identifiers Documentation

Release 0.4.1

Michael Amrhein

Dec 11, 2020

Contents

1 Overview **3**
- 1.1 Identifiers standardized by GS1 . . . 3
- 1.2 Identifiers for publications . . . 3
- 1.3 Identifiers for banks and bank accounts . . . 4
- 1.4 Identifiers for exchanges and exchange traded financial assets . . . 4
- 1.5 Identifiers for Value Added Taxation . . . 4

2 Common Features **5**

3 Identifiers standardized by GS1 **7**

4 Identifiers for publications **15**

5 Identifiers for banks and bank accounts **19**

6 Identifiers for exchanges and exchange traded financial assets **23**

7 European Union VAT Registration Number **25**

Python Module Index **27**

Index **29**

This package provides classes that can be used to create and check identifiers for unique objects or unique classes of objects according to international standards.

Definition from Wikipedia:

"An identifier is a name that identifies (that is, labels the identity of) either a unique object or a unique class of objects, where the "object" or class may be an idea, physical [countable] object (or class thereof), or physical [noncountable] substance (or class thereof). The abbreviation ID often refers to identity, identification (the process of identifying), or an identifier (that is, an instance of identification). An identifier may be a word, number, letter, symbol, or any combination of those."

Contents:

CHAPTER 1

Overview

1.1 Identifiers standardized by GS1

See http://www.gs1.org/id-keys.

- Global Trade Item Number (GTIN)

  Used to identify products and services.

  Variants: GTIN12, GTIN13, GTIN14

- Global Location Number (GLN)

  Used to identify parties and locations, for example companies, warehouses, factories and stores.

- Global Shipment Identification Number (GSIN)

  Used to identify shipments, i. e. logistics units delivered to a customer together.

- Serial Shipping Container Code (SSCC)

  Used to identify logistics units, for example loads on pallets, roll cages or parcels.

1.2 Identifiers for publications

- International Standard Book Number (ISBN)

  A unique international identifier for monographic publications.

  See http://www.isbn-international.org.

- International Standard Music Number (ISMN)

  A unique international identifier of all notated music publications, whether available for sale, hire or gratis, whether a part, a score, or an element in a multi-media kit.

  See http://www.ismn-international.org.

- International Standard Serial Number (ISSN)

  Used to identify newspapers, journals, magazines and periodicals of all kinds and on all media - print and electronic.

  See http://www.issn.org.

1.3 Identifiers for banks and bank accounts

  See http://www.swift.com.

- Business Identifier Code (BIC)

  Used to identify financial and non-financial institutions.

- International Bank Account Number (IBAN)

  An internationally agreed system of identifying bank accounts across national borders.

1.4 Identifiers for exchanges and exchange traded financial assets

- Market Identifier Code (MIC)

  A unique identification code used to identify securities trading exchanges, regulated and non-regulated trading markets.

  See https://www.iso20022.org/10383/iso-10383-market-identifier-codes

- International Securities Identification Number (ISIN)

  Uniquely identifies a tradable financial asset, a.k.a security.

  See http://www.anna-web.org/standards/isin-iso-6166/.

1.5 Identifiers for Value Added Taxation

- European Union VAT Registration Number

  The VAT Registration Number is used to identify natural and legal persons located in the European Union for purposes of VAT handling.

  See http://ec.europa.eu/taxation_customs/business/vat

CHAPTER 2

Common Features

class identifiers.**Identifier**(**args*, ***kwds*)

Abstract base class for identifiers.

__copy__() → identifiers.identifier.Identifier

copy(self)

Returns self (identifiers are immutable).

__deepcopy__(*memo: Optional[Mapping[KT, VT_co]]*) → identifiers.identifier.Identifier

deepcopy(self)

Returns self (identifiers are immutable).

__hash__() → int

hash(self)

__repr__() → str

repr(self)

__str__() → str

str(self)

CHAPTER 3

Identifiers standardized by GS1

class identifiers.**GLN**(**args*)

Global Location Number

Used to identify parties and locations, for example companies, warehouses, factories and stores.

An instance of GLN can be created in two ways, by providing a Unicode string representation of a GLN or by providing the GLN's elements as separate arguments.

1. Form

Parameters **id** (*str*) – string representation of a GLN (with or without a check digit)

If *id* does not include a check digit, it is calculated and appended automatically.

Returns instance of *GLN*

Raises

- **TypeError** – given *id* is not an instance of *str*
- **ValueError** – given *id* contains character(s) other than digits 0-9
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – length of given *id* not valid

2. Form

Parameters

- **company_prefix** (*str*) – number identifying the company issuing the identifier, starting with a 3-digit GS1 prefix
- **location_reference** (*str*) – number identifying the location
- **check_digit** (*str*) – 1-digit number (optional)

If *check_digit* is omitted, it is calculated and appended automatically.

Returns instance of *GLN*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not an instance of *str*
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – length of given *company_prefix* not valid
- **ValueError** – given *company_prefix* not valid
- **ValueError** – length of given *location_reference* not valid
- **ValueError** – invalid check digit

elements() → Tuple[str]
Return the identifier's elements as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

company_prefix
Return the identifier's company prefix part.

gs1_prefix
Return the identifier's GS1 prefix part.

location_reference
Return the identifier's reference part.

class identifiers.**GTIN12**(**args*)
Global Trade Item Number (12 digits)

Used to identify products and services.

An instance of GTIN12 can be created in two ways, by providing a Unicode string representation of a GTIN12 or by providing the GTIN12's elements as separate arguments.

1. Form

Parameters **id** (*str*) – string representation of a GTIN12 (with or without a check digit)

If *id* does not include a check digit, it is calculated and appended automatically.

Returns instance of *GTIN12*

Raises

- **TypeError** – given *id* is not an instance of *str*
- **ValueError** – given *id* contains character(s) other than digits 0-9
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – length of given *id* not valid

2. Form

Parameters

- **company_prefix** (*str*) – number identifying the company issuing the identifier, starting with a 3-digit GS1 prefix
- **item_reference** (*str*) – number identifying the trade item
- **check_digit** (*str*) – 1-digit number (optional)

If *check_digit* is omitted, it is calculated and appended automatically.

Returns instance of `GTIN12`

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not an instance of *str*
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – length of given *company_prefix* not valid
- **ValueError** – given *company_prefix* not valid
- **ValueError** – length of given *item_reference* not valid
- **ValueError** – invalid check digit

elements() → Tuple[str]
Return the identifier's elements as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

company_prefix
Return the identifier's company prefix part.

gs1_prefix
Return the identifier's GS1 prefix part.

item_reference
Return the identifier's reference part.

class identifiers.**GTIN13**(**args*)
Global Trade Item Number (13 digits)

Used to identify products and services.

An instance of GTIN13 can be created in two ways, by providing a Unicode string representation of a GTIN13 or by providing the GTIN13's elements as separate arguments.

1. Form

Parameters **id** (*str*) – string representation of a GTIN13 (with or without a check digit)

If *id* does not include a check digit, it is calculated and appended automatically.

Returns instance of `GTIN13`

Raises

- **TypeError** – given *id* is not an instance of *str*
- **ValueError** – given *id* contains character(s) other than digits 0-9
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – length of given *id* not valid

2. Form

Parameters

- **company_prefix** (*str*) – number identifying the company issuing the identifier, starting with a 3-digit GS1 prefix
- **item_reference** (*str*) – number identifying the trade item
- **check_digit** (*str*) – 1-digit number (optional)

If *check_digit* is omitted, it is calculated and appended automatically.

Returns instance of *GTIN13*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not an instance of *str*
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – length of given *company_prefix* not valid
- **ValueError** – given *company_prefix* not valid
- **ValueError** – length of given *item_reference* not valid
- **ValueError** – invalid check digit

elements() → Tuple[str]
Return the identifier's elements as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

company_prefix
Return the identifier's company prefix part.

gs1_prefix
Return the identifier's GS1 prefix part.

item_reference
Return the identifier's reference part.

class identifiers.**GTIN14**(**args*)
Global Trade Item Number (14 digits)

Used to identify products and services.

An instance of GTIN14 can be created in three ways, by providing a GTIN12 or GTIN13, by providing a string representation of a GTIN14 or by providing the GTIN14's elements as separate arguments.

1. Form

Parameters **gtin** (*GTIN12* or *GTIN13*) – GTIN to be expanded to a GTIN14

Returns instance of *GTIN14*

2. Form

Parameters **id** (*str*) – string representation of a GTIN14 (with or without a check digit)

If *id* does not include a check digit, it is calculated and appended automatically.

Returns instance of *GTIN14*

Raises

- **TypeError** – given *id* is not an instance of *str*
- **ValueError** – given *id* contains character(s) other than digits 0-9
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – length of given *id* not valid

3. Form

Parameters

- **level_indicator** (*str*) – 1-digit number providing additional name space
- **company_prefix** (*str*) – number identifying the company issuing the identifier, starting with a 3-digit GS1 prefix
- **item_reference** (*str*) – number identifying the trade item
- **check_digit** (*str*) – 1-digit number (optional)

If *check_digit* is omitted, it is calculated and appended automatically.

Returns instance of *GTIN14*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not a Unicode string
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – length of given *company_prefix* not valid
- **ValueError** – given *company_prefix* not valid
- **ValueError** – length of given *item_reference* not valid
- **ValueError** – invalid check digit

elements() → Tuple[str]
Return the identifier's elements as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

company_prefix
Return the identifier's company prefix part.

gs1_prefix
Return the identifier's GS1 prefix part.

item_reference
Return the identifier's reference part.

level_indicator
Return the identifier's level indicator (i.e. the first digit).

class identifiers.**GSIN**(**args*)
Global Shipment Identification Number

Used to identify shipments, i. e. logistics units delivered to a customer together.

An instance of GSIN can be created in two ways, by providing a Unicode string representation of a GSIN or by providing the GSIN's elements as separate arguments.

1. Form

Parameters **id** (*str*) – string representation of a GSIN (with or without a check digit)

If *id* does not include a check digit, it is calculated and appended automatically.

Returns instance of *GSIN*

Raises

- **TypeError** – given *id* is not an instance of *str*
- **ValueError** – given *id* contains character(s) other than digits 0-9
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – length of given *id* not valid

2. Form

Parameters

- **company_prefix** (*str*) – number identifying the company issuing the identifier, starting with a 3-digit GS1 prefix
- **shipper_reference** (*str*) – number identifying the shipment
- **check_digit** (*str*) – 1-digit number (optional)

If *check_digit* is omitted, it is calculated and appended automatically.

Returns instance of *GSIN*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not an instance of *str*
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – length of given *company_prefix* not valid
- **ValueError** – given *company_prefix* not valid
- **ValueError** – length of given *shipper_reference* not valid
- **ValueError** – invalid check digit

elements() → Tuple[str]
Return the identifier's elements as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

company_prefix
Return the identifier's company prefix part.

gs1_prefix
Return the identifier's GS1 prefix part.

shipper_reference
Return the identifier's reference part.

class identifiers.**SSCC**(**args*)

Serial Shipping Container Code

Used to identify logistics units, for example loads on pallets, roll cages or parcels.

An instance of SSCC can be created in two ways, by providing a Unicode string representation of a SSCC or by providing the SSCC's elements as separate arguments.

1. Form

Parameters **id** (*str*) – string representation of a SSCC (with or without a check digit)

If *id* does not include a check digit, it is calculated and appended automatically.

Returns instance of *SSCC*

Raises

- **TypeError** – given *id* is not an instance of *str*
- **ValueError** – given *id* contains character(s) other than digits 0-9
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – length of given *id* not valid

2. Form

Parameters

- **extension_digit** (*str*) –
- **company_prefix** (*str*) – number identifying the company issuing the identifier, starting with a 3-digit GS1 prefix
- **serial_reference** (*str*) – number identifying the shipping container
- **check_digit** (*str*) – 1-digit number (optional)

If *check_digit* is omitted, it is calculated and appended automatically.

Returns instance of *SSCC*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not an instance of *str*
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – length of given *company_prefix* not valid
- **ValueError** – given *company_prefix* not valid
- **ValueError** – length of given *serial_reference* not valid
- **ValueError** – invalid check digit

elements() → Tuple[str]

Return the identifier's elements as tuple.

separated(*separator: Optional[str] = '-'*) → str

Return a string representation of the identifier with its elements separated by the given separator.

check_digit

Return the identifier's check digit.

company_prefix
Return the identifier's company prefix part.

extension_digit
Return the identifier's extension digit (i.e. the first digit).

gs1_prefix
Return the identifier's GS1 prefix part.

serial_reference
Return the identifier's reference part.

CHAPTER 4

Identifiers for publications

class identifiers.**ISBN**(**args*)

International Standard Book Number

The ISBN is a unique international identifier for monographic publications.

Each ISBN consists of 13 digits. It contains a 3-digits GS1 prefix ('978' or '979'), a Registration Group, denoting the local registration authority, a number denoting the Registrant, i. e. the company which registered the ISBN, a number denoting the Publication and a check digit.

An instance of ISBN can be created in two ways, by providing a Unicode string representation of an ISBN or by providing a GS1 prefix, a registration group, a registrant, a publication number and, optionally, a check digit.

1. Form

Parameters **id** (*str*) – string representation of an ISBN

Returns instance of *ISBN*

Raises

- **TypeError** – given *id* is not a str
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – given *id* does not follow the format required for an ISBN

2. Form

Parameters

- **gs1_prefix** (*str*) – 3-digit GS1 prefix
- **registration_group** (*str*) – number identifying the local registration authority
- **registrant** (*str*) – number identifying the publishing company
- **publication** (*str*) – number identifying the publication
- **check_digit** (*str*) – 1-digit number (optional)

Returns instance of *ISBN*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not a Unicode string
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – given *gs1_prefix* not valid
- **ValueError** – given *registration_group* not valid
- **ValueError** – given *registrant* not valid
- **ValueError** – combined length of arguments not valid
- **ValueError** – wrong check digit

elements() → Tuple[str, str, str, str, str]
Return the identifier's elements (gs1_prefix, registration_group, registrant, publication, check_digit) as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

gs1_prefix
Return the identifier's GS1 prefix part.

publication
Return the identifier's reference part.

registrant
Return the Registrant of the identifier.

registration_group
Return the Registration Group of the identifier.

class identifiers.**ISMN**(**args*)
International Standard Music Number

The ISMN is a unique international identifier of all notated music publications, whether available for sale, hire or gratis, whether a part, a score, or an element in a multi-media kit.

Each ISMN consists of 13 digits. It contains a 3-digits GS1 prefix ('979'), a Registration Group ('0'), a number denoting the Registrant, i. e. the company which registered the ISMN, a number denoting the Publication and a check digit.

An instance of ISMN can be created in two ways, by providing a Unicode string representation of an ISMN or by providing a GS1 prefix, a registration group, a registrant, a publication number and, optionally, a check digit.

1. Form

Parameters **id**(*str*) – string representation of an ISMN

Returns instance of *ISMN*

Raises

- **TypeError** – given *id* is not a str
- **ValueError** – given *id* contains wrong check digit
- **ValueError** – given *id* does not follow the format required for an ISMN

2. Form

Parameters

- **gs1_prefix** (*str*) – 3-digit GS1 prefix
- **registration_group** (*str*) – number identifying the local registration authority
- **registrant** (*str*) – number identifying the publishing company
- **publication** (*str*) – number identifying the publication
- **check_digit** (*str*) – 1-digit number (optional)

Returns instance of *ISMN*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – a given argument is not a Unicode string
- **ValueError** – a given argument contains character(s) other than digits 0-9
- **ValueError** – given *gs1_prefix* not valid
- **ValueError** – given *registration_group* not valid
- **ValueError** – given *registrant* not valid
- **ValueError** – combined length of arguments not valid
- **ValueError** – wrong check digit

elements() → Tuple[str, str, str, str, str]
Return the identifier's elements (gs1_prefix, registration_group, registrant, publication, check_digit) as tuple.

separated(*separator: Optional[str] = '-'*) → str
Return a string representation of the identifier with its elements separated by the given separator.

check_digit
Return the identifier's check digit.

gs1_prefix
Return the identifier's GS1 prefix part.

publication
Return the identifier's reference part.

registrant
Return the Registrant of the identifier.

registration_group
Return the Registration Group of the identifier.

class identifiers.**ISSN**(*digits: str*)
International Standard Serial Number

The ISSN is used to identify newspapers, journals, magazines and periodicals of all kinds and on all media - print and electronic.

Parameters **digits** (*str*) – string representation of the ISSN

The argument must be a string with 8 digits, with 7 digits or with 7 digits followed by an 'X', optionally separated by a blank or a hyphen after the fourth digit.

Returns instance of *ISSN*

Raises

- **ValueError** – given arg has invalid format
- **ValueError** – given arg contains wrong check digit

as_gtin(*addon=None*) → identifiers.bookland.ISSN13
Return GTIN13 created from *self* + *addon*.

separated(*separator: Optional[str] = '-'*) → str
Return ISSN as string, optionally split by *separator*.

check_digit
Return the ISSN's check digit.

raw_number
Return the ISSN without check digit.

CHAPTER 5

Identifiers for banks and bank accounts

class identifiers.**BIC**(*bic: str*)

Business Identifier Code

Used to identify financial and non-financial institutions.

Each BIC consists of the Party Prefix (4 alpha-numeric), the Country Code (2 alpha) and the Party Suffix (2 alpha-numeric), optionally followed by the Branch Code (3 alpha-numeric).

Parameters **bic** (*str*) – string representation of the BIC

Returns *BIC* instance

Raises

- **TypeError** – given *bic* is not a *Unicode string*
- **ValueError** – given *bic* (stripped) does not contain 8 or 11 characters
- **ValueError** – given *bic* contains characters other than A-Z or 0-9
- **ValueError** – given *bic* contains an unknown country code

elements() → Tuple[str, str, str, str]

Return the BIC's Party Prefix, Country Code, Party Suffix and Branch Code as a tuple.

branch_code

Return the BIC's Branch Code (maybe empty).

country_code

Return the BIC's Country Code.

party_prefix

Return the BIC's Party Prefix.

party_suffix

Return the BIC's Party Suffix.

class identifiers.**IBAN**(**args*)

International Bank Account Number

An internationally agreed system of identifying bank accounts across national borders.

Each IBAN consists of a two-letter ISO 3166-1 Country Code, followed by two check digits and up to thirty alphanumeric characters for a BBAN (Basic Bank Account Number) which has a fixed length per country and, included within it, a bank identifier with a fixed position and a fixed length per country. The check digits are calculated based on the scheme defined in ISO/IEC 7064 (MOD97-10).

Instances of `IBAN` can be created in two ways, by providing a Unicode string representation of an IBAN or by providing a country code, a bank identifier and a bank account number.

1. Form

Parameters **iban** (*Unicode string*) – string representation of an IBAN

Returns instance of `IBAN`

Raises

- **TypeError** – given *iban* is not a *Unicode string*
- **ValueError** – given *iban* contains an unknown country code
- **ValueError** – given *iban* contains wrong check digits
- **ValueError** – given *iban* does not follow the format required for the given country code

2. Form

Parameters

- **country_code** (*Unicode string*) – 2-character country code according to ISO 3166
- **bank_identifier** (*Unicode string* or *int*) – code identifying the bank maintaining the account
- **bank_account_number** (*Unicode string* or *int*) – code identifying the account (within the namespace of the bank)

Returns instance of `IBAN`

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – given *country_code* is not a *Unicode string*
- **ValueError** – given *country_code* contains an invalid or unknown country code
- **TypeError** – given *bank_identifier* is not a *Unicode string* or not an *int*
- **ValueError** – length of given *bank_identifier* not valid for the given country code
- **TypeError** – given *bank_account_number* is not a *Unicode string* or not an *int*
- **ValueError** – length of given *bank_account_number* not valid for the given country code
- **ValueError** – given *bank_identifier* and *bank_account_number* do not follow the BBAN format required for the given country code

elements() → Tuple[str, str, str, str]

Return the IBAN's Country Code, check digits, Bank Identifier and Bank Account Number as tuple.

bank_account_number

Return the IBAN's Bank Account Number.

bank_identifier

Return the IBAN's Bank Identifier.

check_digits
Return the IBAN's check digits.

country_code
Return the IBAN's Country Code.

CHAPTER 6

Identifiers for exchanges and exchange traded financial assets

class identifiers.**MIC**(*mic: str*)

Market Identifier Code

A unique identification code used to identify securities trading exchanges, regulated and non-regulated trading markets.

Each MIC is a four alpha character code, defined in ISO 10383.

Parameters **mic** (*str*) – string representation of the MIC

Returns *MIC* instance

Raises

- **TypeError** – given *mic* is not an instance of str
- **ValueError** – given *mic* not found in the registry

class identifiers.**ISIN**(**args*)

International Securities Identification Number

An International Securities Identification Number uniquely identifies a tradable financial asset, a.k.a security.

As defined in ISO 6166, each ISIN consists of a two-letter ISO 3166-1 Country Code for the issuing country, followed by nine alpha-numeric characters (the National Securities Identifying Number, or NSIN, which identifies the security), and one numerical check digit, calculated by the Luhn algorithm.

Instances of *ISIN* can be created in two ways, by providing a Unicode string representation of an ISIN or by providing a country code and a national securities identifying number.

1. Form

Parameters **isin** (*str*) – string representation of an ISIN

Returns instance of *ISIN*

Raises

- **TypeError** – given *isin* is not a *Unicode string*
- **ValueError** – given *isin* contains an unknown country code

- **ValueError** – given *isin* contains a wrong check digit
- **ValueError** – given *isin* must be 12 characters long
- **ValueError** – given *isin* contains invalid character(s)

2. Form

Parameters

- **country_code** (*str*) – 2-character country code according to ISO 3166
- **nsin** (*str*) – national securities identifying number

Returns instance of *ISIN*

Raises

- **TypeError** – invalid number of arguments
- **TypeError** – given *country_code* is not a *Unicode string*
- **ValueError** – given *country_code* contains an invalid or unknown country code
- **TypeError** – given *nsin* is not a *Unicode string*
- **ValueError** – length of given *nsin* not valid
- **ValueError** – given *nsin* contains invalid character(s)

elements() → Tuple[str, str, str]

Return the ISIN's Country Code, National Securities Identifying Number and check digit as tuple.

check_digit

Return the ISIN's check digits.

country_code

Return the ISIN's Country Code.

nsin

Return the ISIN's National Securities Identifying Number.

CHAPTER 7

European Union VAT Registration Number

class identifiers.**EUVATId**(*vat_id: str*)

European Union VAT Registration Number

The VAT Registration Number is used to identify natural and legal persons located in the European Union for purposes of VAT handling.

Each VAT Id begins with the two-letter ISO 3166-1 code(*) for the country where it is registered, followed by a registration code. The structure of the latter depends on the country.

(*) As an exception, European Directive 2001/115 allowed 'EL' as synonym for 'GR'.

Instances of EUVATId are created from a string containing the country code and the registration code.

Depending on the country, different patterns and check algorithms are applied to validate an id.

Parameters **vat_id** (*Unicode string*) – string representation of a VAT-Id

Returns instance of *EUVATId*

Raises

- **TypeError** – given *vat_id* is not a *Unicode string*
- **ValueError** – given *vat_id* contains an unknown country code
- **ValueError** – given *vat_id* does not follow the format required for the given country code
- **ValueError** – given *vat_id* does not pass the checks for the given country code

elements() → Tuple[str, str]

Return the VAT Id's country code and registration code as a tuple.

country_code

Return the VAT Id's country code.

registration_code

Return the VAT Id's registration code.

genindex

Python Module Index

i

identifiers, 5

Index

Symbols

__copy__() (*identifiers.Identifier method*), 5
__deepcopy__() (*identifiers.Identifier method*), 5
__hash__() (*identifiers.Identifier method*), 5
__repr__() (*identifiers.Identifier method*), 5
__str__() (*identifiers.Identifier method*), 5

A

as_gtin() (*identifiers.ISSN method*), 18

B

bank_account_number (*identifiers.IBAN attribute*), 20
bank_identifier (*identifiers.IBAN attribute*), 20
BIC (*class in identifiers*), 19
branch_code (*identifiers.BIC attribute*), 19

C

check_digit (*identifiers.GLN attribute*), 8
check_digit (*identifiers.GSIN attribute*), 12
check_digit (*identifiers.GTIN12 attribute*), 9
check_digit (*identifiers.GTIN13 attribute*), 10
check_digit (*identifiers.GTIN14 attribute*), 11
check_digit (*identifiers.ISBN attribute*), 16
check_digit (*identifiers.ISIN attribute*), 24
check_digit (*identifiers.ISMN attribute*), 17
check_digit (*identifiers.ISSN attribute*), 18
check_digit (*identifiers.SSCC attribute*), 13
check_digits (*identifiers.IBAN attribute*), 20
company_prefix (*identifiers.GLN attribute*), 8
company_prefix (*identifiers.GSIN attribute*), 12
company_prefix (*identifiers.GTIN12 attribute*), 9
company_prefix (*identifiers.GTIN13 attribute*), 10
company_prefix (*identifiers.GTIN14 attribute*), 11
company_prefix (*identifiers.SSCC attribute*), 13
country_code (*identifiers.BIC attribute*), 19
country_code (*identifiers.EUVATId attribute*), 25
country_code (*identifiers.IBAN attribute*), 21
country_code (*identifiers.ISIN attribute*), 24

E

elements() (*identifiers.BIC method*), 19
elements() (*identifiers.EUVATId method*), 25
elements() (*identifiers.GLN method*), 8
elements() (*identifiers.GSIN method*), 12
elements() (*identifiers.GTIN12 method*), 9
elements() (*identifiers.GTIN13 method*), 10
elements() (*identifiers.GTIN14 method*), 11
elements() (*identifiers.IBAN method*), 20
elements() (*identifiers.ISBN method*), 16
elements() (*identifiers.ISIN method*), 24
elements() (*identifiers.ISMN method*), 17
elements() (*identifiers.SSCC method*), 13
EUVATId (*class in identifiers*), 25
extension_digit (*identifiers.SSCC attribute*), 14

G

GLN (*class in identifiers*), 7
gs1_prefix (*identifiers.GLN attribute*), 8
gs1_prefix (*identifiers.GSIN attribute*), 12
gs1_prefix (*identifiers.GTIN12 attribute*), 9
gs1_prefix (*identifiers.GTIN13 attribute*), 10
gs1_prefix (*identifiers.GTIN14 attribute*), 11
gs1_prefix (*identifiers.ISBN attribute*), 16
gs1_prefix (*identifiers.ISMN attribute*), 17
gs1_prefix (*identifiers.SSCC attribute*), 14
GSIN (*class in identifiers*), 11
GTIN12 (*class in identifiers*), 8
GTIN13 (*class in identifiers*), 9
GTIN14 (*class in identifiers*), 10

I

IBAN (*class in identifiers*), 19
Identifier (*class in identifiers*), 5
identifiers (*module*), 5
ISBN (*class in identifiers*), 15
ISIN (*class in identifiers*), 23
ISMN (*class in identifiers*), 16
ISSN (*class in identifiers*), 17

item_reference (*identifiers.GTIN12 attribute*), 9
item_reference (*identifiers.GTIN13 attribute*), 10
item_reference (*identifiers.GTIN14 attribute*), 11

L

level_indicator (*identifiers.GTIN14 attribute*), 11
location_reference (*identifiers.GLN attribute*), 8

M

MIC (*class in identifiers*), 23

N

nsin (*identifiers.ISIN attribute*), 24

P

party_prefix (*identifiers.BIC attribute*), 19
party_suffix (*identifiers.BIC attribute*), 19
publication (*identifiers.ISBN attribute*), 16
publication (*identifiers.ISMN attribute*), 17

R

raw_number (*identifiers.ISSN attribute*), 18
registrant (*identifiers.ISBN attribute*), 16
registrant (*identifiers.ISMN attribute*), 17
registration_code (*identifiers.EUVATId attribute*), 25
registration_group (*identifiers.ISBN attribute*), 16
registration_group (*identifiers.ISMN attribute*), 17

S

separated() (*identifiers.GLN method*), 8
separated() (*identifiers.GSIN method*), 12
separated() (*identifiers.GTIN12 method*), 9
separated() (*identifiers.GTIN13 method*), 10
separated() (*identifiers.GTIN14 method*), 11
separated() (*identifiers.ISBN method*), 16
separated() (*identifiers.ISMN method*), 17
separated() (*identifiers.ISSN method*), 18
separated() (*identifiers.SSCC method*), 13
serial_reference (*identifiers.SSCC attribute*), 14
shipper_reference (*identifiers.GSIN attribute*), 12
SSCC (*class in identifiers*), 12