VAT Documentation

Release 0.3.2

Alastair Houghton

Oct 18, 2018

Contents

CHAPTER 1

What is this?

It’s a package of useful Python code relating to VAT, that’s what. It includes

- Tables mapping ISO alpha 2 country codes to VAT codes and vice-versa
- Tables of VAT number formats
- Tables of VAT rates, thresholds and registration periods
- Support for querying the VIES VAT database system (to check VAT details), *including fuzzy matching support for names and addresses*
- Utility functions for UK VAT (in the vat.gb package), including
  - EC sales list XML generation
  - Mini One Stop Shop (MOSS) return generation

CHAPTER 2

A few words about VAT

In the European Union, companies are expected to charge and account for Value Added Tax (VAT). This has a variety of other names (IVA, TVA, MwSt., USt., BTW, , ΦΠΑ, DPH, PDV, Moms, km, ALV, ÁFA, CBL, PVN, PVM, PTU, DDV, IGIC and TAW), but it's the same basic idea, and the rules are—to some extent—harmonised.

From the consumer's perspective, VAT is a sales tax. However, from a business' perspective, it is not; unlike a sales tax, businesses charge VAT to *everybody*, whether or not they are a business customer, and are able to offset the VAT they have to pay over to the tax authority against the VAT they have themselves paid.

Sounds simple? Hah! You have *no* idea.

The first problem is that all of the different countries have their own rates for VAT, and most of them have at least two rates that apply to different categories of goods and services. The second problem is cross-border transactions; depending on the nature of the transaction, the supplier may or may not be expected to charge VAT to the purchaser, and when they do not, the purchaser will generally be expected to account for VAT under the "reverse charge" rules (essentially by charging himself VAT, which will normally result in no payment to the tax authority because the amount to be paid will be offset entirely by itself).

There are all kinds of additional complications—some member states operate registration thresholds, others don't; some provide a grace period for registration, in others you've broken the law if you make a supply that takes you over the threshold; some allow you to register and submit returns yourself, others require that you hire people to do it for you; and so on.

One further niggle is that **suppliers of electronic services outside of the European Union are expected to charge VAT to their European customers**. Many don't, with the result that their products can be priced more competitively than would be possible for a company operating within the European Union.

CHAPTER 3

DISCLAIMER

It is YOUR RESPONSIBILITY if you use this code to check that the rates and thresholds you are using are the correct ones. The author cannot be held liable for errors in this material, particularly as much of it comes from documents on the EU website that may be out of date and that themselves include disclaimers.

CHAPTER 4

Documentation

The documentation for this package is available on Read the Docs.

Contents:

4.1 A quick VAT Primer

This is a very quick overview of the terminology and workings of VAT in the EU. It isn't intended to be exhaustive, but it should cover the basics and help you to understand the more detailed material supplied by EU member states' VAT administrations.

This guide was written on the 1st of January 2015 and should be accurate for that date. Nevertheless, the author accepts no responsibility for any errors container herein; if in doubt, check with a qualified accountant or your own tax authority.

4.1.1 What is VAT?

Value Added Tax, or VAT, looks from the perspective of a consumer to be the same as a Sales Tax — that is, a tax based on the price of the item they have purchased.

The main difference between the two is that a true Sales Tax is only applied at the point of sale to a consumer. VAT, on the other hand, is applied for every intermediate transaction involved. For instance, imagine company A makes a widget, which it sells to a wholesaler, B, who then sells it on to a retailer, C, from where it is finally sold to a consumer, D. Let's also say tha the rate of Sales Tax or VAT we're considering is 10%. Here's what happens in both cases:

Sales Tax

1. Company A sells widget for £10 to Wholesaler B.
2. Wholesaler B sells widget for £15 to Retailer C.
3. Retailer C sells widget to Consumer D for £20, plus £2 sales tax.

4. Retailer C now pays the £2 to the government (typically — sometimes Consumer D might need to pay it instead).

VAT

1. Company A sells widget for £10 plus £1 VAT to Wholesaler B. On its tax return, Company A lists the £1 as "output" tax, and Wholesaler B lists the £1 as "input" tax.
2. Wholesaler B sells widget for £15 plus £1.50 VAT to Retailer C. On its tax return, Wholesaler B lists the £1.50 as "output" tax, and Retailer C lists the £1.50 as "input" tax.
3. Retailer C sells widget to Consumer D for £20, plus £2 VAT. On its tax return, Retailer C lists the £2 as "output" tax.
4. Each of the companies now pays the difference between their output and input tax to the government, so Company A pays £1, Wholesaler B pays £0.50, and Retailer C pays a further £0.50.

   The total received by the government is still £2, but it has been collected piecemeal from all of the suppliers in the chain.

4.1.2 Implications of VAT

The first implication of the above is that all businesses are going to have to file tax returns for VAT purposes, because they need to account for the "input" tax (tax that has already been paid) and "output" tax (tax they have collected on a sale).

The second implication is that government is going to have to regulate invoices and credit notes, because those are going to be the primary source of information on how much VAT is payable, and they must be usable as proof of tax paid by businesses claiming input tax as well as showing clearly the amount of output tax that has been collected at each step.

Additionally, supply chains are rarely as simple as the above; typically, Company A would have multiple inputs to its widget, and would want to claim the input tax on each of those.

One further complication is that, as with Sales Tax, it is desirable that some things be exempt from VAT. In a pure Sales Tax, that's easy — you just don't charge VAT to the consumer when they purchase an item. With VAT, you wouldn't want the government to end up out of pocket because of input tax claims against something that was exempt, so typically businesses are prohibited from claiming input tax for any item that was used to produce an exempt item for sale. If an item they purchase is partly used for the production of VAT-exempt items, and partly for items subject to VAT, they may have to work out what proportion applies and can only reclaim that proportion of the input tax.

If that sounds complicated, that's because it is.

So why would anyone choose a VAT over a Sales Tax? The idea is that it is harder to avoid paying VAT. In a Sales Tax system, Consumer D could pretend to be a business and purchase from Retailer C without paying the Sales Tax. But in the VAT system, *VAT is always charged*, so Consumer D can't avoid paying it.

4.1.3 Self-billing and the Reverse Charge

In the EU VAT system, there are actually two instances where VAT is not charged. The first is if you operate a "self-billing" arrangement with a customer or supplier; let's imagine that Company A and Wholesaler B have an agreement to operate self-billing. Company A now does not need to worry about VAT; it sells its widget directly to Wholesaler B for £10, marking the invoice clearly to indicate that the self-billing arrangement applies.

Wholesaler B then accounts for VAT as if it has sold the item to itself. So, it lists £1 as "output" tax and also £1 as "input" tax. When it sells the widget to Retailer C, with whom it does not operate self-billing, it lists the £1.50 of "output" tax as before, so Wholesaler B will now pay £1.50 to the government. Notice that the final amount paid to the government is still the same £2.

The second place is actually a variant of self-billing, and applies where a business purchases a *service* (more on this in a moment) from an overseas supplier. In that specific case, the EU "reverse charge" mechanism applies; the supplier need not worry about VAT, though if they are also in the EU they may need to indicate on the invoice that the reverse charge applies. The purchaser then accounts for VAT as if it had sold the service to itself. The upshot of this rule is that services supplied to business customers are taxed at the rate in the customer's country, not in the supplier's country as might otherwise be the case.

4.1.4 Goods and Services

We've skirted around this a bit so far, but EU VAT law distinguishes between *goods* (broadly speaking, physical items) and *services* (pretty much everything else, *including digital "goods" like software, music and films*).

The rules for goods, services and e-services are very different and so the first thing to consider when you are going to sell something in the EU is whether what you are selling is a good, a service, or an e-service.

(An *e-service*, in case you are wondering, is a service that is provided electronically with minimal human intervention. Downloadable software is one such example. The rules on what is or is not an e-service can be quite tricky so if you need clarification you may need to consult your tax authority.)

4.1.5 When must I charge VAT and at what rate?

Case 1: You are in the same EU member state as your customer

In this case, the rules that apply are those of that member state.

Case 2: You are in a different EU member state to that of your customer

- If you are selling a *good*, then provided your sales have not exceeded the distance selling threshold for your customer's member state, you can charge VAT according to *your* country's rules. Otherwise, you must register in the customer's member state and apply that member state's rules to the sale.
- If you are selling a *service* or an *e-service* and the customer is a business, you charge no VAT but indicate to the customer that they should apply the reverse charge mechanism.
- If you are selling a *service* to a consumer, you charge VAT according to *your* country's rules.
- If you are selling an *e-service* to a consumer, you can either register in the customer's member state and apply the relevant rules, *or* you can register for the Mini One Stop Shop in your member state. In both cases you will charge VAT at the rate applicable in the customer's member state.

Case 3: You are outside the EU and your customer is in an EU member state

- If you are selling a *good*, VAT is your customer's responsibility and should be dealt with, along with import duty, by the shipping company.
- If you are selling a *service*, VAT is your customer's responsibility and they are supposed to operate the reverse charge.
- If you are selling an *e-service* to a business, VAT is your customer's responsibility and they are supposed to operate the reverse charge.
- If you are selling an *e-service* to a consumer, you need to register for the Mini One Stop Shop in *any* EU member state (you can pick whichever you please), and charge VAT at the rate applicable in the customer's member state.

Case 4: The customer is outside the EU

In this case, the sale is outside of the scope of EU VAT. Note that this is different from being exempt, in that you *are* allowed to reclaim any input tax associated with the sale.

4.2 vat package

4.2.1 Functions

4.2.2 Classes

class vat.**VIESResponseBase**
class vat.**VIESResponse**
class vat.**VIESApproxResponse**

Imported from `vat.vies` for convenient access.

See `vat.vies.VIESResponseBase`, `vat.vies.VIESResponse` and `vat.vies.VIESApproxResponse`.

class vat.**VIESException**
class vat.**VIESSOAPException**
class vat.**VIESHTTPException**

Imported from `vat.vies` for convenient access.

See `vat.vies.VIESException`, `vat.vies.VIESSOAPException` and `vat.vies.VIESHTTPException`.

class vat.**VRWSException**
class vat.**VRWSErrorException**
class vat.**VRWSSOAPException**
class vat.**VRWSHTTPException**

Imported from `vat.vrws` for convenient access.

See `vat.vrws.VRWSException`, `vat.vrws.VRWSErrorException`, `vat.vrws.VRWSSOAPException` and `vat.vrws.VRWSHTTPException`.

4.2.3 Data

vat.**member_states**

A list of all the member states of the European Union, as `MemberState` objects.

vat.**BROADCASTING**
vat.**TELECOMS**
vat.**ESERVICES**

Category constants. There may be other categories besides these ones, but these are known to be defined at present.

4.3 vat.vies package

4.3.1 Functions

`vat.vies.`**`check_vat`**`()`

Check a VAT number using VIES. Returns a *`VIESResponse`* on success, or in case of error raises an exception.

The complete VAT number, including the two character EU VAT country code.

Note: You should probably use `vat.check_details()` rather than this function.

`vat.vies.`**`check_vat_approx`**`(`*vat_number*, *extra={}*, *requester=None*`)`

Check a VAT number using VIES, passing in additional information about the entity being checked. Returns a *`VIESApproxResponse`* on success, or raises an exception.

The complete VAT number, including the two character EU VAT country code.

A dictionary containing additional information to be passed to VIES to check. Available items are:

Key	Value
name	The name of the trader.
company-type	The company type, if applicable.
street	The street address of the trader.
postcode	The postal code of the trader.
city	The city of the trader.

Note that company types are language and country specific.

If the member state does fuzzy matching, passing these details will allow their system to try to match them.

The requester's VAT number (optional).

Note: You should probably use `vat.check_details()` rather than this function.

4.3.2 Classes

`class` `vat.vies.`**`VIESException`**

The base class of all exceptions raised by this module.

`class` `vat.vies.`**`VIESSOAPException`**

Represents a SOAP fault encountered when trying to talk to VIES.

`code`

The SOAP fault code.

`string`

A description of the fault.

`actor`

Used to indicate the source of the fault.

`detail`

Provides additional information about the fault.

class `vat.vies.`**`VIESResponseBase`**

The base class of all VIES response classes.

`country`

The EU VAT country code.

`vat_number`

The VAT number *excluding* the country code.

`request_date`

The date on which the request was submitted to VIES.

`valid`

True if the VAT number is valid, False otherwise.

class `vat.vies.`**`VIESResponse`**

Represents the response from VIES to a basic request.

`name`

The trader's name, if provided (not all member states do).

`address`

The trader's address, if provided (not all member states do).

class `vat.vies.`**`VIESApproxResponse`**

Represents the response from VIES to a request to verify a complete set of details, including the trader's name and address.

Note that the details contained in the response are dependent on the member state in question. Some member states try to fuzzy-match the information you pass in; other member states supply the trader information themselves; and there is at least one member state (Germany) that at time of writing does neither and (worse) returns the information you gave it.

`trader_info`

A dictionary containing (any of) the following elements:

Key	Value
name	The name of the trader.
company-type	The company type, if applicable.
address	The full address of the trader.
street	The street address of the trader.
postcode	The postal code of the trader.
city	The city of the trader.

Note that the address may be specified using either the "address" field or the individual "street", "postcode" and "city" fields.

`trader_match_info`

A dictionary containing the same keys as above, but for each item the value is one of

Value	Meaning
MATCH_VALID	The details matched those supplied.
MATCH_INVALID	The details did not match those supplied.
MATCH_NOT_PROCESSED	The details were not processed.

request_id

A unique request ID generated by the VIES system. This can be stored and later used as proof that VIES was checked for these details.

4.3.3 Constants

vat.vies.**MATCH_VALID**
vat.vies.**MATCH_INVALID**
vat.vies.**MATCH_NOT_PROCESSED**
Matching results.

4.4 vat.vrws package

4.4.1 Functions

vat.vrws.**get_rates**()
Obtain VAT rates for the specified country. Returns a *Rates* on success, or in case of error raises an exception.

The two character EU VAT country code.

If set, only dates in force on the specified date will be returned (optional).

If *True*, the default, include reduced rates (optional).

If *True*, the default, include category rates (optional).

If *True*, the default, include regional rates (optional).

Note: You should probably use the `vat.RateCache` object rather than this function. Not only does this avoid hitting the EU web service more often than necessary, but `vat.RateCache` keeps back-up information on reduced rates that were not (as of 1st September 2015) available through the web service.

vat.vrws.**get_changes**()
Retrieve any VAT rates that changed during the specified period. Returns a *Rates* on success, or in case of error raises an exception.

Returns VAT rates that changed after this date.

Returns only VAT rates that changed before this date (optional).

A two character EU VAT country code (optional). If specified, restricts results to that country only.

4.4.2 Classes

class vat.vrws.**VRWSException**
The base class of all exceptions raised by this module.

class `vat.vrws.`**`VRWSSOAPException`**

Represents a SOAP fault encountered when trying to talk to VIES.

`code`

The SOAP fault code.

`string`

A description of the fault.

`actor`

Used to indicate the source of the fault.

`detail`

Provides additional information about the fault.

class `vat.vrws.`**`VRWSErrorException`**

Represents an error raised by the VRWS system.

`code`

A numeric code identifying the error.

`reason`

A string describing the error.

class `vat.vrws.`**`Rates`**

A collection of VAT rate information.

`types`

A dictionary keyed on rate type. Each entry is a list of applicable `Rate` objects. The module includes constants for the following rate types:

Constant	Meaning
STANDARD	Standard rate
REDUCED	Reduced rate

Note: In general there can be more than one rate per type. You may need to inspect the attributes of the `Rate` object to determine which rate is applicable.

`categories`

A dictionary keyed on rate category. Each entry is a list of applicable `Rate` objects. The module includes constants for the following categories:

Constant	Meaning
BROADCASTING	Broadcasting
TELECOMS	Telecommunication Services
ESERVICES	E-Services

Note: Again, there can be multiple rates in each category, and you may need to use `Rate.detail` to distinguish them.

regions

A dictionary keyed on region. Each entry is a *Rates* object.

class vat.vrws.**Rate**

Represents an individual VAT rate.

rate

A decimal.Decimal holding the percentage rate.

application_date

The date from which this rate applies.

detail

Any notes that apply to this rate. For instance, if this rate applies to a specific type of product, or has special requirements associated with it.

Note: There is no standard format for the *Rate.detail* attribute and it may be written in the language of the associated member state.

4.4.3 Constants

vat.vrws.**STANDARD**
vat.vrws.**REDUCED**

VAT rate types. There may be additional rate types in future.

vat.vrws.**BROADCASTING**
vat.vrws.**TELECOMS**
vat.vrws.**ESERVICES**

VAT categories. There may be additional categories in future.

4.5 vat.gb package

The *vat.gb* module contains VAT code specific to the United Kingdom.

4.5.1 EC Sales Lists - vat.gb.ecsl

EC Sales Lists are a requirement where you are making B2B sales to other member states within the European Union. See VAT: how to report your EU sales for more information.

vat.gb.ecsl.**B2B_GOODS**
vat.gb.ecsl.**B2B_INTERMEDIARY**
vat.gb.ecsl.**B2B_SERVICES**

Used to specify the type of sale being reported.

4.5.2 Mini One Stop Shop - vat.gb.moss

The Mini One Stop Shop is a service run by HMRC that allows you to report sales of e-services into EU member states without requiring you to register separately in every tax jurisdiction. See VAT on digital services in the EU for more information.

`vat.gb.moss.`**`STANDARD_RATE`**
`vat.gb.moss.`**`REDUCED_RATE`**

Used to specify the type of VAT rate being used.

4.6 Indices and tables

- genindex
- modindex
- search

Python Module Index

Index