

6 Customers API: How to create a customer handle

This is a series of articles dedicated to managing customer objects via Openprovider API.

This is a first article in a series in which we are going to cover customer handle creation. To get more information about our API in general, please use our [documentation portal](#) as your reference.

Customer handles are used throughout the whole system in all product modules (domains, SSL, licenses etc). Every customer created in our system gets a unique identifier or a customer "handle". These handles allow you to reuse recurring customers, meaning you will not have to fill in all the information every time you wish to [register](#) a domain name. Another benefit is that you can simply update the customer handle if they ever change their details, this will result in your domain names being updated as well.

Creating a handle

First, let's focus on adding that handle. While some domain registries require specific data to be provided within the handle object (more on this could be found in [Customer Extension Additional Data](#) section of the Openprovider docs) it is important that we have at least the bare minimum of information to be able to register domain names and order SSL certificates. All of the required fields are marked accordingly in the following table.

↓ Use endpoint path below. Base URL is determined by both the target environment and API version.

POST [{base_url}](#)/customers

REQUEST VALUES

Name	Type	Description
company_name	string	Optional. If customer is a company
vat	string	Optional.

name	array with keys: (initials, first_name, prefix, last_name, full_name)	Required. Keys 'first_name' and 'last_name' are mandatory.
address	array with keys : (street, number, suffix, zipcode, city, country, state)	Required. Keys 'street', 'number', 'zipcode', 'city' and 'country' are mandatory. Some registries also require a valid value for 'state'.
phone	array with keys: (country_code, area_code, subscriber_number)	Required.
fax	array with keys: (country_code, area_code, subscriber_number)	Optional.
email	various	Required.
additional_data	various	Optional.
extension_additional_data	various	Required for some TLDs
locale	predefined	Optional.
tags	array with keys (key, value)	Optional.

REQUEST EXAMPLE

```
curl -X POST \
  http://api.openprovider.eu/v1beta/customers \
  -H 'Authorization: ,Bearer 2831a37fb*****d790b5aac822' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "additional_data": {
      "birth_address": "test street 45",
      "birth_city": "Test city",
      "birth_country": "NL",
      "birth_date": "1970-01-01",
      "birth_state": "Testprovince",
      "birth_zipcode": "123",
      "company_registration_city": "Testcity",
      "company_registration_number": "XX123456789X04",
      "company_registration_subscription_date": "2000-01-01",
      "headquarters_address": "Test street 5",
      "headquarters_city": "Testcity",
      "headquarters_country": "NL",
      "headquarters_state": "Testprovince",
      "headquarters_zipcode": "123",
      "passport_number": "123456",
      "social_security_number": "6543210"
    },
    "address": {
```

```
"city": "Test city",
"country": "NL",
"number": "1",
"state": "Testprovince",
"street": "Test street",
"suffix": "A",
"zipcode": "1236 XX"
},
"comments": "",
"company_name": "Example company",
"email": "test@mail.tld",
"extension_additional_data": [
{
"data": {
"admin_type": "1",
"applicant_purpose": "P3",
"birth_date": "1970-01-01",
"company_name_cyrillic": "Example company",
"company_name_latin": "Example company",
"company_registration_number": "XX123456789X11",
"contact_title": "CEO",
"ens_auth_id": "",
"ens_key": "",
"first_name_cyrillic": "Test",
"first_name_latin": "Test",
"hr_member": "",
"industry_class": "19",
"is_private_entrepreneur": "",
"last_name_cyrillic": "Person",
"last_name_latin": "Person",
"legal_address_cyrillic": "13245 Country, City, street, 2",
"middle_name_cyrillic": "Nikolaevich",
"middle_name_latin": "Nikolaevich",
"mobile_phone_number": "+7 (123) 456-78-90",
"nexus_category": "C31",
"org_type": "1",
"passport_issue_date": "2009-01-11",
"passport_issuer": "Issuer name",
"passport_number": "12345678",
"passport_series": "1234",
"postal_address_cyrillic": "12345 Country, City, street, 1",
"social_security_number": "X123455123X",
"tax_payer_number": "1345678",
"uin": "12345-6",
"user_type": "Individual",
"vat": "XX123456789X12",
"website": "http://example.com",
"whois_email": "test@mail.tld"
},
"name": "ru"
}
],
```

```
"fax": {
  "area_code": "111",
  "country_code": "+05",
  "subscriber_number": "123456"
},
"locale": "nl_NL",
"name": {
  "first_name": "Test",
  "full_name": "Test Person",
  "initials": "T P",
  "last_name": "Person",
  "prefix": "van der"
},
"phone": {
  "area_code": "111",
  "country_code": "+04",
  "subscriber_number": "123456"
},
"tags": [
  {
    "key": "customer",
    "value": "VIP"
  }
],
"vat": "NL123456789X11"
}'
```

RESPONSE EXAMPLE

```
{
  "code": 0,
  "data": {
    "handle": "XX123456-XX"
  },
  "desc": ""
}
```

RESPONSE VALUES

Name	Type	Description
handle	varchar	Customer unique identifier or handle

In the [following article](#) we'll get into retrieving information from an existing handle.

