



Blesta API Documentation

Note: Last Updated October 29, 2008. This document intended for use with Blesta v. 2.0.0+. If you require documentation for a previous version of Blesta you may find it packaged with your installation.

Introduction:

The Blesta API system allows local and remote access to the Blesta backend system. API calls are made through POST requests. A POST request consists of a single field-value pair. There are two ways to interface with the API, using PHP's serialization or using JSON.

For PHP's serialization the field name must be "base64", and the value sent with this field must be a base64 encoded, serialized array. Similarly, all responses returned from the API system will be base64 encoded, serialized arrays.

For JSON the field name must be "j son", and the value sent with this field must be a base64 encoded, JSON encoded array. Similarly, all responses returned from the API system will be base64 encoded, JSON encoded arrays.

To perform an API call, you must be authenticated. The Blesta API system requires that you pass your Remote Access Key (found in [Settings] > [API/Cron Settings]) for authentication. For this reason it is highly recommended that all API calls be made using SSL. For more information on performing POST requests via SSL please see the PHP CURL documentation at <http://us.php.net/curl>.

There are a series of action calls that may be made to the API. Each action is listed below.

Actions:

- applypayment
- createacct
- createcontact
- createinvoice
- createservice
- formatcurrency
- getclient
- getcoupon
- getcoupons
- getcountries
- getcustomfield
- getcustomfields
- getstates
- getgatesways
- getpackage
- getpackages
- getmodulerow
- getmodulerows
- getservice
- getservicefields
- getservices
- gettransapplied

gettransreceived
moduleRPC
notifyadmins
processpayment
sendwelcome
updatetransaction

Action Descriptions:

- * denotes optional field
- + denotes module specific requirement

applypayment – Applies a payment amount to an account, or an invoice for a particular account.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [applypayment] => true  
  [amount] => 15.00  
  [userid] => 1514  
  [type] => paypal  
  * [transid] => PP10345  
  * [invoiceid] => 1085  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
)
```

Definition:

[amount] – Amount to be applied (Must be in the format ####.##).

[userid] – User ID (int).

[type] – Type of transaction (see Appendix A for complete listing of valid types).

[transid] – Transaction Identifier given by the payment processing gateway.

[invoiceid] – Invoice ID (int).

createacct – Creates a Client Account.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [createacct] => true  
  [userinfo] => Array (  
    [email] => client@blesta.com  
    [fname] => John  
    [lname] => Doe  
    [cname] => Blesta Demo Co.  
    [pass] => mypassword123  
    [address] => 123 East Street  
    [address2] => Apt 3C  
    [city] => Big City  
    * [state] => CA  
    * [zip] => 90000  
    [country] => USA  
    * [phone] => (123) 123-1234  
    * [fax] => (123) 123-1235  
    * [autod] => true  
    * [suspendable] => true  
    * [imethod] => email  
    * [status] => 1  
    * [notes] => Met this client at conference in NYC.  
    * [custom] => Array ([k] => value)  
  )  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Account created.  
  [uid] => 1514  
)
```

Definition:

[userinfo][email] – E-mail Address (requires DNS record, 255 char max).

[userinfo][fname] – First Name (255 char max).

[userinfo][lname] – Last Name (255 char max).

[userinfo][cname] – Company Name/Organization (255 char max).

[userinfo][pass] – Password (6 to 20 char).

[userinfo][address] – Address (255 char max).

[userinfo][address2] – Address 2 (255 char max).

[userinfo][city] – City (255 char max).

[userinfo][state] – State (2 char).

[userinfo][zip] – Zip/Postal Code (255 char max).

[userinfo][country] – Country (3 char).

[userinfo][phone] – Phone Number (255 char max).

[userinfo][fax] – Fax Number (255 char max).

[userinfo][autod] – Auto Debit ("true" or "false").

[userinfo][suspendable] – Auto Suspendable ("true" or "false").

[userinfo][imethod] – Invoice Method (“email” or “paper”).
[userinfo][imethod] – Client status (int) 0 - Inactive, 1 – Active (default), 2 – Fraud.
[userinfo][notes] – Client notes.
[userinfo][custom] – Custom field information **[k]** is the ID of the custom field, **value** is the value.

createcontact – Creates a Contact for a given Client.

Input:

```
Array (
  [key] => abcdefg1234567
  [createcontact] => true
  [contactinfo] => Array (
    [uid] => 1514
    [email] => contact@blesta.com
    [fname] => John
    [lname] => Doe
    *[cname] => Blesta Demo Co.
    [address] => 123 East Street
    *[address2] => Apt 3C
    [city] => Big City
    *[state] => CA
    *[zip] => 90000
    [country] => USA
    *[phone] => (123) 123-1234
    *[fax] => (123) 123-1235
  )
)
```

Returns:

```
Array (
  [response] => success
  [message] => Contact created.
  [cid] => 100
)
```

Definition:

[contactinfo][uid] – Client ID to assign this contact to (int).
[contactinfo][email] – E-mail Address (requires DNS record, 255 char max).
[contactinfo][fname] – First Name (255 char max).
[contactinfo][lname] – Last Name (255 char max).
[contactinfo][cname] – Company Name/Organization (255 char max).
[contactinfo][address] – Address (255 char max).
[contactinfo][city] – City (255 char max).
[contactinfo][country] – Country (3 char).

createinvoice – Creates an Invoice for the given user.

Input:

```
Array (
  [key] => abcdefg1234567
  [createinvoice] => true
  [uid] = 1514
  [deliverinv] = true
  [dateb] = 2008-01-31
  [dated] = 2008-02-15
  *[notes] = Some notes.
  [line] => Array (
    [0] => Line item 1
    [1] => Line item 2
    ...
  )
  *[sid] => Array (
    [0] => 42
    [1] => 41
    ...
  )
  [amount] => Array (
    [0] => 5.95
    [1] => 14.68
    ...
  )
  *[taxable] => Array (
    [0] => 0-true
    [1] => 1-false
    ...
  )
)
```

Returns:

```
Array (
  [response] => success
  [message] => An invoice was successfully created for UID: 1514
)
```

Definition:

[uid] – Client User ID (int).

[deliverinv] – Deliver the Invoice (true/false).

[dateb] – Date Billed (YYYY-MM-DD format).

[dated] – Date Due (YYYY-MM-DD format).

[notes] – Notes (string).

[line][i] – Line item description (string). Will appear in the order given.

[sid][i] – Service ID to attach this line item to (int).

[amount][i] – Amount of line item (####.## format).

[taxable][i] – Applies tax to line i, specified by the string "i-true" where "i" is the line number.

createservice – Creates a Service for a Client.

Input:

```
Array (
  [key] => abcdefg1234567
  [createservice] => true
  [userid] => 1608
  [serviceinfo] => Array (
    [pid] => 3
    [term] => 6
    +[user1]
    +[user2]
    +[pass]
    +[opt1]
    +[opt2]
    +[notes]
    *[pending] => true
  )
  *[createinv] => true
)
```

Returns:

```
Array (
  [response] => success
  [message] => Account created.
  [uid] => 1608
  [serviceid] => 64
)
```

Definition:

[userid] – User ID value (int).

[serviceinfo][pid] – Package ID (int).

[serviceinfo][term] – Term in Months (int).

[serviceinfo][user1] – Module dependant and defined field (string).

[serviceinfo][user2] – Module dependant and defined field (string).

[serviceinfo][pass] – Module dependant and defined field (255 char max).

[serviceinfo][opt1] – Module dependant and defined field (string).

[serviceinfo][opt2] – Module dependant and defined field (string).

[serviceinfo][notes] – Module dependant and defined field (string).

[serviceinfo][pending] – Add the service as pending ("true", "false", or null).

[createinv] – Create Invoice ("true" or "false").

Note: If **[userid]** is not given, the account will first attempt to be created provided the **createacct** action is used. If **[serviceinfo][pending]** is left null the default setting in Blesta will be used for adding this service as pending or immediately.

formatcurrency – Formats a decimal amount into the currency format defined in Blesta.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [formatcurrency] => true  
  [amount] => 1234.56  
  *[pretty] => true  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [amount] => USD $1,234.56  
)
```

Definition:

[amount] – The amount to format (####.## format)

[pretty] – The type of formatting to use ("true" or "false"). Set to true to return prefix and suffix symbols for the currency set in Blesta and, if enabled, the currency code.

getclient – Retrieves all information for a given client

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getclient] => true  
  * [email] => client@blesta.com  
  * [userid] => 1514  
  * [password] => mypassword123  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [client] => Array (  
    [id] => 1514  
    [company] => Blesta Demo Company  
    [fname] => John  
    [lname] => Doe  
    [address] => 123 East Street  
    [address2] => Apt 3C  
    [city] => Big City  
    [state] => CA  
    [zip] => 90000  
    [country] => USA  
    [phone] => (714) 123-1234  
    [fax] => (714) 123-1235  
    [email] => client@blesta.com  
    [signupdate] => 2005-11-30 16:56:02  
    [lastseen] => 2007-08-08 10:55:37  
    [referredby] =>  
    [invoice method] => email  
    [autodebit] => 1  
    [active] => 1  
    [contactid] => 17  
    [password] => 5f4dcc3b5aa765d61d8327deb882cf99  
    [suspendable] => 1  
    [custom] => Array ([k] => Array ([id] => k  
                                     [value] => value  
                                   )  
                      )  
  )  
)
```

Definition:

[email] – E-mail Address. Required if **[userid]** is not defined.

[userid] – User ID. Required if **[email]** is not defined.

[password] – Client Password. If defined, **getclient** will return success if and only if, **[password]** matches the Client's password.

[custom] – Custom field values. **[k]** is the custom field ID.

getcoupon – Retrieves all information for a given coupon code

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getcoupon] => true  
  *[couponid] => 7  
  *[couponcode] => ABCDE  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [coupon] => Array (  
    [id] => 7  
    [code] => ABCDE  
    [discount] => 5.00  
    [used] => 11  
    [max] => 15  
    [start] => 2007-08-02 00:00:00  
    [end] => 0000-00-00 00:00:00  
    [active] => 1  
    [packages] => Array (  
      [0] => 1  
      [1] => 30  
      [2] => 31  
    )  
  )  
)
```

Definition:

[couponid] – Coupon ID. Required if **[couponcode]** is not defined.

[couponcode] – Coupon Code. Required if **[couponid]** is not defined.

Note: The **[coupon][packages]** Array contains the Package IDs to which this coupon applies.

getcoupons – Retrieves all information for all coupons of a type (active/inactive)

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getcoupons] => true  
  *[active] => true  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [coupons] => Array (  
    [0] => Array (  
      [id] => 7  
      [code] => DEF  
      [discount] => 5.00  
      [used] => 11  
      [max] => 15  
      [start] => 2007-08-02 00:00:00  
      [end] => 0000-00-00 00:00:00  
      [active] => 1  
      [packages] => Array (  
        [0] => 1  
        [1] => 30  
        [2] => 31  
      )  
    )  
    ...  
    ...  
  )  
)
```

Definition:

[active] – “true” or “false” Determines what type of coupon to get.

Note: The **[coupons][x][packages]** Array contains the Package IDs to which each coupon applies.

getcountries – Retrieves all countries.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getcountries] => true  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [countries] => Array (  
    [0] = Array (  
      [id] => 1  
      [name] => Afghani stan  
      [2char] => AF  
      [3char] => AFG  
      [allow] => false  
    )  
    ...  
  )  
)
```

Note: **[countries][x][3char]** should ALWAYS be used when posting to Blesta.
[countries][x][allow] is currently not being used.

getcustomfield – Returns details for a given custom field.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getcustomfield] => true  
  [id] => 3  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [customfield] => Array (  
    [id] => 3  
    [name] => Mobile Phone  
    [type] => text  
    [regex] => [0-9]{7,10}  
    [showclient] => 1  
    [typename] => Text box  
    [required] => 0  
  )  
)
```

Note: **[regex]** is the regular expression used to validate that the field was entered correctly. **[type]** is the field type that should be used to display this field. If the field is a select field it may also contain a **[valuesarray]** element that contains an array of all available values to select from.

getcustomfields – Returns details for all custom fields.

Input:

```
Array (
  [key] => abcdefg1234567
  [getcustomfields] => true
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
  [customfields] => Array (
    [k] = Array (
      [id] => k
      [name] => Mobile Phone
      [type] => text
      [regex] => [0-9]{7,10}
      [showclient] => 1
      [typename] => Text box
      [required] => 0
    )
    ...
  )
)
```

Note: **[k]** is the ID for that custom field, therefore when iterating through you should use **foreach**.
[regex] is the regular expression used to validate that the field was entered correctly.

getstates – Retrieves all states.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getstates] => true  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [states] => Array (  
    [0] = Array (  
      [id] => 1  
      [name] => Al aska  
      [2char] => AK  
    )  
    ...  
  )  
)
```

Note: **[states][x][2char]** should ALWAYS be used when posting to Blesta. Optionally, if **[getcountries]** is passed in conjunction with **[getstates]**, both countries and states will be returned together.

getgateways – Returns merchant and non-merchant gateway information.

Input:

```
Array (
  [key] => abcdefg1234567
  [getgateways] => true
  [type] => all
  *[orderinfo] => Array (
    [uid] = 1514
    *[iid] = 1023
    [amount] = 5.00
  )
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
  [gateways] => Array (
    [nonmerchant] = Array (
      [paypal] => Array (
      )
      [google checkout] => Array (
      )
      ...
    )
    [merchant] = Array (
      [username] => Array (
        [value] => yourloginid
        [friendlyName] => Login ID
      )
      [key] => Array (
        [value] => yourkey
        [friendlyName] => Transaction Key
      )
      [testmode] => Array (
        [value] => True
        [friendlyName] => Test Mode
      )
      [currency] => USD
      [filename] => authnet.class.php
      [name] => AuthorizeNet
      [handle] => authnet
      [merchant] => true
    )
  )
)
```

Definition:

[type] – “all”, “merchant”, or “nonmerchant”.

[orderinfo] – Required if a nonmerchant gateway is requested.

[orderinfo][iid] – Invoice ID (int). Required if processing payment for a specific invoice.

Note: The information returned in **[gateways][nonmerchant][x]**, where “x” is the name of any given non-merchant gateway is completely dependant on the non-merchant gateway class file used to create the gateway. Generally, however, the data contained here relates to the hidden form fields required by the non-merchant gateway processor.

getpackage – Retrieves all information for a given package.

Input:

```
Array (
  [key] => abcdefg1234567
  [getpackage] => true
  [packageid] => 4
  *[couponcode] => ABC
  *[status] => active
  *[formatcurrency] => false
  *[userid] => 1514
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
  [package] => Array (
    [id] => 4
    [name] => Bronze Hosting
    [status] => 1
    [notes] =>
    [instantact] => true
    [module] => cpanel
    [modulestring] => bronze
    [pricing] => Array (
      [1] => Array (
        [full] => 10.00
        [price] => 9.00
        [discount] => 10.00
      )
      [3] => Array (
        [full] => 30.00
        [price] => 27.00
        [discount] => 10.00
      )
      [12] => Array (
        [full] => 120.00
        [price] => 108.00
        [discount] => 10.00
      )
    )
    [taxable] = true
    [tax] = 15.00
  )
)
```

Definition:

[packageid] – Package ID (int).

[couponcode] – Coupon Code (255 char max).

[status] – Package Status (“active” or “inactive”).

[formatcurrency] – Formats the values under [pricing] according to the currency set in Blesta.

[userid] – User ID (int). If given will determine the percentage of tax to be charged for this package (if applicable).

Note: **[pricing][x][price]** should ALWAYS be used, whether or not a coupon is present, to display the current charge for any given item, where x is the term of service in months. All prices are WITHOUT tax.

getpackages – Retrieves all information for all packages.

Input:

```
Array (
  [key] => abcdefg1234567
  [getpackages] => true
  * [status] => active
  * [modulename] => cpanel
  * [instantact] => true
  * [formatcurrency] => false
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
  [packages] => Array (
    [0] => Array (
      [id] => 4
      [name] => Bronze Hosting
      [status] => 1
      [notes] =>
      [instantact] => true
      [module] => cpanel
      [modulestring] => bronze
      [pricing] => Array (
        [1] => 10.00
        [3] => 30.00
        [12] => 120.00
      )
      [taxable] = true
    )
    ...
    ...
  )
)
```

Definition:

[status] – Package Status (“active” or “inactive”).

[modulename] – Module Name.

[instantact] – Instant Activation (“true” or “false”).

[formatcurrency] – Formats the values under [pricing] according to the currency set in Blesta.

getmodul erow – Retrieves all information for a row in a given module’s table (e.g. a cPanel server).

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getmodul erow] => true  
  [modul ename] => cpanel  
  [modul ei d] => 3  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [modul e] => Array (  
    [i d] => 3  
    +[x]  
    ...  
    ...  
  )  
)
```

Definition:

[modulename] – Module Name.

[moduleid] – Module row ID (int).

getmoduleroes – Retrieves all information from a given module's table.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getmoduleroes] => true  
  [modulename] => cpanel  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [modules] => Array (  
    [0] => Array (  
      [id] => 3  
      +[x]  
      ...  
    )  
    ...  
  )  
)
```

Definition:

[modulename] – Module Name.

[moduleid] – Module row ID (int).

getservice – Returns an Array containing all the service information for a given serviceid.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [gerservice] => true  
  [serviceid] => 16  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [service] => Array (  
    [serviceid] => 16  
    [packageid] => 24  
    [userid] => 1514  
    [moduleid] => 16  
    +[user1] =>  
    +[user2] =>  
    +[pass] =>  
    +[opt1] =>  
    +[opt2] =>  
    +[notes] =>  
    [datestarted] => 2007-07-16  
    [datedeleted] => 0000-00-00  
    [daterenews] => 2007-10-14  
    [datesuspended] => 0000-00-00  
    [term] => 3  
    [adminid] => 0  
  )  
)
```

Definition:

[serviceid] – Service ID (int)

Notes: **[service][adminid]** is 0 when the service was created automatically, and not directly by an admin.

getservicefields – Returns an Array containing all the service information for a given serviceid.

Input:

```
Array (
  [key] => abcdefg1234567
  [gerservicefields] => true
  [module] => cpanel
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
  +[cpanel] => Array (
    [fields] => Array (
      [0] => Array (
        +[type]
        +[dbFieldName]
        +[friendlyName]
        +[option] => Array (
          [0] => Array (
            +[name]
            +[value]
          )
          ...
        )
        +[regex]
        +[errorMessage]
      )
      ...
    )
    [instantact] => true
  )
)
```

Definition:

[userid] – User ID (int)

Notes: The **[cpanel][fields][i][option]** Array is only present on fields where **[type]** is "select". See Appendix A for more information on HTML field types. Please note that **[cpanel]** is used here as an associative index, but depending on the module you request this index may vary.

getservices – Returns an Array containing all services for a given userid.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [getservices] => true  
  [userid] => 1514  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [services] => Array (  
    [0] => Array (  
      [serviceid] => 16  
      [packageid] => 24  
      [userid] => 1514  
      [moduleid] => 16  
      +[user1] =>  
      +[user2] =>  
      +[pass] =>  
      +[opt1] =>  
      +[opt2] =>  
      +[notes] =>  
      [datestarted] => 2007-07-16  
      [datedeleted] => 0000-00-00  
      [daterenews] => 2007-10-14  
      [datesuspended] => 0000-00-00  
      [term] => 3  
      [adminid] => 0  
    )  
    ...  
    ...  
  )  
)
```

Definition:

[userid] – User ID (int).

Notes: **[service][i][adminid]** is 0 when the service was created automatically, and not directly by an admin.

gettransapplied – Returns an Array containing the applied details for a given transaction.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [gettransapplied] => true  
  [uid] => 1514  
  [rid] => 327  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [applied] => Array (  
    [0] => Array (  
      [rid] => 327  
      [uid] => 1514  
      [cid] => 24  
      [iid] => 1002  
      [amount] => 14.68  
      [datetime] => 2008-01-31 22: 15: 03  
    )  
    ...  
    ...  
  )  
)
```

Definition:

[uid] – User ID (int).

[rid] – Received transaction ID (int).

[cid] – Contact ID used to process the transaction.

[amount] – Amount of transaction (####.## format).

[datetime] – Date/time when the transaction was applied.

Notes: **[applied]** may return 0 or more elements, depending on how many invoices the transaction was applied to.

gettransreceived – Returns an Array containing transactions received for (optionally) a given user.

Input:

```
Array (
  [key] => abcdefg1234567
  [gettransapplied] => true
  *[uid] => 1514
  *[count] => false
  *[page] => 1
  *[status] = approved
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
  [received] => Array (
    [0] => Array (
      [rid] => 327
      [uid] => 1514
      [cid] => 24
      [amount] => 14.68
      [credited] => 0.00
      [applied] => 14.68
      [type] => credit
      [transid] => 123ABC
      [status] => approved
      [message] => APPROVED
      [date] => 2008-01-31 22:15:03
    )
    ...
    ...
  )
)
```

Definition:

[uid] – User ID (int).

[count] – If true the number of transactions given by the input will be returned in [received] (true/false).

[page] – Specifies the page of results to return. Pages are defined in /inc/config.php by LIST1

[status] – Status of the transaction (approved, error, other).

[rid] – The received transaction ID.

[cid] – The contact ID used to process the transaction.

[amount] – Amount of the transaction (####.## format)

[type] – One of various PAYTYPES as defined in /inc/config.php.

[status] – The status of the transaction (approved, declined, error, pending, voided)

[message] – The message returned by the payment gateway for this transaction.

[date] – Date/time when the transaction was received.

moduleRPC – Allows the execution of any public method contained in the given module.

Input:

```
Array (
    [key] => abcdefg1234567
    [moduleRPC] => true
    [module] => enom
    [method] => checkdomain
    [param] => blesta.com
)
```

Returns:

```
Array (
    [response] => success
    [message] => Success.
    +[returned] => false
)
```

Definition:

[module] – Module Name.

[method] – Method name as defined in the module class file (e.g. /inc/modules/**module**.class.php).

[param] – Parameter being passed to the method.

Note: **[returned]** may be anything, and is completely dependent upon the module. Only one **[param]** may be passed, but this may be of any non-Object type (i.e. Array, String, etc..).

notifyadmins – Sends Order Failure Notice or Order Success Notice e-mails to all Admins set to receive notices.

Input:

```
Array (
  [key] => abcdefg1234567
  [notifyadmins] => true
  [status] => failure
  *[orderinfo] => Array (
    [pid] => 14
    [packagename] => Bronze Hosting
    [term] => 3
    [serviceinfo] => Domain Name: blesta.com
    Password: mypassword123
    [uid] => 1514
    [fname] => John
    [lname] => Doe
    [cname] => Blesta Demo Co
    [email] => client@blesta.com
    [address] => 123 East Street
    [address2] => Apt 3C
    [city] => Big City
    [state] => CA
    [country] => USA
    [zip] => 90000
    [amount] => 29.95
    [transid] => PP1234
    [mailid] => 107
  )
)
```

Returns:

```
Array (
  [response] => success
  [message] => Success.
)
```

Definition:

[status] – Order Status (“success” or “failure”).

[orderinfo][pid] – Package ID ordered (int).

[orderinfo][packagename] – Name of the Package Ordered (255 char max).

[orderinfo][term] – Term in months (int).

[orderinfo][serviceinfo] – Text to replace [serviceinfo] tag in E-mail template. Usually the field name defined for a given package’s HTML field followed by the value entered by the user for that field (string).

[orderinfo][uid] – User ID (int).

[orderinfo][fname] – First Name (255 char max).

[orderinfo][lname] – Last Name (255 char max).

[orderinfo][cname] – Company Name/Organization (255 char max).

[orderinfo][email] – E-mail Address (255 char max).

[orderinfo][address] – Address (255 char max).

[orderinfo][state] – State/Province (2 char).

[orderinfo][country] – Country (3 char).

[orderid][zip] – Zip/Postal Code (255 char max).
[orderid][amount] – Amount of the order (must in be #####.## format).
[orderid][transid] – Transaction Identifier given by the payment processing gateway.
[orderid][mailid] – The Mail ID in the maillog table from that the last payment success e-mail.

processpayment – Process a Credit Card Payment.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [processpayment] => true  
  [userinfo] => Array (  
    [fname] => John  
    [lname] => Doe  
    [email] => client@blesta.com  
    [address] => 123 East Street  
    *[address2] => Apt 3A  
    [city] => Big City  
    *[state] => CA  
    *[zip] => 90000  
    [country] => USA  
    [ccno] => 4111111111111111  
    [ccexp] => 0813  
  )  
  [amount] => 39.95  
  [emailreceipt] => true  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
  [mailid] => 136  
)
```

Definition:

[userinfo][fname] – First Name (255 char max).
[userinfo][lname] – Last Name (255 char max).
[userinfo][email] – E-mail Address (255 char max).
[userinfo][address] – Address (255 char max).
[userinfo][address2] – Address 2 (255 char max).
[userinfo][city] – City (255 char max).
[userinfo][state] – State (2 char).
[userinfo][zip] – Zip/Postal Code (255 char max).
[userinfo][country] – Country (3 char).
[userinfo][ccno] – Credit Card Number.
[userinfo][ccexp] – Credit Card Expiration (in mmyy format).
[amount] – Amount of transaction (in #####.## format).
[emailreceipt] – E-mail receipt (“true” or “false”).

Note: **[mailid]** will only be returned if **[emailreceipt]** is set to "true", and the payment was processed successfully.

sendwelcome – Sends Welcome E-mail.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [sendwelcome] => true  
  [userid] => 1514  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
)
```

Definition:

[userid] – User ID (int)

updatetransaction – Updates a transaction with the given userid. This optionally updates the mail log with the proper userid for the transaction e-mail response.

Input:

```
Array (  
  [key] => abcdefg1234567  
  [updatetransaction] => true  
  [userid] => 1514  
  *[transid] => 102  
  *[mailid] => 223  
)
```

Returns:

```
Array (  
  [response] => success  
  [message] => Success.  
)
```

Definition:

[userid] – User ID (int)

[transid] – Transaction ID to be updated with the User ID (int)

[mailid] – Mail ID to be updated with the User ID (int)

Appendix A

HTML Types: checkbox, password, radio, select, text, textarea

Payment Types: May vary. Please see the PAYTYPES define in your installation's /inc/condef.php file.