Check VAT ID with regular expressions and VIES

31 Jul 2022

In the European Union, each business registered for VAT has an unique identification number like IE9825613N or LU20260743. When selling to EU companies, you need to ask for their VAT ID, validate it, and include it into the invoice. The tax rate depends on the place of taxation and the client type (a person or a company). Some customers may provide a wrong VAT ID — either by mistake or in an attempt to avoid paying the tax. So it's important to check the VAT number.

EU provides the VIES page (VAT Information Exchange System) and a free SOAP API for the VAT ID validation. Here is how you can query the API:

```
Linux / macOS:

curl -d '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:ec.europa.eu:taxud:vies:services:checkVat:types"><soapenv:Body><urn:checkVat><urn:countryCode>IE</urn:countryCode><urn:vatNumber>9825613N</urn:vatNumber></urn:checkVat></soapenv:Body></soapenv:Envelope>' 'https://ec.europa.eu/taxation_customs/vies/services/checkVatService'

Windows:

(iwr 'https://ec.europa.eu/taxation_customs/vies/services/checkVatService' -method post -body '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:ec.europa.eu:taxud:vies:services:checkVat:types"><soapenv:Body><urn:checkVat><urn:countryCode>IE</urn:countryCode><urn:vatNumber>9825613N</urn:vatNumber></urn:checkVat></soapenv:Body></soapenv:Envelope>').content
```

If the VAT number is invalid, you will get `<valid>false</valid>` in the response. You can use a SOAP library or just concatenate the XML string with the VAT identification number. In the latter case, you should quickly check the VAT number with a regular expression, otherwise an attacker can include an arbitrary XML code into it. The VIES WSDL file provides these regular expressions:

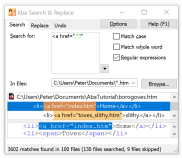
```
Country code: [A-Z]{2}
VAT ID without the country code: [0-9A-Za-z\+\*\.]{2,12}
```

The country code consists of two capital letters; the VAT ID itself is from 2 to 12 letters, digits, or these characters: `+` `*` `.`

So the finished code for VAT ID validation could look like this:

```
import re, urllib.request, xml.etree.ElementTree as XmlElementTree

# Return a dictionary with some information about the company, or False if the vat_id is invalid
def check_vat_id(vat_id):
  m = re.match('^([A-Z]{2})([0-9A-Za-z\+\*\.]{2,12})$', vat_id.replace(' ', ''))
  if not m:
    return False

  data = '<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" ' + \
         'xmlns:urn="urn:ec.europa.eu:taxud:vies:services:checkVat:types"> + \
         '<soapenv:Body><urn:checkVat><urn:countryCode>' + m.group(1) + '</urn:countryCode> + \
         '<urn:vatNumber>' + m.group(2) + '</urn:vatNumber></urn:checkVat></soapenv:Body></soapenv:Envelope>'

  with urllib.request.urlopen('https://ec.europa.eu/taxation_customs/vies/services/checkVatService', data.encode('ascii')) as response:
    resp = response.read().decode('utf-8')

    ns = {
      'soap': 'http://schemas.xmlsoap.org/soap/envelope/',
      'checkVat': 'urn:ec.europa.eu:taxud:vies:services:checkVat:types',
    }

    checkVatResponse = XmlElementTree.fromstring(resp).find('./soap:Body/checkVat:checkVatResponse', ns)
    if checkVatResponse.find('./checkVat:valid', ns).text != 'true':
      return False

    res = {}
    for child in checkVatResponse:
      res[child.tag.replace('{urn:ec.europa.eu:taxud:vies:services:checkVat:types}', '')] = child.text
    return res


print(check_vat_id('IE9825613N'))
```

Each EU country also has its own rules for a VAT identification number, so you can do a stricter pre-check with a complex regular expression, but VIES already covers this for you. Also note that some payment processors (e.g. Stripe) already do a VIES query under the hood.



Replacing text in several files used to be a tedious and error-prone task. Aba Search and Replace solves the problem, allowing you to **correct errors** on your web pages, **replace banners** and copyright notices, change method names, and perform other text-processing tasks.

Download now

This is a blog about Aba Search and Replace, a tool for replacing text in multiple files.

Subscribe

- Privacy Policy Update - December 2022
- Aba 2.5 released
- Our response to the war in Ukraine
- **Check VAT ID with regular expressions and VIES**
- Which special characters must be escaped in regular expressions?
- Aba 2.4 released
- Privacy Policy Update - April 2021
- Review of Aba Search and Replace with video
- Aba 2.2 released
- Discount on Aba Search and Replace
- Using search and replace to rename a method

- Cleaning the output of a converter
- Aba 2.1 released
- How to replace HTML tags using regular expressions
- Video trailer for Aba
- Aba 2.0 released

© Peter Kankowski, 2007—2023 (contact, privacy policy)