BorderGuru
Technical Integration

This document describes all options for merchants
to use the cross border logistics service of BorderGuru

Contact information:
info@borderguru.com
+49 40 2261 62860

1. Abstract

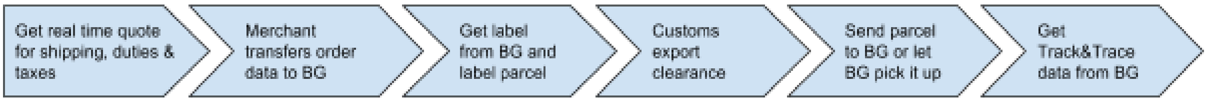
BorderGuru offers a cross border logistics service empowering online retailers to internationalize their web shop. With BorderGuru online retailers can offer their customers global delivery with full customs service and guaranteed prices for customs duties & taxes as well as shipping. This document describes all possible options how you can start using the BorderGuru service.

2. Table of contents

- 1. Abstract
- 2. Table of contents
- 3. General process overview
- 4. Get quotes for shipping, duties & taxes
- 5. Transfer of order data
- 6. Getting shipping labels
- 7. Export customs clearance
- 8. Sending your parcels
- 9. Get Track&Trace information
- 10. Technical reference
    - 10.1. Getting quotes for your shopping carts
        - 10.1.1. Using our javascript frontend API
        - 10.1.2. Using our REST API
    - 10.2. Transfer of customer order data
        - 10.2.1. Using Google Spreadsheet
        - 10.2.2. Using the javascript frontend API
        - 10.2.3. Using the REST API
    - 10.3. Getting the label
        - 10.3.1. Label client
        - 10.3.2. Labeling REST API
    - 10.4. Export customs clearance
    - 10.5. Track & Trace REST API

3. General process overview

In general the process to use BorderGuru as your cross border shipping provider follows the following steps:


Get real time quote for shipping, duties & taxes
Merchant transfers order data to BG
Get label from BG and label parcel
Customs export clearance
Send parcel to BG or let BG pick it up
Get Track&Trace data from BG

1. Quotes for shipping, duties & taxes:
   We can provide quotes with guaranteed prices for shipping, duties & taxes. This step is completely optional, but you as a merchant can use this to enable your customers to pay duties & taxes upfront to avoid customs clearance. Some merchants prefer not to display the duties & taxes that need to be paid for the import into the destination country but rather include that in the shipping costs that are displayed to their customers. Both options are possible with BorderGuru.

2. Transfer of order data:
   After the optional quoting everything starts with the transmission of customer order data to BorderGuru. BorderGuru needs this data to ship your parcels to your customers and to do all customs clearance.

3. Get shipping labels:
   After BorderGuru has the customer order data we can provide you with the shipping label that you can use to label your parcels. Alternatively you can also label your parcels with your order ID and we do a relabeling with the final shipping labels for you.

4. Export customs clearance
   In some countries the merchant needs to declare the export to the domestic customs. BorderGuru will do that for you, but first needs to have some additional information from you. If your country’s customs do not require this, then this step can be left out.

5. Get parcels to the BG warehouse:
   Next step is to get your labeled parcels to the next BorderGuru export warehouse. We can either pick them up at your logistics center or you send them to our export warehouse on your own.

6. Get track & trace data:
   After your parcel was scanned in the BorderGuru export warehouse we can provide you or your customers with end-to-end track & trace information, so you and your customers always know where the parcel currently is.

4. Get quotes for shipping, duties & taxes

As a first step you can use our quoting interface to provide your customer with a guaranteed prices for shipping, duties & taxes. This step is completely optional, but enables your customer to already have guaranteed prices upfront and also pay these costs upfront to avoid customs clearance. It is also ok, if you hide the duties & taxes in the shipping costs displayed to your customers, if you like to do that. To use our quoting interface you have two options:

A) Using our javascript frontend API
Our javascript frontend API can directly be integrated into your checkout. It works very similar to other 3rd party frontend integrations like for example the eCommerce extension of Google Analytics, which is very familiar for most developers.

B) Using our REST API
Our REST API can be used from virtually any technical platform as it means that you can just call our quoting web service to receive the quotes.

For both options you basically send the content of your customer's shopping cart to the BorderGuru system and our system replies with a guaranteed quote for the duties & taxes together with the shipping costs that would apply to that certain customer order.

Please find detailed technical instruction by following the links above.

5. Transfer of order data

In order to transport your parcels to your customers BorderGuru needs to get certain customer order data. BorderGuru offers several options to submit your order data into the BorderGuru system:

A) Using Google Spreadsheet
For the easiest start we can provide you with a Google Spreadsheet where you can enter your order data. This way you do not need any kind of technical integration. This is the easiest way to get started, but should only be used for starting early and on a low volume level. For much tighter integration and better automation we recommend technical integration with one of the below options.

B) Using the javascript frontend API
With our javascript frontend API you can submit order data to BorderGuru directly from your web shop's checkout. The integration is very similar to the integration of other 3rd party integrations like for example the eCommerce extension of Google Analytics, which is very familiar for most developers.

C) Using the REST API
Our REST API can be used from virtually any technical platform to submit customer's order data. All other integrations options are internally based on our REST API thus this gives you the most possible flexibility on the costs of a slightly higher integration effort.

In all cases you basically send the detailed order data to the BorderGuru system. Detailed order data is everything that is needed by BorderGuru to deliver your parcel to your your customer's address. This includes the shipping and the billing address of your customer, the physical attributes of the shipment like dimensions and weight and the products that were ordered. The BorderGuru system then replies with a shipment identifier, which you can use as a reference of the shipping in further steps, e.g. to download a label for the shipment.

Please find detailed technical instruction by following the links above.

6. Getting shipping labels

After the customer order data is transmitted to the BorderGuru, we can provide you with shipping labels that you can put on your parcels. There are two options for you to get the shipping labels:

A) Using the BorderGuru label client
The BorderGuru label client is a web application you can use to enter an order ID and get a ready to be printed shipping label as PDF.

B) Using the BorderGuru labeling REST API
Alternatively you can also use the labeling REST API to fully integrate the label creation into your logistics systems.

Please find detailed technical instruction by following the links above.

However this is completely optional for you! Instead using our shipping labels you can also use any label with your order ID and we can do a relabeling with the final shipping labels in our export warehouse for you.

7. Export customs clearance

In some countries the domestic customs require the merchant to declare an export. BorderGuru provides this service for you, but you need to trigger this from your side. For this BorderGuru offers a REST API which you need to use to declare which parcels you want to export on a daily base.
If your domestic customs do not require this step, then this can be left out.

Please find detailed technical instructions here:
Customs export clearance

8. Sending your parcels

As all your parcels now have labels, they can be shipped to our export warehouse, where we consolidate the shipments and prepare them for export. You have two options:

A) We pick-up the parcels from your warehouse

B) You organise the transport to our export warehouse yourself

For both options, an e-mail pre-alert is necessary for us to know which shipments are coming in and potentially organize the export declaration for you with the domestic customs, if required.

9. Get Track&Trace information

As soon as your parcels arrived in our export warehouse we can provide you or your customers with end-to-end track & trace information. Again you have several options here:

A) We send an email to your customers with a track & trace link

B) We send an email with a track & trace link to you

C) You get the track & trace link via our Shipping API and can send it to your customer

D) You can use our track & trace REST API to get all track & trace events
This is the most flexible option for you. By using our track & trace REST API you can get all track & trace events directly into your system. You can use this data to inform your customers in any way you like about track & trace events.

10. Technical reference

10.1. Getting quotes for your shopping carts

If you like you can display the duties & taxes to be paid for a certain order to your customers directly in your checkout process on your web shop. You can collect this money directly along with the order placement of your customers to guarantee your customers that they will not be affected by customs clearance. Technically you have two options to integrate the duties & taxes quotes into your webshop. Either by using our javascript frontend API or by using our quoting REST API.

10.1.1. Using our javascript frontend API

The javascript frontend API can be integrated directly into the frontend code of your checkout and is used in a very similar way as the Google Analytics eCommerce extension. Being like that most developers will feel very familiar with it.
The BorderGuru frontend API is depending on jQuery. So if you do not already have jQuery loaded, please make sure it is loaded!

Example code:

```
<script src="://ajax.googleapis.com/ajax/libs/jquery/2.2.2/jquery.min.js"></script>
<script src="://app.borderguru.com/api/js?apikey=<your-key>">
<!-- // <your-key> should be the merchant or store identifier

/* create a new merchant order */
var order = new bg.order();

/*** fields of * in comment are mandatory ***/

/* merchant information */
order.storeName        = "", // *name of your webshop
order.merchantIdentifier    = "", // *your merchant id provided by BG
order.merchantOrderId  = "", // your order ID
order.submerchant = {               // information about submerchant
       "company":            "", // *name of company
       "streetName":         "", // *street name
       "houseNo":            "", // *house number
       "postcode":           "", // *postalcode
       "city":               "", // *city
       "state":              "", // *state
       "countryCode":        "", // *country code, 2-char
       "EORI":               ""  // *DE export registration number
},

/* parcel measurements */
order.totalWeight              = 0,  // * weight of the parcel
order.totalWeightScale         = "", // * weight unit, kg, lb
order.dimensionalWeight        = 0,  // dimensional weight
order.dimensionalWeightScale = "", // dimensional weight unit
order.dimensions = {
       "width":                    0,  // wdith of the parcel
       "length":                   0,  // length of the parcel
       "height":                   0,  // height of the parcel
       "dimensionalScale":         ""  // dimension unit, cm, in
},

/* order prices */
order.subtotal               = 0,  // * order price without shipping
order.shippingCost           = 0,  // shipping price
order.totalPrice             = 0,  // * total price
order.currency               = "", // * 3-char currency
order.exportCurrency         = "", // currency used for export
```

```
/* origin and destination */
order.countryOfOrigin          = "", // * origin country 2-char
order.countryOfDestination   = "", // * destination country 2-char

/* id card verification for china import */
order.identificationData = {
        "idType":                    "0",// always 0
        "identificationNumber":      "", // id card number
        "identificationName":        "", // id card name
},

/* shipping address */
order.shippingAddress = [{
        "firstName":                 "", // first name
        "lastName":                  "", // last name
        "streetName":                "", // street name
        "houseNo":                   "", // house number
        "additionalInfo":            "", // additional address info
        "postcode":                  "", // postalcode
        "city":                      "", // city
        "country":                   "", // country name
        "telephone":                 "", // telephone number
        "email":                     "", // email address
        "countryCode":               ""   // country 2-char
}],

/* billing address */
order.billingAddress = [{
        "firstName":                 "", // first name
        "lastName":                  "", // last name
        "streetName":                "", // street name
        "houseNo":                   "", // house number
        "additionalInfo":            "", // additional address info
        "postcode":                  "", // postalcode
        "city":                      "", // city
        "country":                   "", // country name
        "telephone":                 "", // telephone number
        "email":                     "", // email address
        "countryCode":               ""   // country 2-char
}],

/* order lines */
order.lineItems = [{
        "sku":                       "", // * product SKU
        "shortDescription":          "", // * short description
        "quantity":                  0,  // * quantity
        "price":                     0,  // * selling price
        "category":                  "", // * category name
        "weight":                    0,  // * weight
        "weightScale":               "", // * weight unit
        "hsCode":                    "", // HS code for import
        "exportHsCode":              "", // HS code for export
        "exportPrice":               0,  // price for export
        "exportDescription":         ""   // description for export
}]

/* request quote */
var quote = order.quote();

/* evaluate API responses */
quote.done = function(response) {/* your code to evaluate response */}
quote.fail = function(response, status) {/* your code to evaluate response */}
quote.timeout function() {/* your code to react on timeouts */
```

10.1.2. Using our REST API

To get a quote for the duties & taxes for a certain customer order you can use the following API:

Endpoint:

```
POST https://app.borderguru.com/api/quotes/calculate?apikey=<API access key>
```

Please substitute <API access key> with the API access key that we provided you.

Header:

```
Content-Type: application/json
```

Request body:

```
{
       "merchantIdentifier": "0123456789abcdef01234567",  // string, mandatory

       "subtotal":                  79.7,                        // float, mandatory
       "shippingCost":              10,                          // float, optional
       "totalPrice":                92.4034,                     // float, mandatory
       "currency":                  "USD",                       // string, mandatory

       "countryOfOrigin":           "US",                        // string, mandatory
       "countryOfDestination":      "CN",                        // string, mandatory

       "totalWeight":        3.87,                        // float, mandatory
       "totalWeightScale":          "kg",                        // string, mandatory
       "dimensionalWeight":         1.95,                        // float, optional
       "dimensionalWeightScale":    "kg",                        // string, optional
       "dimensions": {                                           // object, optional
              "length":             10.5,                        // float, optional
              "width":              1.8,                         // float, optional
              "height":             21.3,                        // float, optional
              "dimensionalScale":   "cm",                        // string, mandatory
       },


       "lineItems": [{
              "sku":                "T14K00357",          // string, mandatory
              "shortDescription":          "Awesome E-Reader",   // string, mandatory
              "price":                     10,                   // float, mandatory
              "category":                  "E-readers",          // string, mandatory
              "weight":                    0.57,                 // float, mandatory
```

```
                "weightScale":          "kg",                   // string, mandatory
                "quantity":                     3,                      // float, mandatory
                "countryOfManufacture":         "CN",                   // string, mandatory
                "hsCode":                       "9202903000",           // string, optional
        },
        // of course you can add more line items here
        ]
}
```

Request body needs to be json as raw body in UTF-8

Response:

```
{
        "response": {
                "success":      true,
                "result": {
                        "quoteIdentifier":      "0123456789abcdef01234567",
                        "lineItems": [{
                                "sku":  "T14K00357",
                                "shortDescription":             "SSEw1jx4zP",
                                "quantity":                     3,
                                "price":                        10,
                                "weight":                       0.57,
                                "weightScale":          "kg",
                                "duty":                 0,
                                "tax":                          8,
                                },
                        // over here additional items would be shown
                        ],
                        "taxAndDutyCost":       8,
                        "shippingCost":         10,
                        "quoteCurrency":        "USD",
                        "quoteSubtotal":        30,
                        "total":                48t
                }
        }
}
```

10.2. Transfer of customer order data

To be able to ship your parcels directly to your customers BorderGuru needs to have certain customer order data like for example ordered items and customer’s delivery address. There are several options for you to submit the necessary order data to BorderGuru.

10.2.1. Using Google Spreadsheet

The easiest way to start fast is probably to use a Google spreadsheet that we can provide for you. In this spreadsheet you can enter all order details needed to ship your parcels and also directly transfer this order data to BorderGuru. Please contact us, if this is an interesting option for you and we will provide you with a copy of the spreadsheet.

10.2.2. Using the javascript frontend API

The javascript frontend API can be integrated directly into the frontend code of your checkout and is used in a very similar way as the Google Analytics eCommerce extension. Being like that most developers will feel very familiar with it.

Example:

```
<script src="://ajax.googleapis.com/ajax/libs/jquery/2.2.2/jquery.min.js"></script>
<script src="://app.borderguru.com/api/js?apikey=<your-key>">
<!-- // <your-key> should be the merchant or store identifier

/* create a new merchant order */
var order = new bg.order();

/*** fields of * in comment are mandatory ***/

/* merchant information */
order.storeName        = "", // *name of your webshop
order.merchantIdentifier    = "", // *your merchant id provided by BG
order.merchantOrderId  = "", // your order ID
order.submerchant = {               // information about submerchant
        "company":           "", // *name of company
        "streetName":        "", // *street name
        "houseNo":           "", // *house number
        "postcode":          "", // *postalcode
        "city":              "", // *city
        "state":             "", // *state
        "countryCode":       "", // *country code, 2-char
        "EORI":              ""  // *DE export registration number
},

/* parcel measurements */
order.totalWeight              = 0,  // * weight of the parcel
order.totalWeightScale         = "", // * weight unit, kg, lb
order.dimensionalWeight        = 0,  // dimensional weight
order.dimensionalWeightScale = "", // dimensional weight unit
order.dimensions = {
        "width":                     0,  // wdith of the parcel
        "length":                    0,  // length of the parcel
        "height":                    0,  // height of the parcel
        "dimensionalScale":          ""  // dimension unit, cm, in
},

/* order prices */
order.subtotal               = 0,  // * order price without shipping
order.shippingCost           = 0,  // shipping price
order.totalPrice             = 0,  // * total price
order.currency               = "", // * 3-char currency
order.exportCurrency         = "", // currency used for export

/* origin and destination */
order.countryOfOrigin          = "", // * origin country 2-char
order.countryOfDestination   = "", // * destination country 2-char
```

```
/* id card verification for china import */
order.identificationData = {
        "idType":                   "0",// always 0
        "identificationNumber":     "", // id card number
        "identificationName":       "", // id card name
},

/* shipping address */
order.shippingAddress = [{
        "firstName":                "", // *first name
        "lastName":                 "", // *last name
        "streetName":               "", // *street name
        "houseNo":                  "", // house number
        "additionalInfo":           "", // additional address info
        "postcode":                 "", // *postalcode
        "city":                     "", // *city
        "state":                    "", // state
        "country":                  "", // country name
        "telephone":                "", // telephone number
        "email":                    "", // *email address
        "countryCode":              ""  // *country 2-char
}],

/* billing address */
order.billingAddress = [{
        "firstName":                "", // *first name
        "lastName":                 "", // *last name
        "streetName":               "", // *street name
        "houseNo":                  "", // house number
        "additionalInfo":           "", // additional address info
        "postcode":                 "", // *postalcode
        "city":                     "", // *city
        "state":                    "", // state
        "country":                  "", // country name
        "telephone":                "", // telephone number
        "email":                    "", // *email address
        "countryCode":              ""  // *country 2-char
}],

/* order lines */
order.lineItems = [{
        "sku":                      "", // *product SKU
        "shortDescription":         "", // *short description
        "quantity":                 0,  // *quantity
        "price":                    0,  // *selling price
        "category":                 "", // *category name
        "weight":                   0,  // *weight
        "weightScale":              "", // *weight unit
        "hsCode":                   "", // HS code for import
        "exportHsCode":             "", // HS code for export
        "exportPrice":              0,  // price for export
        "exportDescription":        ""  // description for export
}]

/* request quote */
var ship = order.ship();

/* evaluate API responses */
ship.done = function(response) {/* your code to evaluate response */}
ship.fail = function(response, status) {/* your code to evaluate response */}
ship.timeout function() {/* your code to react on timeouts */
```

10.2.3. Using the REST API

To enable BorderGuru to ship your parcels you need to transmit certain customer order data to BorderGuru. You can do that using the following REST API:

Endpoint:

```
POST https://app.borderguru.com/api/shipping?apikey=<API access key>
```

Please substitute <API access key> with the API access key that we provided you.

Header:

```
Content-Type: application/json
```

Request body:

```
{
        "quoteIdentifier":           "0123456789abcdef01234567",  // string, [1]
        "merchantOrderId":           "0123456790ABC",             // string, optional
        "storeName":                 "My WebShop",                // string, mandatory
        "merchantIdentifier": "0123456789abcdef01234567",  // string, mandatory

        "subtotal":                  79.7,                        // float, mandatory
        "shippingCost":              10,                          // float, optional
        "totalPrice":                92.4034,                     // float, mandatory
        "currency":                  "USD",                       // string, mandatory
        "exportCurrency":            "USD",                       // string, optional

        "totalWeight":       3.87,                        // float, mandatory
        "totalWeightScale":          "kg",                        // string, mandatory
        "dimensionalWeight":         1.95,                        // float, optional
        "dimensionalWeightScale":    "kg",                        // string, optional
        "dimensions": {                                           // object, optional
                "length":            10.5,                        // float, optional
                "width":             1.8,                         // float, optional
                "height":            21.3,                        // float, optional
                "dimensionalScale":  "cm",                        // string, mandatory
        },

        "countryOfOrigin":           "US",                        // string, mandatory
        "countryOfDestination":      "UK",                        // string, mandatory
```

[1] The quoteIdentifier is mandatory if your merchant account is enabled for quoting

```
"identificationData": {                                   // object, optional, [2]
       "identificationNumber":'abc123',                   // string, mandatory
       "identificationName":'近道',                  // string, mandatory
},
"shippingAddress": [{                              // object, mandatory
       "firstName":          "John",               // string, mandatory
       "lastName":           "Doe",                       // string, mandatory
       "streetName":         "Sample Street",             // string, mandatory
       "houseNo":            "6",                         // string, optional
       "additionalInfo":     "5th floor",                 // string, optional
       "postcode":           "SE28",               // string, mandatory
       "city":        "London",                    // string, mandatory
       "state":              "London",                    // string, optional
       "telephone":          "+44 20 7946 0551",          // string, optional
       "email":              "john@xample.com",           // string, mandatory
       "countryCode": "UK",                        // string, mandatory
},],
"billingAddress": [{                                      // object, mandatory
       "firstName":          "John",               // string, mandatory
       "lastName":           "Doe",                       // string, mandatory
       "streetName":         "Sample Street",             // string, mandatory
       "houseNo":            "6",                         // string, optional
       "additionalInfo":     "5th floor",                 // string, optional
       "postcode":           "SE28",               // string, mandatory
       "city":        "London",                    // string, mandatory
       "state":              "London",                    // string, optional
       "telephone":          "+44 20 7946 0551",          // string, optional
       "email":              "john@example.com",          // string, mandatory
       "countryCode": "UK",                        // string, mandatory
},],
"lineItems": [{
       "sku":                "T14K00357",          // string, mandatory
       "shortDescription":          "Awesome E-Reader",   // string, mandatory
       "price":                     10,                   // float, mandatory
       "category":                  "E-readers",          // string, mandatory
       "weight":                    0.57,                 // float, mandatory
       "weightScale":        "kg",                 // string, mandatory
       "quantity":                  3,                    // float, mandatory
       "countryOfManufacture":      "CN",                 // string, mandatory
       "hsCode":                    "9202903000",         // string, optional
       "exportHsCode":              "",                   // string, optional
       "exportPrice":               0,                    // float, optional
       "exportDescription":         "",                   // string, optional
},
// of course you can add more line items here
],
"submerchant": {                                          // object, optional, [3]
       "company":            "Example Inc.",              // string, mandatory
       "streetName":         "Sample Street",             // string, mandatory
       "houseNo":            "6",                         // string, mandatory
       "postcode":           "SE28",               // string, mandatory
```

[2] Identification data can be used for chinese import customer registration

[3] This object is only relevant for merchants that sell on marketplaces

```
                "city":         "London",                       // string, mandatory
                "state":                "London",                       // string, mandatory
                "email":                "john@example.com",             // string, mandatory
                "countryCode": "UK",                            // string, mandatory
        }
}
```

Response:

```
{
        "response": {
                "success":              true,
                "result":               "061039676",                    // shipment identifier
                "linkTracking":         "https://app.borderguru.com/tracking/1234567890",
        }
}
```

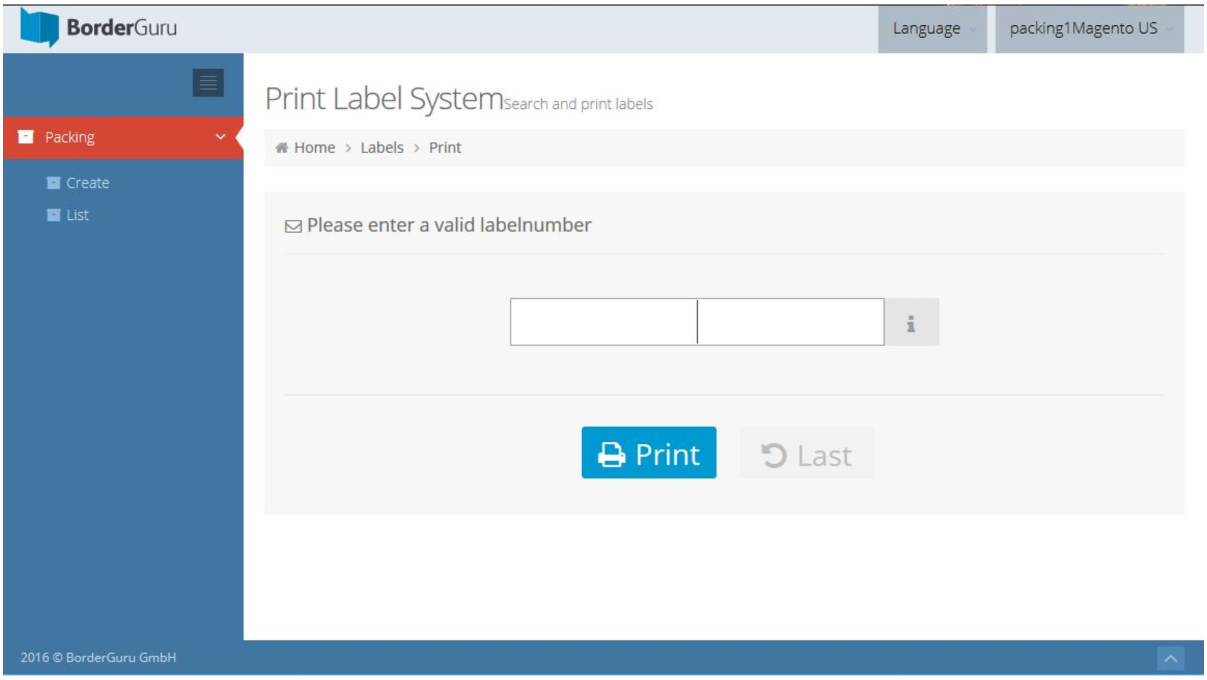
10.3. Getting the label

10.3.1. Label client

Using our label web client you can easily download PDF labels for each or your parcels. The address for your label web client is: https://app.borderguru.com/labels

Please login with the username and password that were provided by BorderGuru. You can then simply enter your internal order ID and download the label for that parcel as PDF.

Entering your internal order ID can of course also be done with a barcode scanner which scans a corresponding barcode of your warehouse management system or from somewhere else.



Print Label SystemSearch and print labels

Please enter a valid labelnumber

10.3.2. Labeling REST API

To receive the shipping labels you should use on your parcels you can use the labeling REST API:

Endpoint:

```
GET https://app.borderguru.com/api/orders/label/<shipping id>?apikey=<API access key>
```

Please substitute <API access key> with the API access key that we provided you.
Also please substitute <shipping id> with the valid shipping ID that you would like to create the label for.

Response:
The label as PDF that you can put on your parcel.

10.4. Export customs clearance

Export customs clearance is mandatory in some countries. BorderGuru will do this for you, but you need trigger this for each bulk of parcels. In countries where this is not required you can leave out this step. To trigger BorderGuru to generate an export declaration we provide a REST API for you.

The REST API is fairly simple, as it just gets a list of orders that should be declared as one bulk. However the REST API will respond with an error in case that our system does not have all data for the order. Necessary for the export declaration in Germany are the following fields that need to be provided before during transferring order data:

- exportHsCode
- exportDescription
- countryOfManufacture

In case the following fields are not provided before they will be set automatically to the following values:

- exportPrice => price
- exportCurrency = currency

Endpoint:

```
https://app.borderguru.com/api/export/<country>?apikey=<apikey>
```

<country> should be the 2-char country code of the country you want to export from. Currently BoderGuru only supports "de".

<apikey> is the API key that BorderGuru provided you.

Request:
You can choose between the following 3 options:

```
[{"merchantOrderId": "..."}/*, … */]
[{"shipmentIdentifier": "..."}/*, … */]
[{"labelId": "..."}/*, … */]
```

10.5. Track & Trace REST API

You can request track & trace information about a certain shipment by using the Track & Trace REST API. Main purpose is to read all tracking information and use it in a customized way to display it your customers.

Endpoint:

```
GET https://app.borderguru.com/api/orders/tracking/<id>
```

Please replace <id> by the order ID or shipping ID.

Response:

```
{
        "tracking": {
                "events": [{
                        "timeStamp": "2016-03-14T21:36:10.000Z",    // tracking timestamp
                        "type": "TRACKING",
                        "content": {
                                "trackingStatus": "CREATED", // tracking status
                                "message": "Order has been created",// tracking message
                                "location": null                    // tracking location
                        },
                },
                // more tracking events can appear here
                ]
        }
}
```