

This manual and the functionality described herein may be subject to changes. Please take this into account when implementing the described functionality.

# **Buckaroo Payment Engine 3.0**

## **Implementation Manual**

### **HTML gateway**

Version 1.02 – March, 2012

## Table of contents

1	Buckaroo Payment Engine.....	3
1.1	Payment methods .....	3
1.2	Actions.....	3
1.3	Additional Services .....	3
2	HTML gateway.....	4
2.1	Applications.....	4
2.2	Testing .....	5
2.3	Gateway URLs.....	5
2.4	Mobile user experience.....	5
3	Payment request .....	6
3.1	General fields.....	6
3.2	Specifying services.....	7
3.3	Custom variables .....	7
4	Payment response.....	8
4.1	Parameters .....	8
5	Push response .....	9
5.1	Parameters .....	9
6	Digital signature.....	10
6.1.1	Examples.....	11
7	Gateway styling .....	12
8	Enumerations .....	13
8.1	Gender.....	13
8.2	Status codes.....	13

# 1 Buckaroo Payment Engine

## 1.1 Payment methods

Buckaroo's classic payment solutions consist of separate gateways for each payment method. The new payment solution, called BPE 3.0, has three centralized gateways: one HTML-based gateway which displays payment pages, another HTML-based gateway which handles server-side payment requests (called NVP-gateway) and one SOAP gateway. All payment requests, regardless of payment method, are sent to the same URL. Selection of the payment method is done inside the payment request.

## 1.2 Actions

A payment method implements one or more actions. Most payment methods implement the action 'Pay', while some may also offer the 'Refund' action. Others, such as credit cards, may implement actions such as Authorize and Capture. Please refer to the payment method manual for further information.

## 1.3 Additional Services

Besides a payment method, a payment request can also specify one or more additional services. An example of an additional service is Credit Management.

## 2 HTML gateway

The HTML gateway receives HTTP POST requests and renders an HTML payment form to the consumer (if applicable).

An important aspect of the HTML gateway is the ability to obtain user input outside the webshop. Several payment methods need user data which may be unknown to the merchant, such as credit card numbers. Using the HTML gateway means it is not necessary to collect all required consumer data in the webshop.

The merchant has the option to specify a payment method, or to leave the choice to the customer. If no payment method is specified in the request, the consumer will be offered a choice of all of the merchant’s active payment methods.

Future developments will include the ability to specify payment method categories to adapt the choice of payment methods to the consumer. For instance, if the consumer’s country is known, it is preferable to not show payment methods which are not applicable in the consumer’s country.

### 2.1 Applications

The schematic below shows the possible applications of the HTML gateway:

	Consumer chooses payment method	Merchant chooses payment method	Consumer chooses paym. mthd with addl. service	Merchant chooses paym. mthd with addl. service	
<b>Payment request</b>					
<b>Basic parameters</b>					
Invoice					
Amount					
Currency					
Digital signature					
<b>Additional parameters</b>					
Description					
Language/culture					Fallback to browser language
Return URL					Fallback to Plaza setting
Return URL cancel					Fallback to Plaza setting
Return URL error					Fallback to Plaza setting
Return URL reject					Fallback to Plaza setting
Custom variables					
Payment method					
Action on payment method					
Payment method parameters					Send all parameters to skip user input screen
Additional service					
Action on addl. service					
Additional service					Send all parameters to skip

parameters

user input screen

Required  
field

Optional

## 2.2 Testing

Buckaroo offers a testing environment for testing payment requests and implementations. Refer to the section 'Gateway URLs' for the URL to the test environment. Any request sent to the test environment will not lead to a financial transaction but only offers the possibility of testing communication between the merchant and the gateway.

Our classic payment solution worked with a parameter in the payment request which indicated if the transaction was meant to be treated as a test- or live transaction. This test parameter is no longer available.

Please note that any request sent to the live environment will be treated as such!

## 2.3 Gateway URLs

Live URL: <https://checkout.buckaroo.nl/html/>

Test URL: <https://testcheckout.buckaroo.nl/html/>

## 2.4 Mobile user experience

For users on mobile devices Buckaroo offers a specialized user experience making payment as easy as possible. This includes a standard stylesheet aimed at touch screen devices and optimized input using the most convenient on-screen keyboard for every field.

The gateway automatically offers the most optimal user experience based on the user's browser information (user agent string). However a merchant can force a certain user experience using the field `brq_uxmode`.

### 3 Payment request

Any payment request should contain at least the following parameters: website key, amount, currency, invoice number and the digital signature. A request that contains only these parameters will result in the consumer being offered the choice of payment method.

More general parameters can be supplied to specify the description, culture and varying return URLs. It is also possible to include custom variables by adding parameters prefixed with either `add_` or `cust_`. Refer to the section 'Custom variables' for more information.

All variables must be sent in a POST request. GET requests are not supported.

**Note:** parameter names are not case sensitive (both in the request and in the response). Parameter values *are* case sensitive.

**Important:** only parameters whose names start with `brq_`, `add_` or `cust_` are processed. Any other fields are ignored and not echoed back to the webshop in the payment response. Also when sending a viewstate to the gateway it will result in a technical error. When using .Net you have to make a regular html form post.

#### 3.1 General fields

Parameter name	Description	Required?
Brq_websitekey	The unique key of the website for which the payment is placed.	Yes
Brq_amount	The amount to pay in the format 12.34 (always use a dot as a decimal separator)	Yes
Brq_currency	The currency code (e.g. EUR, USD, GBP). Make sure the specified payment method supports the specified currency.	Yes
Brq_invoicenummer	The unique invoice number that identifies the payment. This is a free text field of max. 255 characters.	Yes
Brq_signature	The digital signature. Refer to section 'Digital signature' for information on calculating the signature.	Yes
Brq_description	A description of the payment to aid the consumer.	No
Brq_culture	ISO culture code that specifies the language and/or country of residence of the consumer. Examples: en-US, en GB, de-DE, EN or DE. The language part of the culture code is used to apply language localization to the gateway. Currently the following languages are supported: NL, EN, DE. When the culture parameter is not supplied, the default culture en-US is used.	No
Brq_return	The return URL where the consumer is redirected after payment. If not supplied, the value specified in the Payment Plaza is used.	No
Brq_returncancel	The return URL used when the consumer cancels the payment. Fallback is the value in <code>brq_return</code>	No
Brq_returnerror	The return URL used when the request results in an error. Fallback is the value in <code>brq_return</code>	No
Brq_returnreject	The return URL used when the payment is rejected by the processor. Fallback is the value in <code>brq_return</code>	No

Brq_requestedservices	A comma separated list of service codes. If no specific service is passed in the field Brq_payment_method, all available services are displayed to a customer. Use this to specify which services should be shown. (Only services with an active subscription are shown)	No
Brq_uxmode	Overrides automatic user experience selection: use 'touch' to force touchscreen UX, use 'normal' to force regular (mouse) UX	No

### 3.2 Specifying services

Parameter name	Description	Required?
Brq_payment_method	The service code for the payment method. Ex.: visa, mastercard, paypal	No
Brq_additional_Service	The service code to one additional service. Ex.: creditmanagement, antifraud This parameter may occur more than once. Only specify one service code per parameter occurrence.	No
Brq_service_[servicecode]_action	The action to execute on the specified service. To invoke the 'Pay' action on service Visa, use the following: brq_service_visa_action=Pay	Yes, if service is specified in request
(parameters) Starting with brq_service_[servicecode]_	Additional parameters for the requested services. Refer to the service manuals for a list of parameters per service.	Refer to specification

### 3.3 Custom variables

It is possible to include custom variables in the payment request, for instance an ID to identify the order in your webshop. Custom variables come in two kinds.

- Additional variables**

Parameters prefixed with add\_ are simply echoed in the payment response and any push notification following the initial request.

- Custom variables (not yet available)**

These variables must be defined in the Payment Plaza, where validation rules can also be specified. The validation rules will then be checked against any incoming custom variables (prefixed with cust\_)

## 4 Payment response

Upon completion of the payment process, the merchant is notified of the payment result via a POST request to the applicable return URL.

### 4.1 Parameters

Parameter name	Description
Brq_payment	Unique key referring to the payment
Brq_payment_method	The service code of the applicable payment method
Brq_statuscode	Code that indicates the status of the payment
Brq_statusmessage	Description of the payment status, localized to the consumer culture
Brq_invoicenumber	The invoice number under which the payment request was entered
Brq_amount	The payment amount (12.34)
Brq_currency	The payment currency(EUR, USD, GBP,...)
Brq_timestamp	Date/timestamp for the payment (yyyy-MM-dd hh:mm:ss)
Brq_transactions	One or more unique keys referring to transactions which are linked to this payment. Multiple keys are separated by a comma (,)
brq_signature	The digital signature. Refer to section `Digital signature` for the calculation method.

The payment response may also contain the following additional parameters:

- Additional or custom variables sent in the payment request
- Output parameters specific to the used payment method, such as account holder data for a Maestro or iDEAL payment

**Important:** the composition of a payment response varies based on the payment method and the payment status (e.g. account holder data is only returned when the payment is successful). Therefore it is recommended to adopt a flexible approach to working with payment responses. Do not assume a fixed set of return parameters.

## 5 Push response

Besides the regular payment response, payment statuses can also be sent asynchronously. This method is convenient when the payment process is interrupted, such as when the consumer closes their browser window before returning to the webshop.

It also notifies of payments that are not completed immediately, such as bank transfers.

Note: whereas the payment response informs of the payment and transaction status, the push response only informs of the transaction status.

In order to receive the push response, the corresponding options must be activated in the Payment Plaza under Profile -> Websites. At least one of the URLs must be entered.

### 5.1 Parameters

Parameter name	Description
brq_transactions	The unique key for the transaction Important: the payment response also contains a parameter named brq_transactions, but may contain multiple transaction keys. The same field in the push response will always contain one single transaction key. For consistence, both fields have the same name.
brq_transaction_method	Service code for the used payment method
brq_statuscode	Status code for the transaction. See table `Statuscodes`
brq_statusmessage	Description of the payment status, localized to the consumer culture
brq_invoicenumber	The invoice number under which the payment request was entered
brq_amount	The payment amount (12.34)
brq_currency	The payment currency(EUR, USD, GBP,...)
brq_timestamp	Date/timestamp for the payment (yyyy-MM-dd hh:mm:ss)
brq_transaction_type	The type of the transaction
brq_signature	The digital signature. Refer to section `Digital signature` for the calculation method.

The push response may also contain the following additional parameters:

- Additional or custom variables sent in the payment request
- Output parameters specific to the used payment method, such as account holder data for a Maestro or iDEAL payment

## 6 Digital signature

To verify the sender of a payment request or a payment response, Buckaroo uses a custom digital signature.

This signature is a hash of all parameters from the payment request or response and a pre-shared key.

This pre-shared secret key must be specified in the Payment Plaza under Configuratie -> Secret key voor Digitale handtekening.

The signature calculation is as follows:

1. List all parameters prefixed with brq\_, add\_ or cust\_, except brq\_signature, and put them in the following format: brq\_parametername=ParameterValue  
**Please note:** *When verifying a received signature, first url-decode all the field values. A signature is always calculated over the non-encoded values (i.e The value "J.+de+Tester" should be decoded to "J. de Tester").*
2. Sort these parameters alphabetically on the parameter name (brq\_amount comes before brq\_websitekey).  
Note: sorting must be case insensitive (brq\_active comes before BRQ\_AMOUNT) but casing in parameter names and values must be preserved.
3. Concatenate all the parameters, formatted as specified under 1, into one string. Do not use any separator or whitespace.  
Example: brq\_amount=1.00brq\_currency=EUR
4. Add the pre-shared secret key at the end of the string
5. Calculate a SHA-1 hash over this string. Return the hash in hexadecimal format.

How to use the SHA1 algorithm depends on your development platform. Most languages and frameworks (such as PHP and ASP.NET) have built-in implementations of the SHA1 algorithm. For other languages, such as classic ASP, implementations of the SHA1 algorithm are available online.

### 6.1.1 Examples

Below are examples of the usage of the SHA1 algorithm in PHP and C# / .NET.

#### PHP

```
$signature3 = sha1($requestFields);
```

#### C# / .NET

```
using System.Text;
using System.Security.Cryptography;

string message = "your sign string here";

SHA1CryptoServiceProvider sha1Provider = new SHA1CryptoServiceProvider();

//convert input string to a byte array
byte[] messageArray = Encoding.UTF8.GetBytes(message);

//calculate hash over the byte array
byte[] hash1 = sha1Provider.ComputeHash(messageArray);

//convert each byte in the hash to hexadecimal format
foreach (byte b in hash1)
{
    builder.Append(b.ToString("x2"));
}
//retrieve the result from the stringBuilder
String Result = builder.ToString();
```

## 7 Gateway styling

By default Buckaroo offers a standard style in two variants: one optimized for desktop users and one optimized for mobile devices.

A merchant can also add their own gateway styling by creating a custom stylesheet. This custom stylesheet can be sent to Technical Support at [support@buckaroo.nl](mailto:support@buckaroo.nl) with mention of the website key under which the stylesheet should be placed.

In order to offer an optimized user experience for mobile device users, a merchant has two options:

- 1) Create a so-called Responsive stylesheet that shows an optimized layout based on screen size
- 2) Offer separate desktop and mobile stylesheets. The HTML gateway will automatically choose the optimal stylesheet if the user is on a device with a touch screen.

We strongly advise to base any custom stylesheets on the standard Buckaroo stylesheets as some gateway functions rely on correct styling.

Please note that custom stylesheets are bound by the following rules:

- 1) Besides a .css file it is also possible to send images of the formats JPEG, PNG or GIF, all residing in the same folder as the stylesheet. Any other file types are not allowed.
- 2) No external links to images, fonts, other stylesheets or any other file outside the Buckaroo domain are allowed inside a custom stylesheet: any references to images should use relative paths.
- 3) Buckaroo will verify any custom stylesheets and associated files. If a file is found to be a security concern Buckaroo retains the right to reject this file.
- 4) Buckaroo accepts no responsibility for the correct functioning of custom stylesheets.
- 5) A default or desktop stylesheet should have the file name `sslgateway.css`
- 6) A touch screen optimized stylesheet should have the filename `sslgateway.touch.css`

## 8 Enumerations

### 8.1 Gender

ISO gender code consisting of one digit.

0	Unkown
1	Male
2	Female
9	Not applicable

### 8.2 Status codes

Code	Description
190	Payment success
490	Payment failure
491	Validation error
492	Technical error
690	Payment rejected
790	Waiting for user input
791	Waiting for processor
792	Waiting on consumer action (e.g.: initiate money transfer)
793	Payment on hold (e.g. waiting for sufficient balance)
890	Cancelled by consumer
891	Cancelled by merchant