "m_Text": "User not found", "LogTransactionID": 0, "TransactionRefID": 0, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": null}, "arrItemsResponse": []}
```

##### RESPONSE (FAILURE in checkout items)

```
{ "response": {"m_Code": 0, "m_Text": "OK", "LogTransactionID": 0, "TransactionRefID": 0, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": null}, "arrItemsResponse": [{"m_Code": -3, "m_Text": "Such Reference ID: 1234567890 already exists in the system", "LogTransactionID": 0, "TransactionRefID": 0, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": null}]}
```

## eWallet\_Load

Use this function to issue a payout to eWallet.

### Input Fields:

Request Field	Data Type	Required	Comment															
MerchantGUID	Guid	Yes	API Merchant ID															
MerchantPassword	String	Yes	API Merchant password															
PartnerBatchID	String[70]	Yes	You can call <b>eWallet_Load</b> multiple times with the same <i>PartnerBatchID</i> to group multiple loads under the same batch. The best practice is to use a value that is easily readable like, "commissions on 3/12/2012". This field is not displayed in eWallet															
PoolID	String[50]	No	Pool/level/tier. This field is displayed in eWallet in a history section															
arrAccounts	eWalletLoad	Yes	Array of <b>eWalletLoad</b> objects (eWallet accounts) to load <table border="1"><thead><tr><th>Field</th><th>Data Type</th><th>Comment</th></tr></thead><tbody><tr><td>UserName</td><td>String</td><td>eWallet user name of the customer</td></tr><tr><td>Amount</td><td>Decimal</td><td>Commission payout amount. Format xxx.yy</td></tr><tr><td>Comments</td><td>String[125]</td><td>Short description for the commission payout (user will see it in a history page)</td></tr><tr><td>MerchantReferenceID</td><td>String[25]</td><td>Reference id from merchant's platform. MerchantReferenceID can be used in <b>eWallet_FindTransaction</b> method</td></tr></tbody></table>	Field	Data Type	Comment	UserName	String	eWallet user name of the customer	Amount	Decimal	Commission payout amount. Format xxx.yy	Comments	String[125]	Short description for the commission payout (user will see it in a history page)	MerchantReferenceID	String[25]	Reference id from merchant's platform. MerchantReferenceID can be used in <b>eWallet_FindTransaction</b> method
Field	Data Type	Comment																
UserName	String	eWallet user name of the customer																
Amount	Decimal	Commission payout amount. Format xxx.yy																
Comments	String[125]	Short description for the commission payout (user will see it in a history page)																
MerchantReferenceID	String[25]	Reference id from merchant's platform. MerchantReferenceID can be used in <b>eWallet_FindTransaction</b> method																
AllowDuplicates	Boolean	Yes	Allow multiple the same user names in this <i>PartnerBatchID</i>															
AutoLoad	Boolean	Yes	If set to TRUE - automatically approve and load payout into eWallets. A batch will be closed, and no other loads can be added into this batch. <b>NOTE: This flag must be enabled on IPS' side first. To enable, please send a request via email to <a href="mailto:compliance@i-payout.com">compliance@i-payout.com</a>. Once batch is approved and loaded, there is no way to cancel it</b>															
CurrencyCode	String[3]	Yes	Currency code of the transaction, like USD															

### Return:

Response Field	Data Type	Comment
m_Code	Int (JSON) / string (KVP)	0 or NO_ERROR -> success
m_Text	String	Description of m_code

### Remarks

- If *AutoLoad* = false and return is NO\_ERROR (0), eWallet loads are recorded and ready to be approved. To load eWallets a manager with sufficient permissions needs to approve this payout by logging into IPS management console and go to menu **EWALLET -> Payout Requests Management**.
- Once batch is approved, no more loads can be added into it.
- If **m\_code** returns a negative value, no loads will be added into this batch from this call.
- To avoid network timeouts in large loads, you can call **eWallet\_Load** multiple times with 1000 or fewer accounts in each call (use the same *PartnerBatchID*).

### API call example

<b>REQUEST</b> { "fn": "eWallet_Load", "MerchantGUID": "12345678-e2f9-4a01-bef1-258eb99c1b77", "MerchantPassword": "12#@Sdr", "PartnerBatchID": "635330851376218624", "PoolID": "635330851376218624", "arrAccounts": [{"UserName": "ip123", "Amount": 10.50, "Comments": "Test Json", "MerchantReferenceID": "15645121"}, {"UserName": "ip001", "Amount": 20.50, "Comments": "Test Json", "MerchantReferenceID": "1455645121"}], "AllowDuplicates": true, "AutoLoad": false, "CurrencyCode": "USD" }
<b>RESPONSE (SUCCESS)</b> { "response": { "m_Code": 0, "m_Text": "OK", "LogTransactionID": 0, "TransactionRefID": 32493, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": null } }
<b>RESPONSE (FAILURE)</b> { "response": { "m_Code": -2, "m_Text": "Customer with user name ip123 is not found", "LogTransactionID": 0, "TransactionRefID": 0, "ACHTransactionID": 0, "ProcessorTransactionRefNumber": "", "CustomerFeeAmount": 0.0, "CurrencyCode": null } }

## eWallet\_RequestUserAutoLogin

This function generates a login key for auto login feature. You can use a login key only once.

### Input Fields:

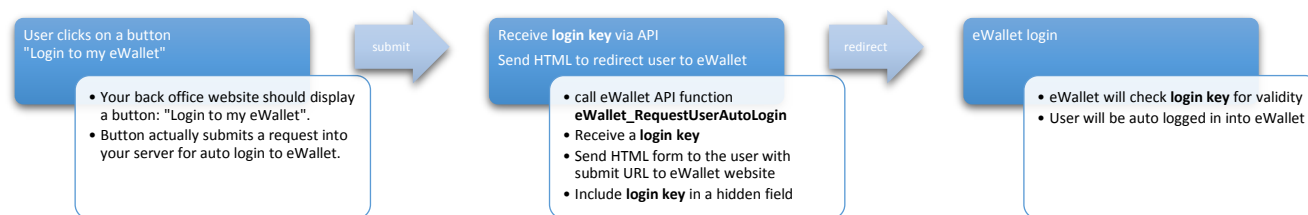
Request Field	Data Type	Required	Comment
MerchantGUID	Guid	Yes	API Merchant ID
MerchantPassword	String	Yes	API Merchant password
UserName	String	Yes	User name

### Return:

Response Field	Data Type	Comment
m_Code	Int (JSON) / string (KVP)	0 or NO_ERROR -> success
m_Text	String	Description of m_code
ProcessorTransactionRefNumber	String	Login key

### Remarks:

This is a workflow chart for auto login.



### HTML form example to push to a user's browser for eWallet auto login

```
<html>
  <body>
    <form id="theForm" action="https://merchantname.globalewallet.com/MemberLogin.aspx" method="post">
      <input type="hidden" name="secKey" value="LOGIN_KEY" />
    </form>
    <script>document.getElementById("theForm").submit();</script>
  </body>
</html>
```

### API call example

```
REQUEST
{"fn":"eWallet_RequestUserAutoLogin","MerchantGUID":"12345678-e2f9-4a01-bef1-258eb99c1b77","MerchantPassword":"12#@Sdr",
"UserName":"ip123"}

RESPONSE (SUCCESS)
{"response":{"m_Code":0,"m_Text":"OK","LogTransactionID":0,"TransactionRefID":0,"ACHTransactionID":0,
"ProcessorTransactionRefNumber":"b0eb897c-4fb2-40ee-bbc6-5a71bcf46fcb","CustomerFeeAmount":0.0,"CurrencyCode":null}}

RESPONSE (FAILURE)
{"response":{"m_Code":-2,"m_Text":"User not found","LogTransactionID":0,"TransactionRefID":0,"ACHTransactionID":0,
"ProcessorTransactionRefNumber":"","CustomerFeeAmount":0.0,"CurrencyCode":null}}
```

## Other useful eWallet API functions

Please refer to a full API document “I-Payout eWallet webService” in <https://www.i-payout.com/api>

**eWallet\_CheckIfUserNameExists** – check if such user is already registered

**eWallet\_DeclineMerchantPayment** – cancel pending invoices

**eWallet\_GetCurrencyBalance** – get eWallet balance

**eWallet\_GetCustomerDetails** – get user details

**eWallet\_GetUserAccountStatus** – get eWallet account status

**eWallet\_GetCheckoutItems** – get current items to checkout

**eWallet\_GetUserRecurringPayins** – get all recurring payins

**eWallet\_GetUserActiveRecurringPayins** - get only active recurring payins

**eWallet\_ClearUserRecurringPayins** – delete all recurring payins

**eWallet\_RemoveUserRecurringPayin** – delete a recurring payin

**eWallet\_ActivateUserRecurringPayin** – activate recurring payin

**eWallet\_AddUserRecurringPayin** – add recurring payin

**eWallet\_UpdateUserRecurringPayin** – update recurring payin

**eWallet\_UpdateUserName** – change user name

**eWallet\_RefundToCustomer** – make a refund. Refunded amount is not reported in a tax form (like 1099)

## Register users in eWallet without API

To register a user in eWallet without API, you need to redirect user’s browser into eWallet page like it is shown in this example (send this HTML to the user once user clicks on a button to register in eWallet):

```
<html>
  <body>
    <form id="theForm" action="https://merchant.globalewallet.com/public/NewUserRegistration.aspx" method="post">
      <input type="hidden" name="UserName" value="username" />
      <input type="hidden" name="hash" value="hashvalue" />
    </form>
    <script>document.getElementById('theForm').submit();</script>
  </body>
</html>
```

**merchant:** merchant name in your eWallet URL. You should receive it in the integration email or contact IPS (compliance@i-payout.com)

**username:** a username of a user to register (from your system: username, user ID, or user’s email)

**hashvalue:** SHA1 hash of: username + MerchantGUID + MerchantPassword

Note: Parameters **username** and **hashvalue** must be submitted using POST. GET method will not be accepted.