

JSON customer import documentation

Contents

1	List of changes	2
2	Preface	3
2.1	Create or update	3
3	Authentication	4
4	Customer import.....	5
5	Contact person import	10
6	Response	12
6.1	Result	12
6.2	Error	13
7	URL for the request.....	14
8	Support.....	14

1 List of changes

Rev.	Description	Date
1.0	<ul style="list-style-type: none"> Initial version. 	01-2014
1.1	<ul style="list-style-type: none"> Modified error numbers. 	02-2014
1.2	<ul style="list-style-type: none"> Error code 15 added (module external systems is now mandatory to use the API) 	11-2016
1.3	<ul style="list-style-type: none"> New parameter “\customers\customerno” added. This can be used to create customers with a custom customer number or to update existing customers. New parameter “\customers\update_on_existing_customerno” added to update new customers. Error code 53 and 54 added. 	10-2017
1.4	<ul style="list-style-type: none"> “\customers\language” added. This option controls the default language of created documents and emails. Error code 55 added. 	01-2018
1.5	<ul style="list-style-type: none"> “\customers\delete_existing_customer_contacts” added to delete existing contacts in order to re-create them. (Only a customer can be updated. Not a contact within a customer). 	03-2018
1.6	<ul style="list-style-type: none"> Error code 16 added. 	10-2018
1.7	<ul style="list-style-type: none"> Contacts within a customer can now also be updated. When a contact is created, the API will respond with the contacts ‘userid’. This id can be used in subsequent API calls to update the contact. “\customers\customer_contacts\userid” added. Error 65 added. “\customers\customer_contacts\use_email_for_invoice” added (default is true). 	02-2020
1.8	<ul style="list-style-type: none"> “\customers\payment_method” added. “\customers\crm_notes” added. 	05-2021
2.0	<ul style="list-style-type: none"> The API now accepts JSON via a raw POST. The old method via a form POST can still be used but should not be used for new implementations. Updated example files using Postman and the Guzzle PHP HTTP client. 	11-2021

2 Preface

This document describes the fields that can be used to import customers and their corresponding contact persons into EasyTrans. The API uses JSON, which is a data-exchange format, just as XML. JSON is easier to read for human and lightweight for machines. There are JSON parsers available for nearly every programming language and platform (more info, see www.json.org).

The import of customers in EasyTrans globally consists of the following scheme:

- { **Authentication** }
- [**Customers**]
 - { Customer 1 }
 - [*Contacts*]
 - { Contact 1 }
 - { Contact 2 }
 - { Contact n }
 - { Customer 2 }
 - [*Contacts*]
 - { Contact 1 }
 - { Contact 2 }
 - { Contact n }
 - { Customer n }
 - [*Contacts*]
 - { Contact 1 }
 - { Contact 2 }
 - { Contact n }

In the tables below, all possible fields are described. Also, the data type, required field yes/no and default value when omitted, is supplied. Below the tables, a JSON example is given.

Please note: All JSON data must be UTF-8 encoded.

Attached to this document are two JSON example files:

- EasyTrans - JSON customer import example - simple.json
A simple example with 1 customer with 1 contact.
- EasyTrans - JSON customer import example - extended.json
An extended example with 2 customers with 3 contacts in total.

2.1 Create or update

The customer import API can be used for creating new customers as well as updating existing customers. To edit an existing customer the parameters "customerno" and "update_on_existing_customerno" must be supplied.

Contact persons can be updated if you have previously stored the userid upon creation of the contact. If the userid is not known, the contact can alternatively be deleted and recreated.

3 Authentication

In order to submit customers in EasyTrans, an authentication needs to be performed. This is handled by the “authentication” object. Authentication is done by a username and password (which can be created with an administrative account in EasyTrans). Together with the credentials, the type of import is provided (currently customers or orders). You can also choose to send a test message to test your implementation, before submitting real customers.

Note: Authentication is not handled via the HTTP header, as is common with a REST API.

Name	authentication
Parents	\
Required?	Yes
Type	Object
Description	Container for authentication.

Name	username
Parents	\authentication\
Required?	Yes
Type	String
Description	Username. Must be a member of one of the following groups: <ul style="list-style-type: none"> - Import/Export API - Administrator Note: Only user accounts that belong to a branch can be used to import customers. A customer account cannot be used to create new customers.

Name	password
Parents	\authentication\
Required?	Yes
Type	String
Description	Password.

Name	type
Parents	\authentication\
Required?	Yes
Type	Enum [order_import, customer_import]
Description	The type of data to be imported: <ul style="list-style-type: none"> - order_import: Import of orders with destinations and goods. (See order import API) - customer_import: Import of customers with contacts.

Name	mode
Parents	\authentication\
Required?	No, default: “test”
Type	Enum [effect, test]
Description	Option to test your data: <ul style="list-style-type: none"> - effect: Data is read, checked and processed. The response contains the amount checked and processed customers, status and error messages. - test: Data is read and checked, but not processed. The response contains the number of checked customers and error messages.

```
{
  "authentication": {
    "username": "user1234",
    "password": "Abcd1234!",
    "type": "customer_import",
    "mode": "test"
  }
}
```

```
},
```

4 Customer import

Name	customers
Parents	\
Required?	Yes
Type	Array with customer objects
Description	Container for the import of orders.

Name	customerno
Parents	\customers\
Required?	No, default: First available free customer number
Type	Integer
Description	Customer number. If this field is omitted, EasyTrans will automatically assign the next free customer number. If this field is filled in and the customer number already exists, an error occurs, unless "\customers\update_on_existing_customerno" is set to true.

Name	update_on_existing_customerno
Parents	\customers\
Required?	No, default: false
Type	Boolean
Description	When this parameter is set to true, the data of the customer will be updated if the customer number already exists. By default, this parameter is set to false and an error will occur when the customer number already exists. This prevents unwanted overwriting of existing customers. Note: This parameter only has a function if "\customers\customerno" is filled in.

Name	delete_existing_customer_contacts
Parents	\customers\
Required?	No, default: false
Type	Boolean
Description	When this parameter is set to true, all existing contacts for this customer will be deleted. This can be used to update the customers contacts by recreating them (if the userid is unknown).

Name	company_name
Parents	\customers\
Required?	Yes
Type	String, 1 to 100 characters
Description	Company name. If the customer is a private person, the person's name is filled in.

Name	attn
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 100 characters
Description	For the attention of. For example, a department within a company.

Name	address
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 100 characters.

Description	Address.
--------------------	----------

Name	houmeno
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 60 characters
Description	House number. The house number and addition can be separated in this field and the next field or combined in this field only.

Name	addition
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 60 characters
Description	House number addition. Can also be combined in the house number field.

Name	address2
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 100 characters
Description	Second address line.

Name	postal_code
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 12 characters
Description	Postal code or ZIP code.

Name	city
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 100 characters
Description	City.

Name	country
Parents	\customers\
Required?	No, default: Country of the carrier
Type	String, 0 to 50 characters
Description	Country name or country code. Either the full country name or the 2 or 3 letter country code can be used. The country must be active.

Name	mail_address
Parents	\customers\
Required?	No, default: Same as "\customers\address\"
Type	String, 0 to 100 characters
Description	Post address.

Name	mail_houmeno
Parents	\customers\
Required?	No, default: Same as "\customers\houmeno\"
Type	String, 0 to 60 characters
Description	Post house number. The post house number and addition can be separated in this field and the next field or combined in this field only.

Name	mail_addition
Parents	\customers\
Required?	No, default: Same as "\customers\addition\"
Type	String, 0 to 60 characters
Description	Post house number addition. Can also be combined in the post house number field.

Name	mail_address2
Parents	\customers\
Required?	No, default: Same as "\customers\address2\"
Type	String, 0 to 100 characters
Description	Post second address line.

Name	mail_postal_code
Parents	\customers\
Required?	No, default: Same as "\customers\postal_code\"
Type	String, 0 to 12 characters
Description	Post Postal code or ZIP code.

Name	mail_city
Parents	\customers\
Required?	No, default: Same as "\customers\city\"
Type	String, 0 to 100 characters
Description	Post City.

Name	mail_country
Parents	\customers\
Required?	No, default: Same as "\customers\country\"
Type	String, 0 to 50 characters.
Description	Post country name or country code. Either the full country name or the 2 or 3 letter country code can be used. The country must be active.

Name	debtorno
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 50 characters
Description	Debtor number. If this field is used, this number will be shown on the invoice, instead of the customer number. Only used if the customer number in EasyTrans and the external system are different.

Name	payment_ref
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 100 characters
Description	Payment reference. Fixed reference that is printed at the top of the invoice.

Name	website
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 255 characters
Description	Website address without "http(s)://".

Name	remark
Parents	\customers\
Required?	No, default: Empty
Type	String
Description	Comments for this customer. These comments can be taken over to the 'Comments on the delivery note' field in a new order, based on the system settings.

Name	crm_notes
Parents	\customers\
Required?	No, default: Empty
Type	String
Description	Comments related to Customer Relationship Management. These comments can be taken over to the 'Comments internal' field in a new order, based on the system settings.

Name	ibanno
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 50 characters
Description	IBAN bank account number.

Name	bicno
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 20 characters
Description	BIC code of the bank account.

Name	cocno
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 50 characters
Description	Chamber of commerce number.

Name	vatno
Parents	\customers\
Required?	No, default: Empty
Type	String, 0 to 50 characters
Description	VAT number.

Name	payment_method
Parents	\customers\
Required?	No, default: Empty
Type	Enum ["", bank_transfer, cash, direct_debit, online_payment, bank_transfer_online_payment, creditcard, factoring]
Description	<ul style="list-style-type: none"> - Empty: Defaults to payment by bank transfer. - bank_transfer: Payment by manual bank transfer. - cash: Payment by cash. - direct_debit: Payment by a direct debit. - online_payment: Payment using an online payment provider. Currently we support Mollie which offers many different payment methods, such as iDEAL, Creditcard, PayPal and many more. <p>This method adds a payment link to the invoice or proforma invoice, to directly pay the invoice. The actual payment status is visible in EasyTrans.</p> <ul style="list-style-type: none"> - bank_transfer_online_payment: Combination of payment by bank transfer or by online payment, so your customer can either pay using the payment link or transfer the money manually.

- creditcard: Payment by Credit/Debit card.
- factoring: Payment by bank transfer to a third party that manages the payments.

Name	vat_liable
Parents	\customers\
Required?	No, default: 1
Type	Enum [0, 1, 2]
Description	1: Customer is liable to pay tax. 0: The customer is not liable to pay tax. VAT shifted. For customers within the EU. 2: The customer is not liable to pay tax. No VAT, export. For customers outside the EU.

Name	language
Parents	\customers\
Required?	No, default: Language of the carrier
Type	Enum ['nl', 'en', 'de', 'fr', ""]
Description	Preferred language for generated documents and emails.

```
"customers": [
{
  "customerno": 12345,
  "update_on_existing_customerno": true,
  "delete_existing_customer_contacts": true,
  "company_name": "Example Company A",
  "attn": "Administration",
  "address": "Keizersgracht",
  "houzeno": "1a",
  "address2": "2nd floor",
  "postal_code": "1015CC",
  "city": "Amsterdam",
  "country": "NL",
  "mail_address": "Postbus",
  "mail_houzeno": "73",
  "mail_postal_code": "1010AA",
  "mail_city": "Amsterdam",
  "mail_country": "NL",
  "payment_ref": "Example project",
  "website": "www.easytrans.nl",
  "remark": "Customer remark",
  "bicno": "INGBNL2A",
  "ibanno": "NL63INGB0004511811",
  "cocno": "50725769",
  "vatno": "NL822891682B01",
  "payment_method": "bank_transfer",
  "vat_liable": 1,
}
```

5 Contact person import

Name	customer_contacts
Parents	\customers\
Required?	No, default: Empty
Type	Array with contact person objects
Description	Container for the import of contact persons with a customer.

Name	userid
Parents	\customers\customer_contacts\
Required?	No, default: 0
Type	Integer
Description	When a contact is created, the API will return it's userid. The userid can be used to update the contact in a later API call. Note: If the userid is not known, it is not possible to update contacts. In this case all old contacts can be deleted and recreated by using 'delete_existing_customer_contacts'.

Name	salutation
Parents	\customers\customer_contacts\
Required?	No, default: 0
Type	Enum [0, 1, 2, 3]
Description	Salutation 0 = Mr./Mrs. 1 = Mr. 2 = Mrs. 3 = Attn.

Name	contact_name
Parents	\customers\customer_contacts\
Required?	No, default: Empty
Type	String, 0 to 100 characters
Description	Name of the contact. Can be the name of a person or a department, such as front desk, warehouse or administration

Name	telephone
Parents	\customers\customer_contacts\
Required?	No, default: Empty
Type	String, 0 to 40 characters
Description	Phone number.

Name	mobile
Parents	\customers\customer_contacts\
Required?	No, default: Empty
Type	String, 0 to 40 characters
Description	Mobile phone number.

Name	email
Parents	\customers\customer_contacts\
Required?	No - Default: Empty
Type	String, 0 to 255 characters
Description	Email address. More than one address per contact can be provided by using a ; (semicolon).

Name	use_email_for_invoice
Parents	\customers\customer_contacts\
Required?	No - Default: True
Type	Boolean
Description	If this parameter is set, the email address for this contact is used to send the digital invoice to.

Name	contact_remark
Parents	\customers\customer_contacts\
Required?	No - Default: Empty
Type	String
Description	Remarks regarding this contact.

Name	username
Parents	\customers\customer_contacts\
Required?	No - Default: Empty
Type	String, 0 to 255 characters
Description	Login name. If you want to give the customer access to the customer portal, you can provide a username and password. When a username and password are provided, the user will automatically be added to the 'Access web-ordering' user rights level. Note: The username must be unique. The response message must be checked for this.

Name	password
Parents	\customers\customer_contacts\
Required?	Conditional, when "\customers\customer_contacts\username" is filled in - Default: Empty
Type	String with a minimum 8 characters. Minimum 1 lowercase letter, 1 uppercase letter and one number.
Description	Password for this user account.

```

"customer_contacts": [
  {
    "userid": 1234567,
    "salutation": 1,
    "contact_name": "Bram Pietersen",
    "telephone": "020-7654321",
    "mobile": "06-12345678",
    "fax": "020-1234567",
    "email": "support@easytrans.nl",
    "use_email_for_invoice": true,
    "contact_remark": "Warehouse manager",
    "username": "bram",
    "password": "v5xhCmRs"
  }
]
}
]
}

```

6 Response

The response contains feedback about the result of the import. The response is also in JSON format. A response can either be successful (result) or unsuccessful (error). If there are no errors, the customers will be saved (unless test mode is used). If an error occurs, all processing stops, and an error is generated. No data is saved when an error occurs, so a batch import can be submitted in total again after the error has been corrected.

6.1 Result

Name	result
Parents	\
Type	Object
Description	Container with the result object.

Name	mode
Parents	\result\
Type	Enum [effect, test]
Description	Feedback of the import mode. (See “\authentication\mode” in the chapter Fout! V erwijzingsbron niet gevonden.).

Name	total_customers
Parents	\result\
Type	Integer
Description	The number of processed customers.

Name	total_customer_contacts
Parents	\result\
Type	Integer
Description	The total amount of processed contacts (of all customers together).

Name	result_description
Parents	\result\
Type	String
Description	Description of the result. Options are: <ul style="list-style-type: none"> - Import completed successfully. ... customers have been added. - Test import completed successfully. No changes have been made.

Name	new_customernos
Parents	\result\
Type	Array van integers
Description	List of all newly added customer numbers. (Empty array when mode=test).

```
{
  "result": {
    "mode": "effect",
    "total_customers": 1,
    "total_customer_contacts": 1,
    "result_description": "Import completed successfully. 1 customers have
been added or updated",
    "new_customernos": [123485],
    "new_userids": {
      "123485": [201]
    }
  }
}
```

6.2 Error

Name	error
Parents	\
Type	Object
Description	Container for errors.

Name	errorno
Parents	\error\
Type	Integer
Description	Number of the error (see error_description for a list of errors with their numbers).

Name	error_description
Parents	\error\
Type	String
Description	<p>A Description of the error. The following errors can occur:</p> <p><u>JSON</u></p> <p>5. JSON error: No valid JSON received. Did you use a raw post?</p> <p><u>authentication</u></p> <p>10. No username given (required field). 11. No password given (required field). 12. Login attempt failed. Invalid username or password. 13. Unknown mode: Possible values are: effect, test. 14. Unknown type: Possible values are: order_import, customer_import, packs_order_import, gls_order_import. 15. The API is not active. Please contact the carrier or EasyTrans Software. 16. This user account cannot be used to import customers. 18. Login attempt failed. Account disabled. 19. Login attempt failed. Invalid user rights.</p> <p><u>customers</u></p> <p>50. No company_name given (required field). 51. Unknown or disabled country: (Check the country list for valid countries). 52. Unknown or disabled mail_country: (Check the country list for valid countries). 53. The customerno ... must be a whole numeric value. 54. The customerno ... already exists. Use update_on_existing_customerno if you want to perform an update. 55. The language must be 'nl', 'en', 'de', 'fr' or empty for the default language. 56. The payment method must be bank_transfer, cash, direct_debit, online_payment, bank_transfer_online_payment, creditcard, factoring or empty string.</p> <p><u>customer contacts</u></p> <p>60. The username ... is already in use. Choose a different username. 61. Username too long (maximum 255 characters). 62. No password given (required when username is not empty). 63. Password too short (minimum 8 characters). 65. Unknown userid: ...</p>

```
{
  "error": {
    "errorno": 12,
    "error_description": "Login attempt failed. Invalid username or password"
  }
}
```

7 URL for the request

The JSON data is sent to the endpoint of the carrier by means of a raw HTTP POST request.

The URL for the request is as follows:

https://www.SERVER/ENVIRONMENT/import_json.php

In this URL:

- **SERVER** the server where the EasyTrans environment is hosted (for instance mytrans.nl, mytrans.be or mytransport.co.uk)
- **ENVIRONMENT** the EasyTrans environment of the carrier

Note: The use of HTTPS is required.

As a part of this documentation package, two examples are attached to perform a request:

- **Postman example**
[EasyTrans - JSON customer import - Postman example.postman_collection.json](#)
Import this example file in Postman to try out requests.
- **PHP example**
[EasyTrans - JSON customer import - Guzzle example.php](#)
Example of a simple request in PHP using the Guzzle HTTP client.

Depending on the platform, several other options are available. Libcurl can for instance be used for C++, Java, Perl, Python, etc.

8 Support

Free support for the implementation and use of the JSON order import API is included in the module "Connection with external systems", which you will need to purchase in order to use the API. Questions can be asked via email on support@easytrans.nl and will generally be answered within 1 business day.