
django-localflavor Documentation

Release 1.4

Django Software Foundation and individual contributors

January 06, 2017

1	Installation	3
2	Internationalization	5
3	Adding flavors	7
4	Releases	9
4.1	Roadmap	9
5	How to migrate	11
6	Backwards compatibility	13
7	Indices and tables	15
7.1	Authors	15
7.2	Changelog	17
7.3	Argentina (ar)	21
7.4	Austria (at)	22
7.5	Australia (au)	22
7.6	Belgium (be)	24
7.7	Bulgaria (bg)	25
7.8	Brazil (br)	26
7.9	Canada (ca)	27
7.10	Switzerland (ch)	28
7.11	Chile (cl)	29
7.12	China (cn)	29
7.13	Colombia (co)	30
7.14	Czech Republic (cz)	31
7.15	Germany (de)	31
7.16	Denmark (dk)	32
7.17	Ecuador (ec)	33
7.18	Estonia (ee)	33
7.19	Spain (es)	34
7.20	Finland (fi)	35
7.21	France (fr)	36
7.22	Great Britain (gb)	37
7.23	Greece (gr)	38
7.24	Hong Kong (hk)	39
7.25	Croatia (hr)	39

7.26	Hungary (hu)	41
7.27	Indonesia (id)	41
7.28	Ireland (ie)	42
7.29	Israel (il)	42
7.30	India (in)	43
7.31	Iceland (is)	44
7.32	Italy (it)	45
7.33	Japan (jp)	46
7.34	Kuwait (kw)	47
7.35	Lithuania (lt)	47
7.36	Latvia (lv)	48
7.37	Macedonia (mk)	49
7.38	Malta (mt)	50
7.39	Mexico (mx)	50
7.40	The Netherlands (nl)	53
7.41	Norway (no)	54
7.42	New Zealand (nz)	55
7.43	Peru (pe)	56
7.44	Pakistan (pk)	56
7.45	Poland (pl)	57
7.46	Portugal (pt)	59
7.47	Paraguay (py)	60
7.48	Romania (ro)	60
7.49	Russia (ru)	62
7.50	Sweden (se)	62
7.51	Singapore (sg)	63
7.52	Slovenia (si)	64
7.53	Slovakia (sk)	64
7.54	Tunisia (tn)	65
7.55	Turkey (tr)	65
7.56	United States of America (us)	66
7.57	Uruguay (uy)	69
7.58	South Africa (za)	70
7.59	Generic helpers	70

Python Module Index **75**

django-localflavor is a collection of assorted pieces of code that are useful for particular countries or cultures. These are called the “local flavor” add-ons and live in the *localflavor* package.

Inside that package, country- or culture-specific code is organized into subpackages, named using ISO 3166 country codes.

Most of the *localflavor* add-ons are localized form components deriving from the forms framework – for example, a *USStateField* that knows how to validate U.S. state abbreviations, and a *FISocialSecurityNumber* that knows how to validate Finnish social security numbers.

- Argentina (ar)
- Austria (at)
- Australia (au)
- Belgium (be)
- Bulgaria (bg)
- Brazil (br)
- Canada (ca)
- Switzerland (ch)
- Chile (cl)
- China (cn)
- Colombia (co)
- Czech Republic (cz)
- Germany (de)
- Denmark (dk)
- Ecuador (ec)
- Estonia (ee)
- Spain (es)
- Finland (fi)
- France (fr)
- Great Britain (gb)
- Greece (gr)
- Hong Kong (hk)
- Croatia (hr)
- Hungary (hu)
- Indonesia (id)
- Ireland (ie)
- Israel (il)
- India (in)
- Iceland (is)
- Italy (it)
- Japan (jp)
- Kuwait (kw)
- Lithuania (lt)
- Latvia (lv)
- Macedonia (mk)
- Malta (mt)
- Mexico (mx)
- The Netherlands (nl)
- Norway (no)
- New Zealand (nz)
- Peru (pe)
- Pakistan (pk)
- Poland (pl)
- Portugal (pt)
- Paraguay (py)

- Romania (ro)
- Russia (ru)
- Sweden (se)
- Slovenia (si)
- Slovakia (sk)
- Tunisia (tn)
- Turkey (tr)
- United States of America (us)
- Uruguay (uy)
- South Africa (za)

To use one of these localized components, just import the relevant subpackage. For example, here's how you can create a form with a field representing a Greek postal code:

```
from django import forms
from localflavor.gr.forms import GRPostalCodeField

class MyForm(forms.Form):
    my_greek_postal_code = GRPostalCodeField()
```

The `localflavor` package also includes a `generic` subpackage, containing useful code that is not specific to one particular country or culture. This package defines date, datetime and split datetime input fields based on those from the forms, but with non-US default formats. Here's an example of how to use them:

```
from django import forms
from localflavor import generic

class MyForm(forms.Form):
    my_date_field = generic.forms.DateField()
```

The `localflavor` `generic` package also has IBAN and BIC model and form fields. Here's an example of how to use the IBAN and BIC form fields:

```
from django import forms
from localflavor.generic.forms import BICFormField, IBANFormField

class MyForm(forms.Form):
    iban = IBANFormField()
    bic = BICFormField()
```

Installation

To install django-localflavor use your favorite packaging tool, e.g.pip:

```
pip install django-localflavor
```

Or download the source distribution from PyPI at <https://pypi.python.org/pypi/django-localflavor>, decompress the file and run `python setup.py install` in the unpacked directory.

Then add 'localflavor' to your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # ...  
    'localflavor',  
)
```

Note: Adding 'localflavor' to your `INSTALLED_APPS` setting is required for migrations and translations to work. Using django-localflavor without adding it to your `INSTALLED_APPS` setting is not recommended.

Internationalization

Local flavor has its own catalog of translations, in the directory `localflavor/locale`, and it's not loaded automatically like Django's general catalog in `django/conf/locale`. If you want local flavor's texts to be translated, like form fields error messages, you must include `localflavor` in the `INSTALLED_APPS` setting, so the internationalization system can find the catalog, as explained in [How Django discovers translations](#).

Adding flavors

We'd love to add more of these, so please [create an issue](#) or [pull request](#) with any code you'd like to contribute. See any of the existing flavors for examples.

See the [contributing documentation](#) for how to run the tests while working on a local flavor.

If you consider adding a new localflavor for country here are some examples that you might consider implementing:

- form fields and form widgets
 - ID verification
 - tax or social security number validator
 - car registration
 - postal code validation
 - country area selects, e.g. cities, counties, states, provinces
- model fields, e.g. for storing any of the above form fields' values
- local translations of English area names. Join your language team at Transifex: <https://www.transifex.com/projects/p/django-localflavor/>

Note: `django-localflavor` does not accept contributions of country specific phone number fields. The `django-phonenumber-field` package has excellent support for validating phone numbers in many countries and we recommend this package.

Releases

Due to django-localflavor' history as a former contrib app, the app is required to be working with the actively maintained Django versions. See the documentation about [Django's release process](#) for more information.

django-localflavor releases are not tied to the release cycle of Django. Version numbers follow the appropriate Python standards, e.g. PEPs [386](#) and [440](#).

4.1 Roadmap

django-localflavor releases follow [semver](#). Within one month of django release we would release a new version. We might have an extra release if there are enough features in between django releases.

Version	Django	Date
1.4	1.10	January 2017
1.x
2.0	2.0	January 2018

How to migrate

If you've used the old `django.contrib.localflavor` package or one of the temporary `django-localflavor-*` releases, follow these two easy steps to update your code:

1. Install the third-party `django-localflavor` package.
2. Change your app's import statements to reference the new packages.

For example, change this:

```
from django.contrib.localflavor.gr.forms import GRPostalCodeField
```

...to this:

```
from localflavor.gr.forms import GRPostalCodeField
```

Or if you used one of the shortlived `django-localflavor-*` packages change:

```
from django_localflavor_gr.forms import GRPostalCodeField
```

...to this:

```
from localflavor.gr.forms import GRPostalCodeField
```

The code in the new package is the same (it was copied directly from Django), so you don't have to worry about backwards compatibility in terms of functionality. Only the imports have changed.

Backwards compatibility

We will always attempt to make *localflavor* reflect the officially gazetted policies of the appropriate local government authority. For example, if a government body makes a change to add, alter, or remove a province (or state, or county), that change will be reflected in *localflavor* in the next release.

When a backwards-incompatible change is made (for example, the removal or renaming of a province) the *localflavor* in question will raise a warning when that *localflavor* is imported. This provides a run-time indication that something may require attention.

However, once you have addressed the backwards compatibility (for example, auditing your code to see if any data migration is required), the warning serves no purpose. The warning can then be suppressed. For example, to suppress the warnings raised by the Indonesian *localflavor* you would use the following code:

```
import warnings
warnings.filterwarnings('ignore',
                        category=RuntimeWarning,
                        module='localflavor.id')
from localflavor.id import forms as id_forms
```

Indices and tables

7.1 Authors

- Aaron Boman
- Adam Taylor
- Adnane Belmadiaf
- Adrian Holovaty
- Agustín Scaramuzza
- Alex Butum
- Alex Gaynor
- Alex Hill
- Alex Zhang
- Alix Martineau
- Alonisser
- Andreas Pelme
- Andres Torres Marroquin
- Andrew Godwin
- Aymeric Augustin
- baffolobill
- Baptiste Darthenay
- Ben Davis
- Ben Konrath
- Bruno M. Custódio
- Burhan Khalid
- Claude Paroz
- Daniel Ampuero
- Daniela Ponader

- Danielle Madeley
- Daniel Roschka
- d.merc
- Douglas Miranda
- Erik Romijn
- Flavio Curella
- Florian Apolloner
- François Constant
- Gary Wilson Jr
- Gerardo Orozco
- Ghassen Telmoudi
- Grzes Furga
- Honza Král
- Horst Gutmann
- Ivan Fisun
- Jaap Roes
- Jacob Kaplan-Moss
- James Bennett
- Jannis Leidel
- Jan Pieter Waagmeester
- Jérémie Ferry
- Jocelyn Delalande
- Johnny Lee Othon
- Jonas Ghyllebert
- Joseph Kocherhans
- Josh Crompton
- Julien Phalip
- Justin Bronn
- Karen Tracey
- László Ratskó
- Lefteris Nikoltsios
- Luis Alberto Santana
- Łukasz Langa
- Luke Benstead
- luyikei
- Malcolm Tredinnick

- Martin Ogden
- Marti Raudsepp
- Matias Dinota
- Michał Sałaban
- Mike Lissner
- Olivier Sels
- Paul Donohue
- Paulo Poiati
- Rael Max
- Ramiro Morales
- Rolf Erik Lekang
- Russell Keith-Magee
- Serafeim Papastefanos
- Sergio Oliveira
- Simonas Kazlauskas
- Simon Charette
- Stefan Kjartansson
- tadeo
- Thiago Avelino
- Tino de Bruijn
- Trey Hunner
- Tyler Ball
- Tom Forbes
- Venelin Stoykov
- Vladimir Nani

7.2 Changelog

7.2.1 1.5 (unreleased)

New flavors:

- None

New fields for existing flavors:

- None

Modifications to existing flavors:

- None

Other changes:

- None

7.2.2 1.4 (2017-01-03)

New flavors:

- Added local flavor for Venezuela ([gh-245](#))
- Added local flavor for Morocco ([gh-270](#))

New fields for existing flavors:

- Added MXCLABEField model and form fields. ([gh-227](#)).
- Added AUTaxFileNumberField model and form fields. ([gh-238](#))
- Added KWGovernorateSelect field to easily select Kuwait governorates. ([gh-231](#)).
- Added FRRegion2016Select field to stick to current legislation ([gh-260](#)) and ([gh-268](#)).

Modifications to existing flavors:

- Enhancements of localflavor.br.forms.BRCNPJField ([gh-240](#) [gh-254](#)).
- Fixed century bug with Kuwait Civil ID verification localflavor.kw.forms ([gh-195](#)).
- Allow passing field name as first positional argument of IBANField. ([gh-236](#)).
- Fixed French FRNationalIdentificationNumber bug with imaginary birth month values. ([gh-242](#)).
- Fixed French FRNationalIdentificationNumber bug with corsican people born after 2000. ([gh-242](#)).
- Fixed the translation for US state ‘Georgia’ from colliding with the country ‘Georgia’ ([gh-250](#)).
- Fixed the styling errors and enabled prospector ([gh-259](#)).
- Allow AU ABN value with spaces to validate ([gh-266](#) [gh-267](#)).

Other changes:

- Drop support for Django 1.7 ([gh-218](#)).
- Ensure the migration framework generates schema migrations for model fields that change the `max_length` ([gh-257](#)). Users will need to generate migrations for any model fields they use with ‘makemigrations’.
- Lazily generate `US_STATES`, `STATE_CHOICES`, and `USPS_CHOICES` ([gh-203](#) [gh-272](#)).
- Deprecated Phone Number fields ([gh-262](#)).
- Bumped versions of requirements for testing ([gh-274](#)).

7.2.3 1.3 (2016-05-06)

New flavors:

- Added local flavor for Bulgaria ([gh-191](#))
- Added local flavor for Tunisia ([gh-141](#))
- Added local flavor for Hungary ([gh-213](#))

New fields for existing flavors:

- Added ARCBUField form field. ([gh-151](#)).

- Added NLZipCodeField, NLProvinceField, NLSoFiNumberField, NLPhoneNumberField model fields. ([gh-152](#)).
- Added AUBusinessNumberField model and form fields. ([gh-63](#))

Modifications to existing flavors:

- Moved Dutch validators from localflavor.nl.forms to localflavor.nl.validators ([gh-152](#)).
- Fix check for promotional social security numbers in USSocialSecurityNumberField ([gh-157](#)).
- Updated IBANField to support the latest additions to the IBAN Registry (version 64 / March 2016).
- Fix bug with MXRFCField where some incorrect values would validate correctly. ([gh-204](#)).
- Fixed bug with IBANFormField validation. ([gh-215](#)).
- Update regex in DEZipCodeField to prohibit invalid postal codes. ([gh-216](#)).
- Added deconstructor methods to validators. ([gh-220](#)).
- Fix bug in ESIdentityCardNumberField where some valid values for NIE numbers were not validating. ([gh-217](#))
- Add deconstruct method to all model fields ([gh-162](#) [gh-224](#)).

Other changes:

- Drop support for Django 1.5, Django 1.6 and Python 2.6 ([gh-170](#)).

7.2.4 1.2 (2015-11-27)

New flavors:

- None

New fields for existing flavors:

- Added form field for Estonian business registration codes ([gh-135](#)).
- Added model field for Ecuadorian provinces ([gh-138](#)).
- Added form field for Swiss Social Security numbers (([gh-155](#)).
- Added form field for Brazilian Legal Process numbers (Processos) ([gh-163](#)).

Modifications to existing flavors:

- Fixed misspelled Polish administrative unit names ([gh-136](#)).
- Added Kosovo and Timor-Leste to list of IBAN countries ([gh-139](#)).
- Fixed error in Romanian fiscal identity code (CIF) field when value has a trailing slash ([gh-146](#)).
- Updated validation in Swiss postal code field to only accept values in the range 1000 - 9000 ([gh-154](#)).
- Added validator for International Article Number (EAN) to the generic module ([gh-156](#)).
- Updated Italian social security number field to use 'tax code' in error message ([gh-167](#)).
- Fixed error in Greek tax number code field when value has only alpha characters ([gh-171](#)).
- Added stricter validation in the Brazilian Cadastro de Pessoas Físicas (CPF) field ([gh-172](#)).
- Corrected Romanian counties choice names to use and (comma below) ([gh-175](#)).
- Updated Brazilian postal code field to also accept values with XX.XXX-XXX and XXXXXXXXX formats ([gh-177](#)).
- Marked US state names for translation ([gh-178](#)).

- Fixed French national identification number validation for people born before 1976 in Corsica ([gh-186](#)).

7.2.5 1.1 (2014-12-10)

New flavors:

- Added local flavor for Denmark ([gh-83](#))
- Added local flavor for Estonia ([gh-70](#))
- Added local flavor for Latvia ([gh-68](#))
- Added local flavor for Malta ([gh-88](#))
- Added local flavor for Pakistan ([gh-41](#))
- Added local flavor for Singapore ([gh-119](#))

New fields for existing flavors:

- Added model and form fields for French SIREN/SIRET numbers ([gh-123](#))
- Added model field for states of Brazil ([gh-22](#))
- Added form field for Indian Aadhaar numbers ([gh-23](#))
- Added model field for states of India ([gh-23](#))
- Added form field for Lithuanian phone numbers
- Added model field for Dutch bank accounts ([gh-42](#))
- Added form field for Italian phone numbers ([gh-74](#))
- Added form field for French National Identification Number ([gh-75](#))
- Added IBAN model and form fields ([gh-86](#))
- Added BIC model and form fields ([gh-125](#))
- Added SSN model field for US ([gh-96](#))
- Added ZIP code model field for US ([gh-55](#))

Other modifications to existing flavors:

- *backward incompatible* Updated the region lists of Great Britain ([gh-43](#), [gh-126](#))
- Added Ceuta and Mellila to regions of Spain ([gh-8](#))
- Added support entities in Italian SSN form field ([gh-20](#))
- Added Japanese prefecture codes and fix prefecture order ([gh-27](#))
- Added normalization for Lithuanian postal code field ([gh-69](#))
- Added whitespace stripping whitespace from US ZIP code field ([gh-77](#))
- Added an option for customizing French form field labels ([gh-102](#))
- Added mapping between provinces and regions for Italy ([gh-105](#))
- Added Telengana to states of India ([gh-107](#))
- Added support for 14X and 17X Chinese cell numbers ([gh-17](#), [gh-120](#))
- Allowed spaces in CPF numbers for Brazil ([gh-32](#))
- Fixed CIF validation for Spain ([gh-78](#))

- Fixed armed forces “states” for US (gh-8)
- Fixed REGON number validation for Poland (gh-62)
- Rejected US SSN starting with 9 (gh-35)
- Rejected Brazilian CPF number when all numbers all numbers are equal (gh-103)
- Added ‘Y’ to the NIE number validation for Spain (gh-127)
- Updated Argentina’s CUIT number validation to support legal types 24 and 33 (gh-121)
- Added ‘R’, ‘V’ and ‘W’ to the Spanish identity card number validation (gh-132)

Other changes:

- Added checksums module (from Django) providing a Luhn validator (gh-122)

7.2.6 1.0 (2013-07-29)

Initial release

- genindex
- modindex
- search

7.3 Argentina (ar)

AR-specific Form helpers.

class `localflavor.ar.forms.ARCBUField` (**args*, ***kwargs*)

This field validates a CBU (Clave Bancaria Uniforme).

A CBU is a 22-digits long number. The first 8 digits denote bank and branch number, plus a verifying digit. The remaining 14 digits denote an account number, plus a verifying digit.

More info: https://es.wikipedia.org/wiki/Clave_Bancaria_Uniforme

New in version 1.3.

clean (*value*)

Value must be a 22 digits long number.

class `localflavor.ar.forms.ARCUITField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

This field validates a CUIT (Código Único de Identificación Tributaria).

ACUIT is of the form XX-XXXXXXXX-V. The last digit is a check digit.

More info: http://es.wikipedia.org/wiki/Clave_%C3%9Anica_de_Identificaci%C3%B3n_Tributaria

English info: <http://www.justlanded.com/english/Argentina/Argentina-Guide/Visas-Permits/Other-Legal-Documents>

clean (*value*)

Value can be either a string in the format XX-XXXXXXXX-X or an 11-digit number.

class `localflavor.ar.forms.ARDNIField` (*max_length=10*, *min_length=7*, **args*, ***kwargs*)

A field that validates ‘Documento Nacional de Identidad’ (DNI) numbers.

clean (*value*)

Value can be a string either in the [X]X.XXX.XXX or [X]XXXXXXXXX formats.

class `localflavor.ar.forms.ARPostalCodeField` (*max_length=8, min_length=4, *args, **kwargs*)

A field that accepts a ‘classic’ NNNN Postal Code or a CPA.

See: http://www.correoargentino.com.ar/cpa/que_es http://www.correoargentino.com.ar/cpa/como_escribirlo

class `localflavor.ar.forms.ARProvinceSelect` (*attrs=None*)

A Select widget that uses a list of Argentinean provinces/autonomous cities as its choices.

`localflavor.ar.ar_provinces.PROVINCE_CHOICES = (('B', 'Buenos Aires'), ('K', 'Catamarca'), ('H', 'Chaco'), ('U', 'Chubut'), ('C', 'Cordoba'), ('E', 'Entre Rios'), ('M', 'Mendoza'), ('N', 'Misiones'), ('P', 'Parana'), ('R', 'Rio Negro'), ('S', 'Salta'), ('T', 'Tucuman'), ('G', 'Guayaquil'), ('D', 'Distrito Federal'))`
 A list of Argentinean provinces and autonomous cities as *choices* in a formfield. From <http://www.argentina.gov.ar/argentina/portal/paginas.dhtml?pagina=425>

7.4 Austria (at)

AT-specific Form helpers.

class `localflavor.at.forms.ATSocialSecurityNumberField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

Austrian Social Security numbers are composed of a 4 digits and 6 digits field.

The latter represents in most cases the person’s birthdate while the first 4 digits represent a 3-digits counter and a one-digit checksum.

The 6-digits field can also differ from the person’s birthdate if the 3-digits counter suffered an overflow.

This code is based on information available on <http://de.wikipedia.org/wiki/Sozialversicherungsnummer#.C3.96sterreich>

class `localflavor.at.forms.ATStateSelect` (*attrs=None*)

A Select widget that uses a list of AT states as its choices.

class `localflavor.at.forms.ATZipCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates its input is an Austrian postcode.

Accepts 4 digits (first digit must be greater than 0).

`localflavor.at.at_states.STATE_CHOICES = (('BL', 'Burgenland'), ('KA', 'Carinthia'), ('NO', 'Lower Austria'), ('U', 'Upper Austria'), ('V', 'Vorarlberg'), ('W', 'Wien'))`
 A list of Austrian states according to https://en.wikipedia.org/wiki/States_of_Austria

7.5 Australia (au)

7.5.1 Forms

Australian-specific Form helpers.

class `localflavor.au.forms.AUBusinessNumberField` (*max_length=None, min_length=None, strip=True, *args, **kwargs*)

A form field that validates input as an Australian Business Number (ABN).

New in version 1.3.

Changed in version 1.4.

prepare_value (*value*)

Format the value for display.

class `localflavor.au.forms.AUPhoneNumberField` (*max_length=None, min_length=None, strip=True, *args, **kwargs*)

A form field that validates input as an Australian phone number.

Valid numbers have ten digits.

clean (*value*)

Validate a phone number. Strips parentheses, whitespace and hyphens.

class `localflavor.au.forms.AUPostCodeField` (*max_length=4, min_length=None, *args, **kwargs*)

Australian post code field.

Assumed to be 4 digits. Northern Territory 3-digit postcodes should have leading zero.

class `localflavor.au.forms.AUStateSelect` (*attrs=None*)

A Select widget that uses a list of Australian states/territories as its choices.

class `localflavor.au.forms.AUTaxFileNumberField` (*max_length=None, min_length=None, strip=True, *args, **kwargs*)

A form field that validates input as an Australian Tax File Number (TFN).

New in version 1.4.

prepare_value (*value*)

Format the value for display.

7.5.2 Models

class `localflavor.au.models.AUBusinessNumberField` (**args, **kwargs*)

A model field that checks that the value is a valid Australian Business Number (ABN).

New in version 1.3.

description = 'Australian Business Number'

formfield (***kwargs*)

to_python (*value*)

Ensure the ABN is stored without spaces.

validators = [`localflavor.au.validators.AUBusinessNumberFieldValidator` object at `0x7f0fc3180160`], `<django.core.validators.ABNCValidator object at 0x7f0fc3180160>`

class `localflavor.au.models.AUPhoneNumberField` (**args, **kwargs*)

A model field that checks that the value is a valid Australian phone number (ten digits).

description = 'Australian Phone number'

formfield (***kwargs*)

class `localflavor.au.models.AUPostCodeField` (**args, **kwargs*)

A model field that stores the four-digit Australian postcode in the database.

This field is represented by forms as a `AUPostCodeField` field.

description = 'Australian Postcode'

formfield (***kwargs*)

class `localflavor.au.models.AUStateField(*args, **kwargs)`
 A model field that stores the three-letter Australian state abbreviation in the database.
 It is represented with `STATE_CHOICES` `choices`.

deconstruct ()

description = 'Australian State'

class `localflavor.au.models.AUTaxFileNumberField(*args, **kwargs)`
 A model field that checks that the value is a valid Tax File Number (TFN).

A TFN is a number issued to a person by the Commissioner of Taxation and is used to verify client identity and establish their income levels. It is a eight or nine digit number without any embedded meaning.

New in version 1.4.

description = 'Australian Tax File Number'

formfield (**kwargs)

to_python (value)

Ensure the TFN is stored without spaces.

validators = [`localflavor.au.validators.AUTaxFileNumberFieldValidator` object at `0x7f0fc3110160`], `<django.core.val`

7.5.3 Data

`localflavor.au.au_states.STATE_CHOICES` = (('ACT', 'Australian Capital Territory'), ('NSW', 'New South Wales'),
 An alphabetical list of states for use as *choices* in a formfield.

7.6 Belgium (be)

7.6.1 Forms

Belgium-specific Form helpers.

class `localflavor.be.forms.BEPhoneNumberField(max_length=None, min_length=None, *args, **kwargs)`

A form field that validates its input as a belgium phone number.

Landlines have a seven-digit subscriber number and a one-digit area code, while smaller cities have a six-digit subscriber number and a two-digit area code. Cell phones have a six-digit subscriber number and a two-digit area code preceeded by the number 4. `0d ddd dd dd`, `0d/ddd.dd.dd`, `0d.ddd.dd.dd`, `0ddddddd` - dialling a bigger city
`0dd dd dd dd`, `0dd/dd.dd.dd`, `0dd.dd.dd.dd`, `0ddddddd` - dialling a smaller city
`04dd ddd dd dd`, `04dd/ddd.dd.dd`, `04dd.ddd.dd.dd`, `04dddddddd` - dialling a mobile number

class `localflavor.be.forms.BEPostalCodeField(max_length=None, min_length=None, *args, **kwargs)`

A form field that validates its input as a belgium postal code.

Belgium postal code is a 4 digits string. The first digit indicates the province (except for the 3ddd numbers that are shared by the eastern part of Flemish Brabant and Limburg and the and 1ddd that are shared by the Brussels Capital Region, the western part of Flemish Brabant and Walloon Brabant)

class `localflavor.be.forms.BEProvinceSelect(attrs=None)`

A Select widget that uses a list of belgium provinces as its choices.

class `localflavor.be.forms.BERegionSelect` (*attrs=None*)
 A Select widget that uses a list of belgium regions as its choices.

7.6.2 Data

`localflavor.be.be_provinces.PROVINCE_CHOICES` = (('VAN', 'Antwerp'), ('BRU', 'Brussels'), ('VOV', 'East Flandre'))
 ISO codes

`localflavor.be.be_regions.REGION_CHOICES` = (('BRU', 'Brussels Capital Region'), ('VLG', 'Flemish Region'), ('WAL', 'Walloon Region'))
 ISO codes

7.7 Bulgaria (bg)

7.7.1 Validators

class `localflavor.bg.validators.EGNValidator`
 Check Bulgarian unique citizenship number (EGN) for validity.

More details https://en.wikipedia.org/wiki/Unique_citizenship_number Full information in Bulgarian about algorithm is available here <http://www.grao.bg/esgraon.html#section2>

deconstruct (*obj*)
 Returns a 3-tuple of class import path, positional arguments, and keyword arguments.

class `localflavor.bg.validators.EIKValidator`
 Check Bulgarian EIK/BULSTAT codes for validity.

Full information in Bulgarian about algorithm is available here <http://bulstat.registryagency.bg/About.html>

deconstruct (*obj*)
 Returns a 3-tuple of class import path, positional arguments, and keyword arguments.

7.7.2 Model Fields

class `localflavor.bg.models.BGEGNField` (**args, **kwargs*)
 Field that stores Bulgarian unique citizenship number (EGN).

This is shortcut for:

```
models.CharField(max_length=10, validators=[localflavor.bg.validators.egn_validator])
```

class `localflavor.bg.models.BGEIKField` (**args, **kwargs*)
 Field that stores Bulgarian EIK/BULSTAT codes.

This is shortcut for:

```
models.CharField(max_length=13, validators=[localflavor.bg.validators.eik_validator])
```

7.7.3 Utils

`localflavor.bg.utils.get_egn_birth_date` (*egn*)
 Extract birth date from Bulgarian unique citizenship number (EGN).

More details https://en.wikipedia.org/wiki/Unique_citizenship_number Information in Bulgarian for this can be found here <http://www.grao.bg/esgraon.html#section2>

7.8 Brazil (br)

7.8.1 Forms

BR-specific Form helpers.

class `localflavor.br.forms.BRCNPJField` (*min_length=14, max_length=18, *args, **kwargs*)
 A form field that validates input as [Brazilian CNPJ](#).

Input can either be of the format `XX.XXX.XXX/XXXX-XX` or be a group of 14 digits.

If you want to use the long format only, you can specify: `brcnpj_field = BRCNPJField(min_length=16)`

If you want to use the short format, you can specify: `brcnpj_field = BRCNPJField(max_length=14)`

Otherwise both formats will be valid.

Changed in version 1.4.

clean (*value*)

Value can be either a string in the format `XX.XXX.XXX/XXXX-XX` or a group of 14 characters.

class `localflavor.br.forms.BRCPFField` (*max_length=14, min_length=11, *args, **kwargs*)
 A form field that validates a CPF number or a CPF string.

A CPF number is compounded by `XXX.XXX.XXX-VD`. The two last digits are check digits.

More information: http://en.wikipedia.org/wiki/Cadastro_de_Pessoas_F%C3%ADsicas

clean (*value*)

Value can be either a string in the format `XXX.XXX.XXX-XX` or an 11-digit number.

class `localflavor.br.forms.BRPhoneNumberField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

A form field that validates input as a Brazilian phone number.

The phone number must be in either of the following formats: `XX-XXXX-XXXX` or `XX-XXXXXX-XXXX`.

class `localflavor.br.forms.BRProcessoField` (*max_length=25, min_length=20, *args, **kwargs*)

A form field that validates a Legal Process(Processo) number or a Legal Process string.

A Processo number is compounded by `NNNNNNN-DD.AAAA.J.TR.OOOO`. The two DD digits are check digits. More information: <http://www.cnj.jus.br/busca-atos-adm?documento=2748>

New in version 1.2.

clean (*value*)

Value can be either a string in the format `NNNNNNN-DD.AAAA.J.TR.OOOO` or an 20-digit number.

class `localflavor.br.forms.BRStateChoiceField` (*required=True, widget=None, label=None, initial=None, help_text=None*)

A choice field that uses a list of Brazilian states as its choices.

class `localflavor.br.forms.BRStateSelect` (*attrs=None*)

A Select widget that uses a list of Brazilian states/territories as its choices.

class `localflavor.br.forms.BRZipCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates input as a Brazilian zip code, with the format `XXXXX-XXX`.

7.8.2 Data

`localflavor.br.br_states.STATE_CHOICES = (('AC', 'Acre'), ('AL', 'Alagoas'), ('AP', 'Amapá'), ('AM', 'Amazonas'))`
 An alphabetical list of Brazilian states for use as *choices* in a formfield

7.9 Canada (ca)

7.9.1 Forms

Canada-specific Form helpers.

```
class localflavor.ca.forms.CAPhoneNumberField(required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)
```

Canadian phone number form field.

```
class localflavor.ca.forms.CAPostalCodeField(max_length=None, min_length=None, strip=True, *args, **kwargs)
```

Canadian postal code form field.

Validates against known invalid characters: D, F, I, O, Q, U Additionally the first character cannot be Z or W. For more info see: <http://www.canadapost.ca/tools/pg/manual/PGaddress-e.asp#1402170>

```
class localflavor.ca.forms.CAProvinceField(required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)
```

A form field that validates its input is a Canadian province name or abbreviation.

It normalizes the input to the standard two-letter postal service abbreviation for the given province.

```
class localflavor.ca.forms.CAProvinceSelect(attrs=None)
```

A Select widget that uses a list of Canadian provinces and territories as its choices.

```
class localflavor.ca.forms.CASocialInsuranceNumberField(required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)
```

A Canadian Social Insurance Number (SIN).

Checks the following rules to determine whether the number is valid:

- Conforms to the XXX-XXX-XXX format.
- Passes the check digit process “Luhn Algorithm” See: http://en.wikipedia.org/wiki/Social_Insurance_Number

7.9.2 Data

`localflavor.ca.ca_provinces.PROVINCE_CHOICES = (('AB', 'Alberta'), ('BC', 'British Columbia'), ('MB', 'Manitoba'))`
 An alphabetical list of provinces and territories for use as *choices* in a formfield. Source:

http://www.canada.gc.ca/othergov/prov_e.html

`localflavor.ca.ca_provinces.PROVINCES_NORMALIZED = {'quebec': 'QC', 'alberta': 'AB', 'pei': 'PE', 'yukon terr`
a mapping of province misspellings/abbreviations to normalized abbreviations

7.10 Switzerland (ch)

7.10.1 Forms

Swiss-specific Form helpers.

```
class localflavor.ch.forms.CHIdentityCardNumberField(required=True, widget=
get=None, label=None, ini-
tial=None, help_text='',
error_messages=None,
show_hidden_initial=False,
validators=[], localize=False, dis-
abled=False, label_suffix=None)
```

A Swiss identity card number.

Checks the following rules to determine whether the number is valid:

- Conforms to the X1234567<0 or 1234567890 format.
- Included checksums match calculated checksums

```
class localflavor.ch.forms.CHPhoneNumberField(required=True, widget=None,
label=None, initial=None,
help_text='', error_messages=None,
show_hidden_initial=False, valida-
tors=[], localize=False, disabled=False,
label_suffix=None)
```

Validate local Swiss phone number (not international ones).

The correct format is '0XX XXX XX XX'. '0XX.XXX.XX.XX' and '0XXXXXXXXXX' validate but are corrected to '0XX XXX XX XX'.

```
class localflavor.ch.forms.CHSocialSecurityNumberField(max_length=None,
min_length=None, strip=True,
*args, **kwargs)
```

A Swiss Social Security number (also known as the new AHV Number).

Checks the following rules to determine whether the number is valid:

- Conforms to the 756.XXXX.XXXX.XX
- Included checksums match calculated checksums

See: <http://de.wikipedia.org/wiki/Sozialversicherungsnummer#Versichertenummer>

New in version 1.2.

```
class localflavor.ch.forms.CHStateSelect(attrs=None)
A Select widget that uses a list of CH states as its choices.
```

```
class localflavor.ch.forms.CHZipCodeField(max_length=None, min_length=None, *args,
**kwargs)
```

A form field that validates input as a Swiss zip code.

Valid codes consist of four digits ranging from 1XXX to 9XXX.

See: http://en.wikipedia.org/wiki/Postal_codes_in_Switzerland_and_Liechtenstein

7.10.2 Data

`localflavor.ch.ch_states.STATE_CHOICES` = (('AG', 'Aargau'), ('AI', 'Appenzell Innerrhoden'), ('AR', 'Appenzell A'))
An alphabetical list of states

7.11 Chile (cl)

7.11.1 Forms

Chile specific form helpers.

class `localflavor.cl.forms.CLRegionSelect` (*attrs=None*)
A Select widget that uses a list of Chilean Regions (Regiones) as its choices.

class `localflavor.cl.forms.CLRutField` (**args, **kwargs*)
Chilean “Rol Unico Tributario” (RUT) field.

This is the Chilean national identification number.

Samples for testing are available from https://palena.sii.cl/cvc/dte/ee_empresas_emisoras.html

clean (*value*)
Check and clean the Chilean RUT.

7.11.2 Data

`localflavor.cl.cl_regions.REGION_CHOICES` = (('RM', 'Región Metropolitana de Santiago'), ('T', 'Región de Tarapacá'))
A list of Chilean regions as *choices* in a formfield.

7.12 China (cn)

7.12.1 Forms

China(mainland)-specific Form helpers.

class `localflavor.cn.forms.CNProvinceSelect` (*attrs=None*)
A select widget providing the list of provinces and districts in People’s Republic of China as choices.

class `localflavor.cn.forms.CNPostCodeField` (**args, **kwargs*)
A form field that validates input as postal codes in mainland China.

Valid codes are in the format of XXXXXX where X is a digit.

class `localflavor.cn.forms.CNIDCardField` (*max_length=18, min_length=15, *args, **kwargs*)
A form field that validates input as a Resident Identity Card (PRC) number.

This field would check the following restrictions:

- the length could only be 15 or 18;
- if the length is 18, the last character can be x or X;
- has a valid checksum (only for those with a length of 18);

- has a valid date of birth;
- has a valid province.

The checksum algorithm is described in GB11643-1999. See: http://en.wikipedia.org/wiki/Resident_Identity_Card#Identity_card

clean (*value*)

Check whether the input is a valid ID Card Number.

has_valid_birthday (*value*)

This method grabs the date of birth from the ID card number and test whether it is a valid date.

has_valid_checksum (*value*)

This method checks if the last letter/digit is valid according to GB11643-1999.

has_valid_location (*value*)

This method checks if the first two digits in the ID Card are valid province code.

class `localflavor.cn.forms.CNPhoneNumberField` (**args, **kwargs*)

A form field that validates input as a telephone number in mainland China.

A valid phone number could be like: 010-12345678. Considering there might be extension numbers, this could also be: 010-12345678-35.

class `localflavor.cn.forms.CNCellNumberField` (**args, **kwargs*)

A form field that validates input as a cellphone number in mainland China.

A valid cellphone number could be like: 13012345678.

A very rough rule is used here: the first digit should be 1, the second should be 3, 4, 5, 7 or 8, followed by 9 more digits. The total length of a cellphone number should be 11.

Changed in version 1.1: Added 7 as a valid second digit for Chinese virtual mobile ISPs.

7.12.2 Data

`localflavor.cn.cn_provinces.CN_PROVINCE_CHOICES` = (('anhui', ''), ('beijing', ''), ('chongqing', ''), ('fujian', ''), ('

An alphabetical list of provinces for use as *choices* in a formfield. http://en.wikipedia.org/wiki/ISO_3166-2:CN http://en.wikipedia.org/wiki/Province_%28China%29 http://en.wikipedia.org/wiki/Direct-controlled_municipality http://en.wikipedia.org/wiki/Autonomous_regions_of_China

7.13 Colombia (co)

7.13.1 Forms

Colombian-specific form helpers.

class `localflavor.co.forms.CODEpartmentSelect` (*attrs=None*)

A Select widget that uses a list of Colombian states as its choices.

7.13.2 Data

`localflavor.co.co_departments.DEPARTMENT_CHOICES` = (('AMA', 'Amazonas'), ('ANT', 'Antioquia'), ('ARA', 'A

A list of Colombian departaments as *choices* in a formfield.

7.14 Czech Republic (cz)

7.14.1 Forms

Czech-specific form helpers.

```
class localflavor.cz.forms.CZBirthNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text='', error_messages=None,
                                              show_hidden_initial=False, valida-
                                              tors=[], localize=False, disabled=False,
                                              label_suffix=None)
```

Czech birth number form field.

```
class localflavor.cz.forms.CZICNumberField(required=True, widget=None, label=None, ini-
                                           tial=None, help_text='', error_messages=None,
                                           show_hidden_initial=False, validators=[], local-
                                           ize=False, disabled=False, label_suffix=None)
```

Czech IC number form field.

```
class localflavor.cz.forms.CZPostalCodeField(max_length=None, min_length=None, *args,
                                             **kwargs)
```

A form field that validates its input as Czech postal code.

Valid form is XXXXX or XXX XX, where X represents integer.

clean (*value*)

Validates the input and returns a string that contains only numbers.

Returns an empty string for empty values.

```
class localflavor.cz.forms.CZRegionSelect(attrs=None)
```

A select widget widget with list of Czech regions as choices.

7.14.2 Data

```
localflavor.cz.cz_regions.REGION_CHOICES = (('PR', 'Prague'), ('CE', 'Central Bohemian Region'), ('SO', 'South  
Czech regions, translations get from http://www.crwflags.com/fotw/Flags/cz-re.html)
```

7.15 Germany (de)

7.15.1 Forms

DE-specific Form helpers.

```
class localflavor.de.forms.DEIdentityCardNumberField(required=True, wid-
                                                       get=None, label=None, ini-
                                                       tial=None, help_text='',
                                                       error_messages=None,
                                                       show_hidden_initial=False,
                                                       validators=[], localize=False, dis-
                                                       abled=False, label_suffix=None)
```

A German identity card number.

Checks the following rules to determine whether the number is valid:

- Conforms to the XXXXXXXXXXX-XXXXXXX-XXXXXXX-X format.
- No group consists entirely of zeroes.
- Included checksums match calculated checksums

Algorithm is documented at <http://de.wikipedia.org/wiki/Personalausweis>

class `localflavor.de.forms.DEStateSelect` (*attrs=None*)
 A Select widget that uses a list of DE states as its choices.

class `localflavor.de.forms.DEZipCodeField` (*max_length=None, min_length=None, *args, **kwargs*)
 A form field that validates input as a German zip code.
 Valid zip codes consist of five digits.

7.15.2 Data

`localflavor.de.de_states.STATE_CHOICES` = (('BW', 'Baden-Wuerttemberg'), ('BY', 'Bavaria'), ('BE', 'Berlin'), ('B
 An alphabetical list of states

7.16 Denmark (dk)

New in version 1.1.

7.16.1 Forms

Denmark specific Form helpers.

class `localflavor.dk.forms.DKMunicipalitySelect` (*attrs=None, *args, **kwargs*)
 A Select widget that uses a list of Danish municipalities (kommuner) as its choices.

class `localflavor.dk.forms.DKPhoneNumberField` (*max_length=None, min_length=None, *args, **kwargs*)
 Field with phone number validation.

Requires a phone number with 8 digits and optional country code.

class `localflavor.dk.forms.DKPostalCodeField` (*max_length=None, min_length=None, strip=True, *args, **kwargs*)
 An Input widget that uses a list of Danish postal codes as valid input.

7.16.2 Data

`localflavor.dk.dk_postalcodes.DK_POSTALCODES` = (('0555', 'Scanning'), ('0800', 'Høje Taastrup'), ('0877', 'Køber
 A list of Danish postal codes as *choices* in a formfield.

`localflavor.dk.dk_municipalities.REGION_HOVEDSTADEN` = [('albertslund', 'Albertslund'), ('alleroed', 'Allerød')]
 A list of municipalities in the Danish region Hovedstaden as *choices* in a formfield.

`localflavor.dk.dk_municipalities.REGION_MIDTJYLLAND` = [('favrskov', 'Favrskov'), ('hedensted', 'Hedensted')]
 A list of municipalities in the Danish region Midtjylland as *choices* in a formfield.

`localflavor.dk.dk_municipalities.REGION_NORDJYLLAND` = [('broenderslev', 'Brønderslev'), ('frederikshavn', 'F
 A list of municipalities in the Danish region Nordjylland as *choices* in a formfield.

static ee_checksum (*value*)

Takes a string of digits as input, returns check digit.

class `localflavor.ee.forms.EEZipCodeField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates input as a Estonian zip code.

Valid codes consist of five digits; first digit cannot be 0.

7.18.2 Data

`localflavor.ee.ee_counties.COUNTY_CHOICES = (('37', 'Harju County'), ('39', 'Hiiumaa County'), ('44', 'Ida-Viru County'))`

A list of Estonian counties as *choices* in a formfield. Identifiers based on ISO 3166-2:EE.

https://en.wikipedia.org/wiki/ISO_3166-2:EE

7.19 Spain (es)

7.19.1 Forms

Spanish-specific Form helpers.

class `localflavor.es.forms.ESCCCField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates its input as a Spanish bank account or CCC (Codigo Cuenta Cliente).

Spanish CCC is in format EEEE-OOOO-CC-AAAAAAAAAAA where:

E = entity O = office C = checksum A = account

It's also valid to use a space as delimiter, or to use no delimiter.

First checksum digit validates entity and office, and last one validates account. Validation is done multiplying every digit of 10 digit value (with leading 0 if necessary) by number in its position in string 1, 2, 4, 8, 5, 10, 9, 7, 3, 6. Sum resulting numbers and extract it from 11. Result is checksum except when 10 then is 1, or when 11 then is 0.

TODO: allow IBAN validation too

class `localflavor.es.forms.ESIdentityCardNumberField` (*only_nif=False*, *max_length=None*, *min_length=None*, **args*, ***kwargs*)

Spanish NIF/NIE/CIF (Fiscal Identification Number) code.

Validates three different formats:

NIF (individuals): 12345678A CIF (companies): A12345678 NIE (foreigners): X12345678A

according to a couple of simple checksum algorithms.

Value can include a space or hyphen separator between number and letters. Number length is not checked for NIF (or NIE), old values start with a 1, and future values can contain digits greater than 8. The CIF control digit can be a number or a letter depending on company type. Algorithm is not public, and different authors have different opinions on which ones allows letters, so both validations are assumed true for all types.

http://es.wikipedia.org/wiki/N%C3%BAmero_de_identificaci%C3%B3n_fiscal

Changed in version 1.1.

class `localflavor.es.forms.ESPhoneNumberField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates its input as a Spanish phone number.

Information numbers are omitted.

Spanish phone numbers are nine digit numbers, where first digit is 6 (for cell phones), 8 (for special phones), or 9 (for landlines and special phones)

TODO: accept and strip characters like dot, hyphen... in phone number

class `localflavor.es.forms.ESPostalCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates its input as a spanish postal code.

Spanish postal code is a five digits string, with two first digits between 01 and 52, assigned to provinces code.

class `localflavor.es.forms.ESProvinceSelect` (*attrs=None*)

A Select widget that uses a list of spanish provinces as its choices.

class `localflavor.es.forms.ESRegionSelect` (*attrs=None*)

A Select widget that uses a list of spanish regions as its choices.

7.19.2 Data

`localflavor.es.es_provinces.PROVINCE_CHOICES` = (('01', 'Araba'), ('02', 'Albacete'), ('03', 'Alacant'), ('04', 'Alm
An alphabetical list of provinces

`localflavor.es.es_regions.REGION_CHOICES` = (('AN', 'Andalusia'), ('AR', 'Aragon'), ('O', 'Principality of Asturias
An alphabetical list of regions

7.20 Finland (fi)

7.20.1 Forms

FI-specific Form helpers.

class `localflavor.fi.forms.FIMunicipalitySelect` (*attrs=None*)

A Select widget that uses a list of Finnish municipalities as its choices.

class `localflavor.fi.forms.FISocialSecurityNumber` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

A form field that validates input as a Finnish social security number.

class `localflavor.fi.forms.FIZipCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates input as a Finnish zip code.

Valid codes consist of five digits.

It's under authority of the INSEE. See http://fr.wikipedia.org/wiki/Système_d'identification_du_répertoire_des_établissements for more information.

New in version 1.1.

class `localflavor.fr.forms.FRZipCodeField` (**args*, ***kwargs*)
 Validate local French zip code.

The correct format is 'XXXXX'.

7.21.2 Data

`localflavor.fr.fr_department.DEPARTMENT_CHOICES_PER_REGION` = (('01', 'Ain', '82'), ('02', 'Aisne', '22'), ('03', 'Allier', '12'))
 See the "Code officiel géographique" on the INSEE website <www.insee.fr>.

`localflavor.fr.fr_department.DEPARTMENT_CHOICES` = (('01', '01 - Ain'), ('02', '02 - Aisne'), ('03', '03 - Allier'), ('04', '04 - Ardennes'), ('05', '05 - Aube'), ('06', '06 - Alpes de Hautes Provence'), ('07', '07 - Ardèche'), ('08', '08 - Aisne'), ('09', '09 - Ariège'), ('10', '10 - Aube'), ('11', '11 - Calvados'), ('12', '12 - Allier'), ('13', '13 - Bouches du Rhône'), ('14', '14 - Calvados'), ('15', '15 - Cantal'), ('16', '16 - Charente maritime'), ('17', '17 - Charente'), ('18', '18 - Cher'), ('19', '19 - Corrèze'), ('20', '20 - Corse du Sud'), ('21', '21 - Côte d'Or'), ('22', '22 - Aisne'), ('23', '23 - Creuse'), ('24', '24 - Dordogne'), ('25', '25 - Doubs'), ('26', '26 - Drôme'), ('27', '27 - Eure'), ('28', '28 - Eure-et-Loire'), ('29', '29 - Finistère'), ('30', '30 - Gard'), ('31', '31 - Haute-Garonne'), ('32', '32 - Gers'), ('33', '33 - Gironde'), ('34', '34 - Hérault'), ('35', '35 - Ille-et-Vilaine'), ('36', '36 - Indre-et-Loire'), ('37', '37 - Maine-et-Loire'), ('38', '38 - Isère'), ('39', '39 - Jura'), ('40', '40 - Landes'), ('41', '41 - Loir-et-Cher'), ('42', '42 - Loire'), ('43', '43 - Haute-Loire'), ('44', '44 - Loire-Atlantique'), ('45', '45 - Loiret'), ('46', '46 - Lot'), ('47', '47 - Lot-et-Garonne'), ('48', '48 - Lozère'), ('49', '49 - Mayenne'), ('50', '50 - Manche'), ('51', '51 - Marne'), ('52', '52 - Haute-Marne'), ('53', '53 - Mayenne'), ('54', '54 - Meurthe-et-Moselle'), ('55', '55 - Meuse'), ('56', '56 - Morbihan'), ('57', '57 - Moselle'), ('58', '58 - Nièvre'), ('59', '59 - Nord'), ('60', '60 - Oise'), ('61', '61 - Mayenne'), ('62', '62 - Nord-Pas de Calais'), ('63', '63 - Puy-de-Dôme'), ('64', '64 - Pyrénées-Atlantiques'), ('65', '65 - Hautes-Pyrénées'), ('66', '66 - Pyrénées-Orientales'), ('67', '67 - Bas-Rhin'), ('68', '68 - Haut-Rhin'), ('69', '69 - Rhône'), ('70', '70 - Vosges'), ('71', '71 - Saône-et-Loire'), ('72', '72 - Mayenne'), ('73', '73 - Savoie'), ('74', '74 - Haute-Savoie'), ('75', '75 - Paris'), ('76', '76 - Seine-Maritime'), ('77', '77 - Seine-et-Marne'), ('78', '78 - Yvelines'), ('79', '79 - Deux-Sèvres'), ('80', '80 - Somme'), ('81', '81 - Tarn'), ('82', '82 - Tarn-et-Garonne'), ('83', '83 - Var'), ('84', '84 - Vaucluse'), ('85', '85 - Vendée'), ('86', '86 - Deux-Sèvres'), ('87', '87 - Dordogne'), ('88', '88 - Vosges'), ('89', '89 - Aube'), ('90', '90 - Territoire de Belfort'), ('91', '91 - Essonne'), ('92', '92 - Hauts-de-Seine'), ('93', '93 - Seine-Saint-Denis'), ('94', '94 - Val-de-Marne'), ('95', '95 - Val d'Oise'))
 A list of departments

`localflavor.fr.fr_region.REGION_CHOICES` = (('01', 'Guadeloupe'), ('02', 'Martinique'), ('03', 'Guyane'), ('04', 'La Réunion'))
 See the "Code officiel géographique" on the INSEE website <www.insee.fr>.

`localflavor.fr.fr_region.REGION_2016_CHOICES` = (('01', 'Guadeloupe'), ('02', 'Martinique'), ('03', 'Guyane'), ('04', 'La Réunion'))
 France changed its regions in 2016, see:

7.22 Great Britain (gb)

7.22.1 Forms

GB-specific Form helpers.

class `localflavor.gb.forms.GBCountySelect` (*attrs=None*)
 A Select widget that uses a list of UK Counties/Regions as its choices.

class `localflavor.gb.forms.GBNationSelect` (*attrs=None*)
 A Select widget that uses a list of UK Nations as its choices.

class `localflavor.gb.forms.GBPostcodeField` (*max_length=None*, *min_length=None*, *strip=True*, **args*, ***kwargs*)
 A form field that validates its input is a UK postcode.

The regular expression used is sourced from the schema for British Standard BS7666 address types: <http://www.govtalk.gov.uk/gdsc/schemas/bs7666-v2-0.xsd>

The value is uppercased and a space added in the correct place, if required.

7.22.2 Data

This module contains various lists of regions and subdivisions in Great Britain.

Since subdivisions aren't clear this is supposed to be the most pragmatic collection of lists as possible. Your mileage may vary.

See <https://github.com/django/django-localflavor/pull/43> for a long discussion about it.

Changed in version 1.1.

`localflavor.gb.gb_regions.ENGLAND_REGION_CHOICES` = (('Bath and North East Somerset', 'Bath and North East Metropolitan and non-metropolitan counties of England (not the ceremonial counties) Retrieved 10th Nov 2014 from http://en.wikipedia.org/wiki/Metropolitan_and_non-metropolitan_counties_of_England

`localflavor.gb.gb_regions.GB_NATIONS_CHOICES` = (('England', 'England'), ('Northern Ireland', 'Northern Ireland Nations of the United Kingdom

`localflavor.gb.gb_regions.GB_REGION_CHOICES` = (('Bath and North East Somerset', 'Bath and North East Somerset All regions of United Kingdom

`localflavor.gb.gb_regions.NORTHERN_IRELAND_REGION_CHOICES` = (('Antrim', 'Antrim'), ('Armagh', 'Armagh') Counties of Northern Ireland (not the more recent, but less well-known, districts of Northern Ireland) Retrieved 10th Nov 2014 from http://en.wikipedia.org/wiki/Counties_of_Northern_Ireland

`localflavor.gb.gb_regions.SCOTTISH_REGION_CHOICES` = (('Aberdeen City', 'Aberdeen City'), ('Aberdeenshire', 'Aberdeenshire') Council areas of Scotland Retrieved 10th Nov 2014 from http://en.wikipedia.org/wiki/Subdivisions_of_Scotland#Council_areas

`localflavor.gb.gb_regions.WALES_REGION_CHOICES` = (('Blaenau Gwent', 'Blaenau Gwent'), ('Bridgend', 'Bridgend') Principal areas of Wales (not the preserved or historic counties) Retrieved 10th Nov 2014 from http://en.wikipedia.org/wiki/Local_government_in_Wales

7.23 Greece (gr)

7.23.1 Forms

Greek-specific forms helpers.

```
class localflavor.gr.forms.GRMobilePhoneNumberField(required=True, widget=None,
                                                    label=None, initial=None,
                                                    help_text='', error_messages=None,
                                                    show_hidden_initial=False,
                                                    validators=[], localize=False,
                                                    disabled=False, label_suffix=None)
```

Greek mobile phone field.

10 digits starting with 69 (could also start with +30 which is the country-code for greece)

```
class localflavor.gr.forms.GRPhoneNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text='', error_messages=None,
                                              show_hidden_initial=False,
                                              validators=[], localize=False,
                                              disabled=False, label_suffix=None)
```

Greek general phone field.

10 digits (can also start with +30 which is the country-code for greece)

```
class localflavor.gr.forms.GRPostalCodeField(max_length=None, min_length=None, *args,
                                             **kwargs)
```

Greek Postal code field.

Format: XXXXXX, where X is any digit, and first digit is not 0 or 9.

```
class localflavor.gr.forms.GRTaxNumberCodeField(allow_test_value=False, *args,
                                                  **kwargs)
```

Greek tax number field.

The `allow_test_value` option can be used to enable the usage of the non valid 000000000 value for testing and development

7.24 Hong Kong (hk)

7.24.1 Forms

Hong Kong specific Form helpers.

class `localflavor.hk.forms.HKPhoneNumberField` (**args, **kwargs*)

A form field that validates Hong Kong phone numbers.

The input format can be either one of the followings: 'XXXX-XXXX', '852-XXXX-XXXX', '(+852) XXXX-XXXX', 'XXXX XXXX', or 'XXXXXXXX'. The output format is 'XXXX-XXXX'.

Note: The phone number shall not start with 999, 992, or 112. And, it should start with either 2, 3, 5, 6, 8, or 9.

http://en.wikipedia.org/wiki/Telephone_numbers_in_Hong_Kong

7.25 Croatia (hr)

7.25.1 Forms

HR-specific Form helpers.

class `localflavor.hr.forms.HRCountySelect` (*attrs=None*)

A Select widget that uses a list of counties of Croatia as its choices.

class `localflavor.hr.forms.HRJMBAGField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

Unique Master Academic Citizen Number of Croatia (JMBAG) field.

This number is used by college students and professors in Croatia.

<http://www.cap.srce.hr/IzgledX.aspx>

class `localflavor.hr.forms.HRJMBGField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

Unique Master Citizen Number (JMBG) field.

The number is still in use in Croatia, but it is being replaced by OIB.

Source: http://en.wikipedia.org/wiki/Unique_Master_Citizen_Number

For who might be reimplementing: The "area" regular expression group is used to calculate the region where a person was registered. Additional validation can be implemented in accordance with it, however this could result in exclusion of legit immigrated citizens. Therefore, this field works for any ex-Yugoslavia country.

```
class localflavor.hr.forms.HRLicensePlateField (required=True, widget=None,
                                                label=None, initial=None,
                                                help_text='', error_messages=None,
                                                show_hidden_initial=False, validators=[],
                                                localize=False, disabled=False, la-
                                                bel_suffix=None)
```

Vehicle license plate of Croatia field.

Normalizes to the specific format below. Suffix is constructed from the shared letters of the Croatian and English alphabets.

Format examples: SB 123-A (but also supports more characters) ZG 1234-AA

Used for standardized license plates only.

```
class localflavor.hr.forms.HRLicensePlatePrefixSelect (attrs=None)
    A Select widget that uses a list of vehicle license plate prefixes of Croatia as its choices.
```

```
class localflavor.hr.forms.HROIBField (min_length=11, max_length=11, *args, **kwargs)
    Personal Identification Number of Croatia (OIB) field.
```

<http://www.oib.hr/>

```
class localflavor.hr.forms.HRPhoneNumberField (required=True, widget=None,
                                                label=None, initial=None,
                                                help_text='', error_messages=None,
                                                show_hidden_initial=False, valida-
                                                tors=[], localize=False, disabled=False,
                                                label_suffix=None)
```

Phone number of Croatia field.

Format: Complete country code or leading zero, area code prefix, 6 or 7 digit number. Validates fixed, mobile and FGSM numbers. Normalizes to a full number with country code (+385 prefix).

```
class localflavor.hr.forms.HRPhoneNumberPrefixSelect (attrs=None)
    A Select widget that uses a list of phone number prefixes of Croatia as its choices.
```

```
class localflavor.hr.forms.HRPostalCodeField (required=True, widget=None,
                                                label=None, initial=None,
                                                help_text='', error_messages=None,
                                                show_hidden_initial=False, valida-
                                                tors=[], localize=False, disabled=False,
                                                label_suffix=None)
```

Postal code of Croatia field.

It consists of exactly five digits ranging from 10000 to possibly less than 60000.

<http://www.posta.hr/main.aspx?id=66>

7.25.2 Data

```
localflavor.hr.hr_choices.HR_COUNTY_CHOICES = (('GZG', 'Grad Zagreb'), ('BBŽ', 'Bjelovarsko-bilogorska župa
Croatian Counties: http://en.wikipedia.org/wiki/ISO\_3166-2:HR Croatia doesn't have official abbreviations for
counties. The ones provided are in common use.
```

```
localflavor.hr.hr_choices.HR_LICENSE_PLATE_PREFIX_CHOICES = (('BJ', 'BJ'), ('BM', 'BM'), ('ČK', 'ČK'), ('
Only common license plate prefixes are provided. Special cases and obsolete prefixes are omitted.
http://hr.wikipedia.org/wiki/Dodatak:Popis\_registracijskih\_oznaka\_za\_cestovna\_vozila\_u\_Hrvatskoj
```

`localflavor.hr.hr_choices.HR_PHONE_NUMBER_PREFIX_CHOICES = (('1', '01'), ('20', '020'), ('21', '021'), ('22', '022'))`
 The list includes county and cellular network phone number prefixes.

7.26 Hungary (hu)

7.26.1 Forms

HU-specific Form helpers

class `localflavor.hu.forms.HUCountySelect` (*attrs=None*)
 A Select widget that uses a list of Hungarian Counties as its choices.

New in version 1.3.

7.26.2 Data

`localflavor.hu.hu_counties.HU_COUNTY_CHOICES = (('bacs_kiskun', 'Bács-Kiskun'), ('baranya', 'Baranya'), ('bekes', 'Békés'))`
 Hungarian counties: https://en.wikipedia.org/wiki/Counties_of_Hungary

7.27 Indonesia (id)

7.27.1 Forms

ID-specific Form helpers.

class `localflavor.id_.forms.IDLicensePlateField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

An Indonesian vehicle license plate field.

http://id.wikipedia.org/wiki/Tanda_Nomor_Kendaraan_Bermotor

Plus: “B 12345 12”

class `localflavor.id_.forms.IDLicensePlatePrefixSelect` (*attrs=None*)
 A Select widget that uses a list of vehicle license plate prefix code of Indonesia as its choices.

http://id.wikipedia.org/wiki/Tanda_Nomor_Kendaraan_Bermotor

class `localflavor.id_.forms.IDNationalIdentityNumberField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

An Indonesian national identity number (NIK/KTP#) field.

http://id.wikipedia.org/wiki/Nomor_Induk_Kependudukan

xx.xxxx.ddmmyy.xxxx - 16 digits (excl. dots)

```
class localflavor.id_.forms.IDPhoneNumberField (required=True, widget=None,
                                              label=None, initial=None,
                                              help_text='', error_messages=None,
                                              show_hidden_initial=False, validators=[],
                                              localize=False, disabled=False, la-
                                              bel_suffix=None)
```

An Indonesian telephone number field.

http://id.wikipedia.org/wiki/Daftar_kode_telepon_di_Indonesia

```
class localflavor.id_.forms.IDPostCodeField (required=True, widget=None, label=None, ini-
                                             tial=None, help_text='', error_messages=None,
                                             show_hidden_initial=False, validators=[],
                                             localize=False, disabled=False, la-
                                             bel_suffix=None)
```

An Indonesian post code field.

http://id.wikipedia.org/wiki/Kode_pos

```
class localflavor.id_.forms.IDProvinceSelect (attrs=None)
    A Select widget that uses a list of provinces of Indonesia as its choices.
```

7.27.2 Data

```
localflavor.id_.id_choices.PROVINCE_CHOICES = (('ACE', 'Aceh'), ('BLI', 'Bali'), ('BTN', 'Banten'), ('BKL', 'Ber-
Indonesia does not have an official Province code standard. I decided to use unambiguous and consistent (some
are common) 3-letter codes. http://id.wikipedia.org/wiki/Daftar\_provinsi\_Indonesia
```

```
localflavor.id_.id_choices.LICENSE_PLATE_PREFIX_CHOICES = (('A', 'Banten'), ('AA', 'Magelang'), ('AB', 'Yog-
License plate prefixes
```

7.28 Ireland (ie)

7.28.1 Forms

UK-specific Form helpers.

```
class localflavor.ie.forms.IECountySelect (attrs=None)
    A Select widget that uses a list of Irish Counties as its choices.
```

7.28.2 Data

```
localflavor.ie.ie_counties.IE_COUNTY_CHOICES = (('carlow', 'Carlow'), ('cavan', 'Cavan'), ('clare', 'Clare'), ('co-
Irish Counties: http://en.wikipedia.org/wiki/Counties\_of\_Ireland
```

7.29 Israel (il)

7.29.1 Forms

Israeli-specific form helpers.

It's normalised to 0XXX-XXXXXXX or 0XXX XXXXXXX format. The first string is the STD code which is a '0' followed by 2-4 digits. The second string is 8 digits if the STD code is 3 digits, 7 digits if the STD code is 4 digits and 6 digits if the STD code is 5 digits. The second string will start with numbers between 1 and 6. The separator is either a space or a hyphen.

```
class localflavor.in_.forms.INStateField(required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)
```

A form field that validates its input is a Indian state name or abbreviation.

It normalizes the input to the standard two-letter vehicle registration abbreviation for the given state or union territory

Changed in version 1.1: Added Telangana to list of states. More details at https://en.wikipedia.org/wiki/Telangana#Bifurcation_of_Andhra_Pradesh

```
class localflavor.in_.forms.INStateSelect(attrs=None)
```

A Select widget that uses a list of Indian states/territories as its choices.

Changed in version 1.1: Added Telangana to list of states. More details at https://en.wikipedia.org/wiki/Telangana#Bifurcation_of_Andhra_Pradesh

```
class localflavor.in_.forms.INZipCodeField(max_length=None, min_length=None, *args, **kwargs)
```

A form field that validates input as an Indian zip code, with the format XXXXXXX.

7.30.2 Data

```
localflavor.in_.in_states.STATE_CHOICES = (('KA', 'Karnataka'), ('AP', 'Andhra Pradesh'), ('KL', 'Kerala'), ('TN', 'Tamil Nadu'))
```

A list of states

```
localflavor.in_.in_states.STATES_NORMALIZED = {'kerala': 'KL', 'andhra pradesh': 'AP', 'telangana': 'TG', 'tr': 'TR'}
```

Normalized state names

7.31 Iceland (is)

7.31.1 Forms

Iceland specific form helpers.

```
class localflavor.is_.forms.ISIdNumberField(max_length=11, min_length=10, *args, **kwargs)
```

Icelandic identification number (kennitala).

This is a number every citizen of Iceland has.

```
class localflavor.is_.forms.ISPhoneNumberField(max_length=8, min_length=7, *args, **kwargs)
```

Icelandic phone number.

Seven digits with an optional hyphen or space after the first three digits.

```
class localflavor.is_.forms.ISPostalCodeSelect(attrs=None)
```

A Select widget that uses a list of Icelandic postal codes as its choices.

7.31.2 Data

```
localflavor.is_.is_postalcodes.IS_POSTALCODES = (('101', '101 Reykjavík'), ('103', '103 Reykjavík'), ('104', '104 Reykjavík'))
```

A list of Icelandic postal codes

7.32 Italy (it)

7.32.1 Forms

IT-specific Form helpers.

```
class localflavor.it.forms.ITPhoneNumberField(max_length=None, min_length=None, strip=True, *args, **kwargs)
```

A form field that validates input as an Italian phone number.

Will strip any +39 country prefix from the number.

New in version 1.1.

```
class localflavor.it.forms.ITProvinceSelect(attrs=None)
```

A Select widget that uses a list of IT provinces as its choices.

```
class localflavor.it.forms.ITRegionProvinceSelect(attrs=None)
```

A Select widget that uses a named group list of IT regions mapped to regions as its choices.

```
class localflavor.it.forms.ITRegionSelect(attrs=None)
```

A Select widget that uses a list of IT regions as its choices.

```
class localflavor.it.forms.ITSocialSecurityNumberField(max_length=None, min_length=None, *args, **kwargs)
```

A form field that validates Italian Tax code (codice fiscale) for both persons and entities.

For reference see <http://www.agenziaentrate.it/> and search for:

- ‘Informazioni sulla codificazione delle persone fisiche’ for persons’ SSN
- ‘Codice fiscale Modello AA5/6’ for entities’ SSN

Changed in version 1.1.

The `ITSocialSecurityNumberField` now also accepts SSN values for entities (numeric-only form).

```
class localflavor.it.forms.ITVatNumberField(required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)
```

A form field that validates Italian VAT numbers (partita IVA).

```
class localflavor.it.forms.ITZipCodeField(max_length=None, min_length=None, *args, **kwargs)
```

A form field that validates input as an Italian zip code.

Valid codes must have five digits.

7.32.2 Utilities

`localflavor.it.util.ssn_check_digit` (*value*)
 Calculate Italian social security number check digit.

`localflavor.it.util.ssn_validation` (*ssn_value*)
 Validate Italian SSN for persons
 ValueError is raised if validation fails.

`localflavor.it.util.vat_number_check_digit` (*vat_number*)
 Calculate Italian VAT number check digit.

`localflavor.it.util.vat_number_validation` (*vat_number*)
 Validate Italian VAT number. Used also for entities SSN validation.
 ValueError is raised if validation fails.

7.32.3 Data

`localflavor.it.it_province.PROVINCE_CHOICES` = (('AG', 'Agrigento'), ('AL', 'Alessandria'), ('AN', 'Ancona'), ('A
 An alphabetical list of provinces

`localflavor.it.it_province.PROVINCE_REGIONS` = {'UD': 'FVG', 'SI': 'TOS', 'LE': 'PUG', 'MT': 'BAS', 'BA': 'PU
 A dictionary of provinces mapped to regions
 New in version 1.1.

`localflavor.it.it_region.REGION_CHOICES` = (('ABR', 'Abruzzo'), ('BAS', 'Basilicata'), ('CAL', 'Calabria'), ('CAM
 An alphabetical list of regions

`localflavor.it.it_region.REGION_PROVINCES` = {'BAS': ['MT', 'PZ'], 'LOM': ['BG', 'BS', 'CO', 'CR', 'LC', 'LO'
 A dictionary of regions mapped to provinces
 New in version 1.1.

`localflavor.it.it_region.REGION_PROVINCE_CHOICES` = [('Abruzzo', (('CH', 'Chieti'), ('AQ', 'L'Aquila'), ('PE', '
 A alphabetical list of provinces mapped to regions
 New in version 1.1.

7.33 Japan (jp)

7.33.1 Forms

JP-specific Form helpers.

class `localflavor.jp.forms.JPPostalCodeField` (*max_length=None, min_length=None, *args, **kwargs*)
 A form field that validates its input is a Japanese postcode.
 Accepts 7 digits, with or without a hyphen.

clean (*value*)
 Validates the input and returns a string that contains only numbers.
 Returns an empty string for empty values.

class `localflavor.jp.forms.JPPrefectureCodeSelect` (*attrs=None*)
 A Select widget for Japanese prefecture codes.

It uses a list of Japanese prefectures as its choices and the prefectures code as the post value.

class `localflavor.jp.forms.JPPrefectureSelect` (*attrs=None*)
 A Select widget that uses a list of Japanese prefectures as its choices.

7.33.2 Data

`localflavor.jp.jp_prefectures.JP_PREFECTURES` = (('hokkaido', 'Hokkaido'), ('aomori', 'Aomori'), ('iwate', 'Iwate'))
 A list of prefectures

7.34 Kuwait (kw)

7.34.1 Forms

Kuwait-specific Form helpers.

class `localflavor.kw.forms.KWCivilIDNumberField` (*max_length=12, min_length=12, *args, **kwargs*)

Kuwaiti Civil ID numbers are 12 digits, second to seventh digits represents the person's birthdate.

Checks the following rules to determine the validity of the number:

- The number consist of 12 digits.
- The birthdate of the person is a valid date.
- The calculated checksum equals to the last digit of the Civil ID.

class `localflavor.kw.forms.KWGovernorateSelect` (*attrs=None*)
 A Select widget that uses a list of Kuwait governorates as its choices.

7.35 Lithuania (lt)

7.35.1 Forms

class `localflavor.lt.forms.LTCountySelect` (*attrs=None*)
 A select field with the Lithuanian counties as choices

class `localflavor.lt.forms.LTIDCodeField` (**args, **kwargs*)
 A form field that validates as Lithuanian ID Code.

Checks:

- Made of exactly 11 decimal numbers.
- Checksum is correct.
- ID contains valid date.

valid_date (*value*)

Check if date in ID code is valid.

We won't check for dates in future as it would become too restrictive.

class `localflavor.lt.forms.LTMunicipalitySelect` (*attrs=None*)
A select field with the Lithuanian municipalities as choices

class `localflavor.lt.forms.LTPhoneField` (*mobile=True, landline=True, emergency=False, landline_local=False, service=False, **kwargs*)
Form field that validates as Lithuanian phone number.

You can accept any permutation of following phone numbers:

- Emergency (01, 02, 03, 04, 112)
- Mobile (370 600 00 000)
- Landline
- Service numbers

If you accept landline numbers, you can opt in to accepting local landline numbers too. Local landline numbers are numbers without area code.

This field does not accept multiple numbers (as separated by /).

The field tries its best to convert the number into one you can call to internationally. Currently emergency and most of `landline_local` numbers are not converted.

New in version 1.1.

class `localflavor.lt.forms.LTPostalCodeField` (*required=True, label=None, widget=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

A form field that validates and normalizes Lithuanian postal codes.

Lithuanian postal codes in following forms accepted:

- XXXXX
- LT-XXXXX

7.35.2 Data

`localflavor.lt.lt_choices.COUNTY_CHOICES` = (('alytus', 'Alytus'), ('kaunas', 'Kaunas'), ('klaipeda', 'Klaipda'), ('m...
Alphabetically sorted list of Lithuanian counties.

`localflavor.lt.lt_choices.MUNICIPALITY_CHOICES` = (('akmene', 'Akmen district'), ('alytus_c', 'Alytus city'), ('al...
Alphabetically sorted lists of Lithuanian municipalities.

7.36 Latvia (lv)

New in version 1.1.

7.36.1 Forms

class `localflavor.lv.forms.LVMunicipalitySelect` (*attrs=None*)
A select field of Latvian municipalities.

```
class localflavor.lv.forms.LVPersonalCodeField(required=True, widget=None,
label=None, initial=None,
help_text='', error_messages=None,
show_hidden_initial=False, validators=[],
localize=False, disabled=False, label_suffix=None)
```

A form field that validates input as a Latvian personal code.

static `lv_checksum` (*value*)

Takes a string of 10 digits as input, returns check digit.

```
class localflavor.lv.forms.LVPostalCodeField(required=True, widget=None,
label=None, initial=None,
help_text='', error_messages=None,
show_hidden_initial=False, validators=[],
localize=False, disabled=False,
label_suffix=None)
```

A form field that validates and normalizes Latvian postal codes.

Latvian postal codes in following forms accepted:

- XXXX
- LV-XXXX

7.36.2 Data

```
localflavor.lv.lv_choices.MUNICIPALITY_CHOICES = (('DGV', 'Daugavpils'), ('JEL', 'Jelgava'), ('JKB', 'Jkabpils'))
```

A list of Latvian municipalities and republican cities. Identifiers based on ISO 3166-2:LV.
https://en.wikipedia.org/wiki/ISO_3166-2:LV

7.37 Macedonia (mk)

7.37.1 Forms

```
class localflavor.mk.forms.MKIdentityCardNumberField(*args, **kwargs)
```

A Macedonian ID card number.

Accepts both old and new format.

```
class localflavor.mk.forms.MKMunicipalitySelect(attrs=None)
```

A form `Select` widget that uses a list of Macedonian municipalities as choices.

The label is the name of the municipality and the value is a 2 character code for the municipality.

```
class localflavor.mk.forms.UMCNField(*args, **kwargs)
```

A form field that validates input as a unique master citizen number.

The format of the unique master citizen number has been kept the same from Yugoslavia. It is still in use in other countries as well, it is not applicable solely in Macedonia. For more information see: https://secure.wikimedia.org/wikipedia/en/wiki/Unique_Master_Citizen_Number

A value will pass validation if it complies to the following rules:

- Consists of exactly 13 digits
- The first 7 digits represent a valid past date in the format DDMMYYYY

class `localflavor.mx.forms.MXCLABEField` (*min_length=18, max_length=18, *args, **kwargs*)
 This field validates a CLABE (Clave Bancaria Estandarizada).

A CLABE is a 18-digits long number. The first 6 digits denote bank and branch number. The remaining 12 digits denote an account number, plus a verifying digit.

More info: <https://en.wikipedia.org/wiki/CLABE>

New in version 1.4.

class `localflavor.mx.forms.MXCURPField` (*min_length=18, max_length=18, *args, **kwargs*)
 A field that validates a Mexican Clave Única de Registro de Población.

The CURP is integrated by a juxtaposition of characters following the next pattern:

Index	Format	Accepted Characters
1	X	Any letter
2	X	Any vowel
3-4	XX	Any letter
5-10	YYMMDD	Any valid date
11	X	Either <i>H</i> or <i>M</i> , depending on the person's gender
12-13	XX	Any valid acronym for a state in Mexico
14-16	XXX	Any consonant
17	X	Any number between 0 and 9 or any letter
18	X	Any number between 0 and 9

More info about this: <http://www.condusef.gob.mx/index.php/clave-unica-de-registro-de-poblacion-curp>

class `localflavor.mx.forms.MXRFCField` (*min_length=12, max_length=13, *args, **kwargs*)
 A form field that validates a Mexican *Registro Federal de Contribuyentes*.

Validates either *Persona física* or *Persona moral*. The *Persona física* RFC string is integrated by a juxtaposition of characters following the next pattern:

Index	Format	Accepted Characters
1	X	Any letter
2	X	Any vowel
3-4	XX	Any letter
5-10	YYMMDD	Any valid date
11-12	XX	Any letter or number between 0 and 9
13	X	Any digit between 0 and 9 or the letter <i>A</i>

The *Persona moral* RFC string is integrated by a juxtaposition of characters following the next pattern:

Index	Format	Accepted Characters
1-3	XXX	Any letter including <i>&</i> and <i>Ñ</i> chars
4-9	YYMMDD	Any valid date
10-11	XX	Any letter or number between 0 and 9
12	X	Any number between 0 and 9 or the letter <i>A</i>

More info about this: [http://es.wikipedia.org/wiki/Registro_Federal_de_Contribuyentes_\(M%C3%A9xico\)](http://es.wikipedia.org/wiki/Registro_Federal_de_Contribuyentes_(M%C3%A9xico))

class `localflavor.mx.forms.MXSocialSecurityNumberField` (*min_length=11, max_length=11, *args, **kwargs*)

A field that validates a Mexican Social Security Number.

The Social Security Number is integrated by a juxtaposition of digits following the next pattern:

In- dex	Required numbers
1-2	The number of the branch office where the Social Security Number was designated.
3-4	The year of inscription to the Social Security.
5-6	The year of birth of the Social Security Number owner.
7- 10	The progressive number of procedure for the IMSS. (This digit is provided exclusively by the Institute as it regards the Folio number of such procedure).
11	The verification digit.

class `localflavor.mx.forms.MXStateSelect` (*attrs=None*)
 A Select widget that uses a list of Mexican states as its choices.

class `localflavor.mx.forms.MXZipCodeField` (**args, **kwargs*)
 A form field that accepts a Mexican Zip Code.

More info about this: http://en.wikipedia.org/wiki/List_of_postal_codes_in_Mexico

`localflavor.mx.forms.RFC_INCONVENIENT_WORDS = ['BUEI', 'BUEY', 'CAC', 'CACO', 'CAGA', 'CAGO', 'CAKA']`
 This is the list of inconvenient words according to the *Anexo IV* of the document described in the next link:
http://www.sisi.org.mx/jspsi/documentos/2005/seguimiento/06101/0610100162005_065.doc

7.39.2 Models

class `localflavor.mx.models.MXCLABEField` (**args, **kwargs*)
 A model field that forms represent as a forms.MXCURPField field and stores the value of a valid Mexican CLABE.

New in version 1.4.

class `localflavor.mx.models.MXCURPField` (**args, **kwargs*)
 A model field that forms represent as a forms.MXCURPField field and stores the value of a valid Mexican CURP.

class `localflavor.mx.models.MXRFCField` (**args, **kwargs*)
 A model field that forms represent as a forms.MXRFCField field and stores the value of a valid Mexican RFC.

class `localflavor.mx.models.MXSocialSecurityNumberField` (**args, **kwargs*)
 A model field that forms represent as a forms.MXSocialSecurityNumberField field.
 It stores the value of a valid Mexican Social Security Number.

class `localflavor.mx.models.MXStateField` (**args, **kwargs*)
 A model field that stores the three-letter Mexican state abbreviation in the database.

class `localflavor.mx.models.MXZipCodeField` (**args, **kwargs*)
 A model field that forms represent as a forms.MXZipCodeField field and stores the five-digit Mexican zip code.

7.39.3 Data

`localflavor.mx.mx_states.STATE_CHOICES = (('AGU', 'Aguascalientes'), ('BCN', 'Baja California'), ('BCS', 'Baja C...))`
 All 31 states, plus the *Distrito Federal*.

7.40 The Netherlands (nl)

7.40.1 Forms

NL-specific Form helpers.

```
class localflavor.nl.forms.NLPhoneNumberField(max_length=None, min_length=None,
                                             strip=True, *args, **kwargs)
```

A Dutch telephone number field.

```
class localflavor.nl.forms.NLProvinceSelect(attrs=None)
```

A Select widget that uses a list of provinces of the Netherlands as it's choices.

```
class localflavor.nl.forms.NLSoFiNumberField(*args, **kwargs)
```

A Dutch social security number (SoFi/BSN) field.

<http://nl.wikipedia.org/wiki/Sofinummer>

```
class localflavor.nl.forms.NLZipCodeField(max_length=None, min_length=None, strip=True,
                                           *args, **kwargs)
```

A Dutch zip code field.

7.40.2 Models

```
class localflavor.nl.models.NLBankAccountNumberField(*args, **kwargs)
```

A Dutch bank account model field.

This model field uses `validators.NLBankAccountNumberFieldValidator` for validation.

New in version 1.1.

```
class localflavor.nl.models.NLPhoneNumberField(*args, **kwargs)
```

Dutch phone number model field.

This model field uses `validators.NLPhoneNumberFieldValidator` for validation.

New in version 1.3.

```
description = 'Dutch phone number'
```

```
formfield (**kwargs)
```

```
validator = [localflavor.nl.validators.NLPhoneNumberFieldValidator object at 0x7f0fbfdaf0f0]
```

```
class localflavor.nl.models.NLProvinceField(*args, **kwargs)
```

A Dutch Province field.

New in version 1.3.

```
deconstruct ()
```

```
description = 'Dutch province'
```

```
class localflavor.nl.models.NLSoFiNumberField(*args, **kwargs)
```

A Dutch social security number (SoFi).

This model field uses `validators.NLSoFiNumberFieldValidator` for validation.

New in version 1.3.

```
description = 'Dutch social security number (SoFi)'
```

```
formfield (**kwargs)
```

validators = [<localflavor.nl.validators.NLSoFiNumberFieldValidator object at 0x7f0fbfdafcf8>]

class localflavor.nl.models.NLZipCodeField(*args, **kwargs)

A Dutch zip code model field.

This model field uses validators.NLZipCodeFieldValidator for validation.

New in version 1.3.

description = 'Dutch zipcode'

formfield (**kwargs)

to_python (value)

validators = [<localflavor.nl.validators.NLZipCodeFieldValidator object at 0x7f0fbfdc1c50>]

7.40.3 Data

localflavor.nl.nl_provinces.PROVINCE_CHOICES = (('DR', 'Drenthe'), ('FL', 'Flevoland'), ('FR', 'Fryslân'), ('GL',

An alphabetical list of provinces

7.41 Norway (no)

7.41.1 Forms

Norwegian-specific Form helpers.

class localflavor.no.forms.NOMunicipalitySelect(attrs=None)

A Select widget that uses a list of Norwegian municipalities (fylker) as its choices.

class localflavor.no.forms.NOPhoneNumberField(max_length=None, min_length=None, *args, **kwargs)

Field with phonenumber validation.

Requires a phone number with 8 digits and optional country code

class localflavor.no.forms.NOSocialSecurityNumber(required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)

Algorithm is documented at <http://no.wikipedia.org/wiki/Personnummer>.

class localflavor.no.forms.NOZipCodeField(max_length=None, min_length=None, *args, **kwargs)

A form field that validates input as a Norwegian zip code.

Valid codes have four digits.

7.41.2 Data

localflavor.no.no_municipalities.MUNICIPALITY_CHOICES = (('akershus', 'Akershus'), ('austagder', 'Aust-Agd

An alphabetical list of Norwegian municipalities (fylker) for use as *choices* in a formfield.

7.42 New Zealand (nz)

New in version 1.1.

7.42.1 Forms

New Zealand specific form helpers.

```
class localflavor.nz.forms.NZBankAccountNumberField(required=True, widget=
                                                    get=None, label=None, ini-
                                                    tial=None, help_text='',
                                                    error_messages=None,
                                                    show_hidden_initial=False, val-
                                                    idators=[], localize=False, dis-
                                                    abled=False, label_suffix=None)
```

A form field that validates its input as New Zealand bank account number.

Formats:

```
XX-XXXX-XXXXXXXX-XX
```

```
XX-XXXX-XXXXXXXX-XXX
```

Where:

- the first two digits is the bank ID
- the next four digits are the branch number where the account was opened
- the next 7 digits are the account numbers
- the last two or three digits define type of the account.

```
class localflavor.nz.forms.NZNorthIslandCouncilSelect(attrs=None)
```

A select widget with list of New Zealand North Island city and district councils as its choices.

```
class localflavor.nz.forms.NZPhoneNumberField(required=True, widget=None,
                                              label=None, initial=None,
                                              help_text='', error_messages=None,
                                              show_hidden_initial=False, valida-
                                              tors=[], localize=False, disabled=False,
                                              label_suffix=None)
```

A form field that validates its input as New Zealand phone number.

```
class localflavor.nz.forms.NZPostCodeField(*args, **kwargs)
```

A form field that validates its input as New Zealand postal code.

```
class localflavor.nz.forms.NZProvinceSelect(attrs=None)
```

A select widget with list of New Zealand provinces as its choices.

```
class localflavor.nz.forms.NZRegionSelect(attrs=None)
```

A select widget with list of New Zealand regions as its choices.

```
class localflavor.nz.forms.NZSouthIslandCouncilSelect(attrs=None)
```

A select widget with list of New Zealand South Island city and district councils as its choices.

7.42.2 Data

`localflavor.nz.nz_regions.REGION_CHOICES` = (('NZ-NTL', 'Northland'), ('NZ-AUK', 'Auckland'), ('NZ-WKO', 'Waikato'))
 A list of regions

`localflavor.nz.nz_provinces.PROVINCE_CHOICES` = (('Auckland', 'Auckland'), ('Taranaki', 'Taranaki'), ('Hawke's Bay', 'Hawke's Bay'))
 A list of provinces (abolished in 1876, use regions instead)

`localflavor.nz.nz_councils.NORTH_ISLAND_COUNCIL_CHOICES` = (('Far North', 'Far North District'), ('Whangarei', 'Whangarei'))
 A list of North Island city and district councils

`localflavor.nz.nz_councils.SOUTH_ISLAND_COUNCIL_CHOICES` = (('Tasman', 'Tasman District'), ('Nelson', 'Nelson'))
 A list of South Island city and district councils

7.43 Peru (pe)

7.43.1 Forms

PE-specific Form helpers.

class `localflavor.pe.forms.PEDNIField` (*max_length=8, min_length=8, *args, **kwargs*)
 A field that validates Documento Nacional de Identidad (DNI) numbers.

clean (*value*)
 Value must be a string in the XXXXXXXX formats.

class `localflavor.pe.forms.PERUCField` (*max_length=11, min_length=11, *args, **kwargs*)
 This field validates a RUC (Registro Unico de Contribuyentes).

A RUC is of the form XXXXXXXXXXXXX.

clean (*value*)
 Value must be an 11-digit number.

class `localflavor.pe.forms.PERRegionSelect` (*attrs=None*)
 A Select widget that uses a list of Peruvian Regions as its choices.

7.43.2 Data

`localflavor.pe.pe_region.REGION_CHOICES` = (('AMA', 'Amazonas'), ('ANC', 'Ancash'), ('APU', 'Apurímac'), ('ARE', 'Arequipa'))
 A list of Peru regions as *choices* in a formfield.

7.44 Pakistan (pk)

7.44.1 Forms

Pakistani-specific Form helpers.

class `localflavor.pk.forms.PKPhoneNumberField` (*max_length=None, min_length=None, strip=True, *args, **kwargs*)
 A form field that validates input as an Pakistani phone number.

Valid numbers have nine to eleven digits.

clean (*value*)

Validate a phone number.

Strips parentheses, whitespace and hyphens.

class `localflavor.pk.forms.PKPostCodeField` (**args, **kwargs*)
Pakistani post code field.

Assumed to be 5 digits.

class `localflavor.pk.forms.PKStateSelect` (*attrs=None*)
A Select widget that uses a list of Pakistani states/territories as its choices.

7.44.2 Models

class `localflavor.pk.models.PKPhoneNumberField` (**args, **kwargs*)
A model field that checks that the value is a valid Pakistani phone number (nine to eleven digits).

description = 'Pakistani Phone number'

formfield (***kwargs*)

class `localflavor.pk.models.PKPostCodeField` (**args, **kwargs*)
A model field that stores the five-digit Pakistani postcode in the database

Forms represent it as a `PKPostCodeField` field.

description = 'Pakistani Postcode'

formfield (***kwargs*)

class `localflavor.pk.models.PKStateField` (**args, **kwargs*)
A model field that stores the five-letter Pakistani state abbreviation in the database.

It is represented with `STATE_CHOICES` `choices`.

deconstruct ()

description = 'Pakistani State'

7.44.3 Data

`localflavor.pk.pk_states.STATE_CHOICES` = (('PK-JK', 'Azad Jammu & Kashmir'), ('PK-BA', 'Balochistan'), ('PK-

An alphabetical list of states for use as *choices* in a formfield.

7.45 Poland (p1)

7.45.1 Forms

Polish-specific form helpers.

class `localflavor.pl.forms.PLCountySelect` (*attrs=None*)
A select widget with list of Polish administrative units as choices.

class `localflavor.pl.forms.PLNIPField` (*max_length=None, min_length=None, *args, **kwargs*)
A form field that validates as Polish Tax Number (NIP).

Valid forms are: XXX-YYY-YY-YY, XXX-YY-YY-YYY or XXXYYYYYYYY. Checksum algorithm based on documentation at <http://wipos.p.lodz.pl/zylla/ut/nip-rego.html>

has_valid_checksum (*number*)
 Calculates a checksum with the provided algorithm.

class `localflavor.pl.forms.PLNationalIDCardNumberField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates as Polish National ID Card Number.

Checks the following rules:

- the length consist of 3 letter and 6 digits
- has a valid checksum

The algorithm is documented at http://en.wikipedia.org/wiki/Polish_identity_card.

has_valid_checksum (*number*)
 Calculates a checksum with the provided algorithm.

class `localflavor.pl.forms.PLPESELField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates as Polish Identification Number (PESEL).

Checks the following rules:

- the length consist of 11 digits
- has a valid checksum
- contains a valid birth date

The algorithm is documented at <http://en.wikipedia.org/wiki/PESEL>.

Changed in version 1.4.

has_valid_birth_date (*number*)
 Checks whether the birth date encoded in PESEL is valid.

has_valid_checksum (*number*)
 Calculates a checksum with the provided algorithm.

class `localflavor.pl.forms.PLPostalCodeField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates as Polish postal code.

Valid code is XX-XXX where X is digit.

class `localflavor.pl.forms.PLProvinceSelect` (*attrs=None*)
 A select widget with list of Polish administrative provinces as choices.

class `localflavor.pl.forms.PLREGONField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates its input is a REGON number.

Valid regon number consists of 9 or 14 digits. See http://www.stat.gov.pl/bip/regon_ENG_HTML.htm for more information.

has_valid_checksum (*number*)
 Calculates a checksum with the provided algorithm.

7.45.2 Data

`localflavor.pl.pl_administrativeunits.ADMINISTRATIVE_UNIT_CHOICES = (('wroclaw', 'Wrocław'), ('jelenia`

`localflavor.pl.pl_voivodeships.VOIVODESHIP_CHOICES = (('lower_silesia', 'Lower Silesian'), ('kuyavia-pomeran
Polish voivodeship as in http://en.wikipedia.org/wiki/Poland#Administrative_division`

7.46 Portugal (pt)

7.46.1 Forms

django_localflavot_pt.forms

Contains PT-specific Django form helpers.

```
class localflavor.pt.forms.PTCitizenCardNumberField(required=True, widget=
get=None, label=None, initial=None, help_text='',
error_messages=None,
show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None)
```

A field which validates Portuguese Citizen Card numbers (locally CC - 'Cartão do Cidadão').

- Citizen Card numbers have the format **XXXXXXXXXXYYX** or **XXXXXXXX-YYYYX** (where X is a digit and Y is an alphanumeric character).
- Citizen Card numbers validate as per <http://bit.ly/RPOBzW>.
- The input string may or may not have an hyphen separating the identity number from the document's check-digits.
- This field does NOT validate old ID card numbers (locally BI - 'Bilhete de Identidade').

```
class localflavor.pt.forms.PTPhoneNumberField(required=True, widget=None,
label=None, initial=None,
help_text='', error_messages=None,
show_hidden_initial=False, validators=[], localize=False, disabled=False,
label_suffix=None)
```

A field which validates Portuguese phone numbers.

- Phone numbers have at least 3 and at most 9 digits and may optionally be prefixed with '00351' or '+351'.
- The input string is allowed to contain spaces (though they will be stripped).

```
class localflavor.pt.forms.PTRegionSelect(attrs=None)
```

A select widget which uses a list of Portuguese regions as its choices.

- Regions correspond to the Portuguese 'distritos' and 'regiões autónomas' as per ISO3166:2-PT.

```
class localflavor.pt.forms.PTSocialSecurityNumberField(required=True, widget=
get=None, label=None, initial=None, help_text='',
error_messages=None,
show_hidden_initial=False, validators=[], localize=False, disabled=False,
label_suffix=None)
```

A field which validates Portuguese Social Security numbers.

(locally NISS - 'Número de Identificação na Segurança Social'). - Social Security numbers must be in the format XYYYYYYYYYYY (where X is either 1 or 2 and Y is any other digit).

```
class localflavor.pt.forms.PTZipCodeField(max_length=None, min_length=None, *args,
                                         **kwargs)
```

A field which validates Portuguese zip codes.

NOTE - Zip codes have the format XYYY-YYY (where X is a digit between 1 and 9 and Y is any other digit).

7.46.2 Data

```
localflavor.pt.pt_regions.REGION_CHOICES = (('01', 'Aveiro'), ('02', 'Beja'), ('03', 'Braga'), ('04', 'Bragança'), ('05', 'Castelo Branco'), ('06', 'Coimbra'), ('07', 'Évora'), ('08', 'Faro'), ('09', 'Guarda'), ('10', 'Hespanha'), ('11', 'Leiria'), ('12', 'Lisboa'), ('13', 'Madeira'), ('14', 'Matosinhos'), ('15', 'N. Alentejo'), ('16', 'N. Trás-os-Montes'), ('17', 'Porto'), ('18', 'Região Autónoma da Açores'), ('19', 'Setúbal'), ('20', 'Tribalgarça'), ('21', 'Viana do Castelo'), ('22', 'Vila Real'), ('23', 'Algarve'))
```

A tuple representing Portuguese regions (as per ISO3166:2-PT).

7.47 Paraguay (py)

7.47.1 Forms

PY-specific Form helpers.

```
class localflavor.py_.forms.PyDepartmentSelect(attrs=None)
```

A Select widget with a list of Paraguayan departments as choices.

```
class localflavor.py_.forms.PyNumberedDepartmentSelect(attrs=None)
```

A Select widget with a roman numbered list of Paraguayan departments as choices.

7.47.2 Data

```
localflavor.py_.py_department.DEPARTMENT_CHOICES = (('AG', 'Alto Paraguay'), ('AA', 'Alto Paraná'), ('AM', 'Asunción'), ('AN', 'Itapúa'), ('AS', 'Ñeembucú'), ('BO', 'Boquerón'), ('BR', 'Brazo Oriental del Paraguay'), ('BU', 'Boquerón'), ('CA', 'Caaguazú'), ('CO', 'Cordillera Occidental'), ('CU', 'Candrupy'), ('DI', 'Deseado'), ('DU', 'Deseado'), ('EP', 'Encarnación'), ('FR', 'Frente Oriental del Mbaracayá'), ('GU', 'Guairá'), ('IT', 'Itapúa'), ('MA', 'Misiones'), ('MO', 'Misiones'), ('NE', 'Neembucú'), ('PA', 'Paraguari'), ('PE', 'Paraguari'), ('PI', 'Paraguari'), ('PO', 'Paraguari'), ('PR', 'Paraguari'), ('PY', 'Paraguari'), ('RE', 'Resistencia del Este'), ('RU', 'Resistencia del Este'), ('SD', 'San Pedro del Paraná'), ('ST', 'San Pedro del Paraná'), ('SU', 'San Pedro del Paraná'), ('TA', 'Tandubay'), ('TI', 'Tandubay'), ('TR', 'Tandubay'), ('VA', 'Villarrica'), ('VE', 'Villarrica'), ('VI', 'Villarrica'), ('VO', 'Villarrica'))
```

<http://www.statoids.com/upy.html>

```
localflavor.py_.py_department.DEPARTMENT_ROMAN_CHOICES = (('CN', 'I Concepción'), ('SP', 'II San Pedro'), ('AG', 'III Alto Paraguay'), ('AA', 'IV Alto Paraná'), ('AM', 'V Asunción'), ('AN', 'VI Itapúa'), ('AS', 'VII Ñeembucú'), ('BO', 'VIII Boquerón'), ('BR', 'IX Brazo Oriental del Paraguay'), ('BU', 'X Boquerón'), ('CA', 'XI Caaguazú'), ('CO', 'XII Cordillera Occidental'), ('CU', 'XIII Candrupy'), ('DI', 'XIV Deseado'), ('DU', 'XV Deseado'), ('EP', 'XVI Encarnación'), ('FR', 'XVII Frente Oriental del Mbaracayá'), ('GU', 'XVIII Guairá'), ('IT', 'XIX Itapúa'), ('MA', 'XX Misiones'), ('MO', 'XXI Misiones'), ('NE', 'XXII Neembucú'), ('PA', 'XXIII Paraguari'), ('PE', 'XXIV Paraguari'), ('PI', 'XXV Paraguari'), ('PO', 'XXVI Paraguari'), ('PR', 'XXVII Paraguari'), ('PY', 'XXVIII Paraguari'), ('RE', 'XXIX Resistencia del Este'), ('RU', 'XXX Resistencia del Este'), ('SD', 'XXXI San Pedro del Paraná'), ('ST', 'XXXII San Pedro del Paraná'), ('SU', 'XXXIII San Pedro del Paraná'), ('TA', 'XXXIV Tandubay'), ('TI', 'XXXV Tandubay'), ('TR', 'XXXVI Tandubay'), ('VA', 'XXXVII Villarrica'), ('VE', 'XXXVIII Villarrica'), ('VI', 'XXXIX Villarrica'), ('VO', 'XXXI Villarrica'))
```

list of departments sorted by its roman number

7.48 Romania (ro)

7.48.1 Forms

Romanian specific form helpers.

```
class localflavor.ro.forms.ROCIFField(max_length=10, min_length=2, *args, **kwargs)
```

A Romanian fiscal identity code (CIF) field.

For CIF validation algorithm see: https://ro.wikipedia.org/wiki/Cod_de_Identificare_Fiscal%C4%83

```
clean(value)
```

CIF validation.

Args: value: the CIF code

```
class localflavor.ro.forms.ROCNPField(max_length=13, min_length=13, *args, **kwargs)
```

A Romanian personal identity code (CNP) field.

For CNP validation algorithm see: https://ro.wikipedia.org/wiki/Cod_numeric_personal

clean (*value*)
 CNP validations.

Args: value: the CNP code

class `localflavor.ro.forms.ROCountyField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

A form field that validates its input is a Romanian county name or abbreviation.

It normalizes the input to the standard vehicle registration abbreviation for the given county.

WARNING: This field will only accept names written with diacritics (using comma bellow for *and*); consider using `ROCountySelect` if this behavior is unacceptable for you

For more information regarding diacritics see *Comma-below (and) versus cedilla (ş and ț)* and *Unicode and HTML* sections from: [Romanian alphabet](#).

Example:

Arge => valid (comma bellow)
 Argeş => invalid (cedilla)
 Arges => invalid (no diacritic)

class `localflavor.ro.forms.ROCountySelect` (*attrs=None*)
 A Select widget that uses a list of Romanian counties (judee) as its choices.

class `localflavor.ro.forms.ROIBANField` (**args, **kwargs*)
 Romanian International Bank Account Number (IBAN) field.

Changed in version 1.1: Validation error messages changed to the messages used in `IBANFormField`

Deprecated since version 1.1: Use `IBANFormField` with `included_countries=('RO',)` option instead.

class `localflavor.ro.forms.ROPhoneNumberField` (*max_length=20, min_length=3, *args, **kwargs*)

Romanian phone number field.

Changed in version 1.1: Made the field also accept national short phone numbers and 7-digit regional phone numbers besides the regular ones. Official documentation (in English): http://www.ancom.org.ro/en/pnn_1300
 Official documentation (in Romanian): http://www.ancom.org.ro/pnn_1300

clean (*value*)
 Strips braces, dashes, dots and spaces. Checks the final length.

Args: value: the phone number

class `localflavor.ro.forms.ROPostalCodeField` (*max_length=6, min_length=6, *args, **kwargs*)

Romanian postal code field.

7.48.2 Data

`localflavor.ro.ro_counties.COUNTIES_CHOICES` = (('AB', 'Alba'), ('AR', 'Arad'), ('AG', 'Arge'), ('BC', 'Bacău'), (

A list of Romanian counties as *choices* in a formfield.

7.49 Russia (ru)

7.49.1 Forms

Russian-specific forms helpers.

```
class localflavor.ru.forms.RUAlienPassportNumberField (max_length=None,
                                                    min_length=None,      *args,
                                                    **kwargs)
```

Russian alien's passport number format.

XX XXXXXXXX where X - any digit.

```
class localflavor.ru.forms.RUCountySelect (attrs=None)
```

A Select widget that uses a list of Russian Counties as its choices.

```
class localflavor.ru.forms.RUPassportNumberField (max_length=None,  min_length=None,
                                                    *args, **kwargs)
```

Russian internal passport number format.

XXXX XXXXXXXX where X - any digit.

```
class localflavor.ru.forms.RUPostalCodeField (max_length=None, min_length=None, *args,
                                                    **kwargs)
```

Russian Postal code field.

Format: XXXXXX, where X is any digit, and first digit is not zero.

```
class localflavor.ru.forms.RURegionSelect (attrs=None)
```

A Select widget that uses a list of Russian Regions as its choices.

7.49.2 Data

```
localflavor.ru.ru_regions.RU_COUNTY_CHOICES = (('Central Federal County', 'Central Federal County'), ('South I
http://ru.wikipedia.org/wiki/\_\_\_
```

```
localflavor.ru.ru_regions.RU_REGIONS_CHOICES = (('77', 'Moskva'), ('78', 'Saint-Peterburg'), ('50', 'Moskovskay
http://ru.wikipedia.org/wiki/\_\_\_
```

7.50 Sweden (se)

7.50.1 Forms

Swedish specific Form helpers.

```
class localflavor.se.forms.SECountySelect (attrs=None)
    A Select form widget that uses a list of the Swedish counties (län) as its choices.
```

The cleaned value is the official county code – see http://en.wikipedia.org/wiki/Counties_of_Sweden for a list.

```
class localflavor.se.forms.SEOrganisationNumberField (max_length=None,
                                                    min_length=None,  strip=True,
                                                    *args, **kwargs)
```

A form field that validates input as a Swedish organisation number (organisationsnummer).

It accepts the same input as SEPersonalIdentityField (for sole proprietorships (enskild firma). However, coordination numbers are not accepted.

It also accepts ordinary Swedish organisation numbers with the format NNNNNNNNNN.

The return value will be YYYYMMDDXXXX for sole proprietors, and NNNNNNNNNN for other organisations.

class `localflavor.se.forms.SEPersonalIdentityNumberField` (*coordination_number=True, *args, **kwargs*)

A form field that validates input as a Swedish personal identity number (personnummer).

The correct formats are YYYYMMDD-XXXX, YYYYMMDDXXXX, YYMMDD-XXXX, YYMMDDXXXX and YYMMDD+XXXX.

A + indicates that the person is older than 100 years, which will be taken into consideration when the date is validated.

The checksum will be calculated and checked. The birth date is checked to be a valid date.

By default, co-ordination numbers (samordningsnummer) will be accepted. To only allow real personal identity numbers, pass the keyword argument `coordination_number=False` to the constructor.

The cleaned value will always have the format YYYYMMDDXXXX.

class `localflavor.se.forms.SEPostalCodeField` (**args, **kwargs*)

A form field that validates input as a Swedish postal code (postnummer).

Valid codes consist of five digits (XXXXX). The number can optionally be formatted with a space after the third digit (XXX XX).

The cleaned value will never contain the space.

7.50.2 Utilities

`localflavor.se.utils.id_number_checksum` (*gd*)

Calculates a Swedish ID number checksum, using the “Luhn”-algorithm.

`localflavor.se.utils.validate_id_birthday` (*gd, fix_coordination_number_day=True*)

Validates the `birth_day` and returns the `datetime.date` object for the `birth_day`.

If the date is an invalid birth day, a `ValueError` will be raised.

7.50.3 Data

`localflavor.se.se_counties.COUNTY_CHOICES` = (('AB', 'Stockholm'), ('AC', 'Västerbotten'), ('BD', 'Norrbotten'),

An alphabetical list of Swedish counties, sorted by codes. http://en.wikipedia.org/wiki/Counties_of_Sweden

7.51 Singapore (sg)

New in version 1.1.

7.51.1 Forms

Singapore-specific Form helpers.

class `localflavor.sk.forms.SKPostalCodeField` (*max_length=None*, *min_length=None*, **args*, ***kwargs*)

A form field that validates its input as Slovak postal code.

Valid form is XXXXX or XXX XX, where X represents integer.

clean (*value*)

Validates the input and returns a string that contains only numbers.

Returns an empty string for empty values.

class `localflavor.sk.forms.SKRegionSelect` (*attrs=None*)

A select widget with list of Slovak regions as choices.

7.53.2 Data

`localflavor.sk.sk_districts.DISTRICT_CHOICES` = (('BB', 'Banska Bystrica'), ('BS', 'Banska Stiavnica'), ('BJ', 'Banskobystricky kraj'), ('BR', 'Bratislava'), ('BT', 'Brezno'), ('BN', 'Banska Nová Ves nad Váhom'), ('BV', 'Banská Štiavnica'), ('BA', 'Bratislava'), ('KE', 'Košice'), ('KO', 'Košice'), ('ZL', 'Zilina'), ('PV', 'Prešov'), ('PP', 'Prešov'), ('PU', 'Prešov'), ('PC', 'Prešov'), ('PR', 'Prešov'), ('PS', 'Prešov'), ('PT', 'Prešov'), ('PB', 'Prešov'), ('PD', 'Prešov'), ('PE', 'Prešov'), ('PF', 'Prešov'), ('PG', 'Prešov'), ('PH', 'Prešov'), ('PI', 'Prešov'), ('PJ', 'Prešov'), ('PK', 'Prešov'), ('PL', 'Prešov'), ('PM', 'Prešov'), ('PN', 'Prešov'), ('PO', 'Prešov'), ('PP', 'Prešov'), ('PQ', 'Prešov'), ('PR', 'Prešov'), ('PS', 'Prešov'), ('PT', 'Prešov'), ('PU', 'Prešov'), ('PV', 'Prešov'), ('PW', 'Prešov'), ('PX', 'Prešov'), ('PY', 'Prešov'), ('PZ', 'Prešov'), ('QA', 'Prešov'), ('QB', 'Prešov'), ('QC', 'Prešov'), ('QD', 'Prešov'), ('QE', 'Prešov'), ('QF', 'Prešov'), ('QG', 'Prešov'), ('QH', 'Prešov'), ('QI', 'Prešov'), ('QJ', 'Prešov'), ('QK', 'Prešov'), ('QL', 'Prešov'), ('QM', 'Prešov'), ('QN', 'Prešov'), ('QO', 'Prešov'), ('QP', 'Prešov'), ('QQ', 'Prešov'), ('QR', 'Prešov'), ('QS', 'Prešov'), ('QT', 'Prešov'), ('QU', 'Prešov'), ('QV', 'Prešov'), ('QW', 'Prešov'), ('QX', 'Prešov'), ('QY', 'Prešov'), ('QZ', 'Prešov'), ('RA', 'Prešov'), ('RB', 'Prešov'), ('RC', 'Prešov'), ('RD', 'Prešov'), ('RE', 'Prešov'), ('RF', 'Prešov'), ('RG', 'Prešov'), ('RH', 'Prešov'), ('RI', 'Prešov'), ('RJ', 'Prešov'), ('RK', 'Prešov'), ('RL', 'Prešov'), ('RM', 'Prešov'), ('RN', 'Prešov'), ('RO', 'Prešov'), ('RP', 'Prešov'), ('RQ', 'Prešov'), ('RR', 'Prešov'), ('RS', 'Prešov'), ('RT', 'Prešov'), ('RU', 'Prešov'), ('RV', 'Prešov'), ('RW', 'Prešov'), ('RX', 'Prešov'), ('RY', 'Prešov'), ('RZ', 'Prešov'), ('SA', 'Prešov'), ('SB', 'Prešov'), ('SC', 'Prešov'), ('SD', 'Prešov'), ('SE', 'Prešov'), ('SF', 'Prešov'), ('SG', 'Prešov'), ('SH', 'Prešov'), ('SI', 'Prešov'), ('SJ', 'Prešov'), ('SK', 'Prešov'), ('SL', 'Prešov'), ('SM', 'Prešov'), ('SN', 'Prešov'), ('SO', 'Prešov'), ('SP', 'Prešov'), ('SQ', 'Prešov'), ('SR', 'Prešov'), ('SS', 'Prešov'), ('ST', 'Prešov'), ('SU', 'Prešov'), ('SV', 'Prešov'), ('SW', 'Prešov'), ('SX', 'Prešov'), ('SY', 'Prešov'), ('SZ', 'Prešov'), ('TA', 'Prešov'), ('TB', 'Prešov'), ('TC', 'Prešov'), ('TD', 'Prešov'), ('TE', 'Prešov'), ('TF', 'Prešov'), ('TG', 'Prešov'), ('TH', 'Prešov'), ('TI', 'Prešov'), ('TJ', 'Prešov'), ('TK', 'Prešov'), ('TL', 'Prešov'), ('TM', 'Prešov'), ('TN', 'Prešov'), ('TO', 'Prešov'), ('TP', 'Prešov'), ('TQ', 'Prešov'), ('TR', 'Prešov'), ('TS', 'Prešov'), ('TT', 'Prešov'), ('TU', 'Prešov'), ('TV', 'Prešov'), ('TW', 'Prešov'), ('TX', 'Prešov'), ('TY', 'Prešov'), ('TZ', 'Prešov'), ('UA', 'Prešov'), ('UB', 'Prešov'), ('UC', 'Prešov'), ('UD', 'Prešov'), ('UE', 'Prešov'), ('UF', 'Prešov'), ('UG', 'Prešov'), ('UH', 'Prešov'), ('UI', 'Prešov'), ('UJ', 'Prešov'), ('UK', 'Prešov'), ('UL', 'Prešov'), ('UM', 'Prešov'), ('UN', 'Prešov'), ('UO', 'Prešov'), ('UP', 'Prešov'), ('UQ', 'Prešov'), ('UR', 'Prešov'), ('US', 'Prešov'), ('UT', 'Prešov'), ('UU', 'Prešov'), ('UV', 'Prešov'), ('UW', 'Prešov'), ('UX', 'Prešov'), ('UY', 'Prešov'), ('UZ', 'Prešov'), ('VA', 'Prešov'), ('VB', 'Prešov'), ('VC', 'Prešov'), ('VD', 'Prešov'), ('VE', 'Prešov'), ('VF', 'Prešov'), ('VG', 'Prešov'), ('VH', 'Prešov'), ('VI', 'Prešov'), ('VJ', 'Prešov'), ('VK', 'Prešov'), ('VL', 'Prešov'), ('VM', 'Prešov'), ('VN', 'Prešov'), ('VO', 'Prešov'), ('VP', 'Prešov'), ('VQ', 'Prešov'), ('VR', 'Prešov'), ('VS', 'Prešov'), ('VT', 'Prešov'), ('VU', 'Prešov'), ('VV', 'Prešov'), ('VW', 'Prešov'), ('VX', 'Prešov'), ('VY', 'Prešov'), ('VZ', 'Prešov'), ('WA', 'Prešov'), ('WB', 'Prešov'), ('WC', 'Prešov'), ('WD', 'Prešov'), ('WE', 'Prešov'), ('WF', 'Prešov'), ('WG', 'Prešov'), ('WH', 'Prešov'), ('WI', 'Prešov'), ('WJ', 'Prešov'), ('WK', 'Prešov'), ('WL', 'Prešov'), ('WM', 'Prešov'), ('WN', 'Prešov'), ('WO', 'Prešov'), ('WP', 'Prešov'), ('WQ', 'Prešov'), ('WR', 'Prešov'), ('WS', 'Prešov'), ('WT', 'Prešov'), ('WU', 'Prešov'), ('WV', 'Prešov'), ('WW', 'Prešov'), ('WX', 'Prešov'), ('WY', 'Prešov'), ('WZ', 'Prešov'), ('XA', 'Prešov'), ('XB', 'Prešov'), ('XC', 'Prešov'), ('XD', 'Prešov'), ('XE', 'Prešov'), ('XF', 'Prešov'), ('XG', 'Prešov'), ('XH', 'Prešov'), ('XI', 'Prešov'), ('XJ', 'Prešov'), ('XK', 'Prešov'), ('XL', 'Prešov'), ('XM', 'Prešov'), ('XN', 'Prešov'), ('XO', 'Prešov'), ('XP', 'Prešov'), ('XQ', 'Prešov'), ('XR', 'Prešov'), ('XS', 'Prešov'), ('XT', 'Prešov'), ('XU', 'Prešov'), ('XV', 'Prešov'), ('XW', 'Prešov'), ('XX', 'Prešov'), ('XY', 'Prešov'), ('XZ', 'Prešov'), ('YA', 'Prešov'), ('YB', 'Prešov'), ('YC', 'Prešov'), ('YD', 'Prešov'), ('YE', 'Prešov'), ('YF', 'Prešov'), ('YG', 'Prešov'), ('YH', 'Prešov'), ('YI', 'Prešov'), ('YJ', 'Prešov'), ('YK', 'Prešov'), ('YL', 'Prešov'), ('YM', 'Prešov'), ('YN', 'Prešov'), ('YO', 'Prešov'), ('YP', 'Prešov'), ('YQ', 'Prešov'), ('YR', 'Prešov'), ('YS', 'Prešov'), ('YT', 'Prešov'), ('YU', 'Prešov'), ('YV', 'Prešov'), ('YW', 'Prešov'), ('YX', 'Prešov'), ('YY', 'Prešov'), ('YZ', 'Prešov'), ('ZA', 'Prešov'), ('ZB', 'Prešov'), ('ZC', 'Prešov'), ('ZD', 'Prešov'), ('ZE', 'Prešov'), ('ZF', 'Prešov'), ('ZG', 'Prešov'), ('ZH', 'Prešov'), ('ZI', 'Prešov'), ('ZJ', 'Prešov'), ('ZK', 'Prešov'), ('ZL', 'Prešov'), ('ZM', 'Prešov'), ('ZN', 'Prešov'), ('ZO', 'Prešov'), ('ZP', 'Prešov'), ('ZQ', 'Prešov'), ('ZR', 'Prešov'), ('ZS', 'Prešov'), ('ZT', 'Prešov'), ('ZU', 'Prešov'), ('ZV', 'Prešov'), ('ZW', 'Prešov'), ('ZX', 'Prešov'), ('ZY', 'Prešov'), ('ZZ', 'Prešov'))

http://sk.wikipedia.org/wiki/Administrat%C3%ADvne_%C4%8Dlenenie_Slovenska

`localflavor.sk.sk_regions.REGION_CHOICES` = (('BB', 'Banska Bystrica region'), ('BA', 'Bratislava region'), ('KE', 'Košice region'), ('KO', 'Košice region'), ('ZL', 'Zilina region'), ('PV', 'Prešov region'), ('PP', 'Prešov region'), ('PU', 'Prešov region'), ('PC', 'Prešov region'), ('PR', 'Prešov region'), ('PS', 'Prešov region'), ('PT', 'Prešov region'), ('PB', 'Prešov region'), ('PD', 'Prešov region'), ('PE', 'Prešov region'), ('PF', 'Prešov region'), ('PG', 'Prešov region'), ('PH', 'Prešov region'), ('PI', 'Prešov region'), ('PJ', 'Prešov region'), ('PK', 'Prešov region'), ('PL', 'Prešov region'), ('PM', 'Prešov region'), ('PN', 'Prešov region'), ('PO', 'Prešov region'), ('PP', 'Prešov region'), ('PQ', 'Prešov region'), ('PR', 'Prešov region'), ('PS', 'Prešov region'), ('PT', 'Prešov region'), ('PU', 'Prešov region'), ('PV', 'Prešov region'), ('PW', 'Prešov region'), ('PX', 'Prešov region'), ('PY', 'Prešov region'), ('PZ', 'Prešov region'), ('QA', 'Prešov region'), ('QB', 'Prešov region'), ('QC', 'Prešov region'), ('QD', 'Prešov region'), ('QE', 'Prešov region'), ('QF', 'Prešov region'), ('QG', 'Prešov region'), ('QH', 'Prešov region'), ('QI', 'Prešov region'), ('QJ', 'Prešov region'), ('QK', 'Prešov region'), ('QL', 'Prešov region'), ('QM', 'Prešov region'), ('QN', 'Prešov region'), ('QO', 'Prešov region'), ('QP', 'Prešov region'), ('QQ', 'Prešov region'), ('QR', 'Prešov region'), ('QS', 'Prešov region'), ('QT', 'Prešov region'), ('QU', 'Prešov region'), ('QV', 'Prešov region'), ('QW', 'Prešov region'), ('QX', 'Prešov region'), ('QY', 'Prešov region'), ('QZ', 'Prešov region'), ('RA', 'Prešov region'), ('RB', 'Prešov region'), ('RC', 'Prešov region'), ('RD', 'Prešov region'), ('RE', 'Prešov region'), ('RF', 'Prešov region'), ('RG', 'Prešov region'), ('RH', 'Prešov region'), ('RI', 'Prešov region'), ('RJ', 'Prešov region'), ('RK', 'Prešov region'), ('RL', 'Prešov region'), ('RM', 'Prešov region'), ('RN', 'Prešov region'), ('RO', 'Prešov region'), ('RP', 'Prešov region'), ('RQ', 'Prešov region'), ('RR', 'Prešov region'), ('RS', 'Prešov region'), ('RT', 'Prešov region'), ('RU', 'Prešov region'), ('RV', 'Prešov region'), ('RW', 'Prešov region'), ('RX', 'Prešov region'), ('RY', 'Prešov region'), ('RZ', 'Prešov region'), ('SA', 'Prešov region'), ('SB', 'Prešov region'), ('SC', 'Prešov region'), ('SD', 'Prešov region'), ('SE', 'Prešov region'), ('SF', 'Prešov region'), ('SG', 'Prešov region'), ('SH', 'Prešov region'), ('SI', 'Prešov region'), ('SJ', 'Prešov region'), ('SK', 'Prešov region'), ('SL', 'Prešov region'), ('SM', 'Prešov region'), ('SN', 'Prešov region'), ('SO', 'Prešov region'), ('SP', 'Prešov region'), ('SQ', 'Prešov region'), ('SR', 'Prešov region'), ('SS', 'Prešov region'), ('ST', 'Prešov region'), ('SU', 'Prešov region'), ('SV', 'Prešov region'), ('SW', 'Prešov region'), ('SX', 'Prešov region'), ('SY', 'Prešov region'), ('SZ', 'Prešov region'), ('TA', 'Prešov region'), ('TB', 'Prešov region'), ('TC', 'Prešov region'), ('TD', 'Prešov region'), ('TE', 'Prešov region'), ('TF', 'Prešov region'), ('TG', 'Prešov region'), ('TH', 'Prešov region'), ('TI', 'Prešov region'), ('TJ', 'Prešov region'), ('TK', 'Prešov region'), ('TL', 'Prešov region'), ('TM', 'Prešov region'), ('TN', 'Prešov region'), ('TO', 'Prešov region'), ('TP', 'Prešov region'), ('TQ', 'Prešov region'), ('TR', 'Prešov region'), ('TS', 'Prešov region'), ('TT', 'Prešov region'), ('TU', 'Prešov region'), ('TV', 'Prešov region'), ('TW', 'Prešov region'), ('TX', 'Prešov region'), ('TY', 'Prešov region'), ('TZ', 'Prešov region'), ('UA', 'Prešov region'), ('UB', 'Prešov region'), ('UC', 'Prešov region'), ('UD', 'Prešov region'), ('UE', 'Prešov region'), ('UF', 'Prešov region'), ('UG', 'Prešov region'), ('UH', 'Prešov region'), ('UI', 'Prešov region'), ('UJ', 'Prešov region'), ('UK', 'Prešov region'), ('UL', 'Prešov region'), ('UM', 'Prešov region'), ('UN', 'Prešov region'), ('UO', 'Prešov region'), ('UP', 'Prešov region'), ('UQ', 'Prešov region'), ('UR', 'Prešov region'), ('US', 'Prešov region'), ('UT', 'Prešov region'), ('UU', 'Prešov region'), ('UV', 'Prešov region'), ('UW', 'Prešov region'), ('UX', 'Prešov region'), ('UY', 'Prešov region'), ('UZ', 'Prešov region'), ('VA', 'Prešov region'), ('VB', 'Prešov region'), ('VC', 'Prešov region'), ('VD', 'Prešov region'), ('VE', 'Prešov region'), ('VF', 'Prešov region'), ('VG', 'Prešov region'), ('VH', 'Prešov region'), ('VI', 'Prešov region'), ('VJ', 'Prešov region'), ('VK', 'Prešov region'), ('VL', 'Prešov region'), ('VM', 'Prešov region'), ('VN', 'Prešov region'), ('VO', 'Prešov region'), ('VP', 'Prešov region'), ('VQ', 'Prešov region'), ('VR', 'Prešov region'), ('VS', 'Prešov region'), ('VT', 'Prešov region'), ('VU', 'Prešov region'), ('VV', 'Prešov region'), ('VW', 'Prešov region'), ('VX', 'Prešov region'), ('VY', 'Prešov region'), ('VZ', 'Prešov region'), ('WA', 'Prešov region'), ('WB', 'Prešov region'), ('WC', 'Prešov region'), ('WD', 'Prešov region'), ('WE', 'Prešov region'), ('WF', 'Prešov region'), ('WG', 'Prešov region'), ('WH', 'Prešov region'), ('WI', 'Prešov region'), ('WJ', 'Prešov region'), ('WK', 'Prešov region'), ('WL', 'Prešov region'), ('WM', 'Prešov region'), ('WN', 'Prešov region'), ('WO', 'Prešov region'), ('WP', 'Prešov region'), ('WQ', 'Prešov region'), ('WR', 'Prešov region'), ('WS', 'Prešov region'), ('WT', 'Prešov region'), ('WU', 'Prešov region'), ('WV', 'Prešov region'), ('WW', 'Prešov region'), ('WX', 'Prešov region'), ('WY', 'Prešov region'), ('WZ', 'Prešov region'), ('XA', 'Prešov region'), ('XB', 'Prešov region'), ('XC', 'Prešov region'), ('XD', 'Prešov region'), ('XE', 'Prešov region'), ('XF', 'Prešov region'), ('XG', 'Prešov region'), ('XH', 'Prešov region'), ('XI', 'Prešov region'), ('XJ', 'Prešov region'), ('XK', 'Prešov region'), ('XL', 'Prešov region'), ('XM', 'Prešov region'), ('XN', 'Prešov region'), ('XO', 'Prešov region'), ('XP', 'Prešov region'), ('XQ', 'Prešov region'), ('XR', 'Prešov region'), ('XS', 'Prešov region'), ('XT', 'Prešov region'), ('XU', 'Prešov region'), ('XV', 'Prešov region'), ('XW', 'Prešov region'), ('XX', 'Prešov region'), ('XY', 'Prešov region'), ('XZ', 'Prešov region'), ('YA', 'Prešov region'), ('YB', 'Prešov region'), ('YC', 'Prešov region'), ('YD', 'Prešov region'), ('YE', 'Prešov region'), ('YF', 'Prešov region'), ('YG', 'Prešov region'), ('YH', 'Prešov region'), ('YI', 'Prešov region'), ('YJ', 'Prešov region'), ('YK', 'Prešov region'), ('YL', 'Prešov region'), ('YM', 'Prešov region'), ('YN', 'Prešov region'), ('YO', 'Prešov region'), ('YP', 'Prešov region'), ('YQ', 'Prešov region'), ('YR', 'Prešov region'), ('YS', 'Prešov region'), ('YT', 'Prešov region'), ('YU', 'Prešov region'), ('YV', 'Prešov region'), ('YW', 'Prešov region'), ('YX', 'Prešov region'), ('YY', 'Prešov region'), ('YZ', 'Prešov region'), ('ZA', 'Prešov region'), ('ZB', 'Prešov region'), ('ZC', 'Prešov region'), ('ZD', 'Prešov region'), ('ZE', 'Prešov region'), ('ZF', 'Prešov region'), ('ZG', 'Prešov region'), ('ZH', 'Prešov region'), ('ZI', 'Prešov region'), ('ZJ', 'Prešov region'), ('ZK', 'Prešov region'), ('ZL', 'Prešov region'), ('ZM', 'Prešov region'), ('ZN', 'Prešov region'), ('ZO', 'Prešov region'), ('ZP', 'Prešov region'), ('ZQ', 'Prešov region'), ('ZR', 'Prešov region'), ('ZS', 'Prešov region'), ('ZT', 'Prešov region'), ('ZU', 'Prešov region'), ('ZV', 'Prešov region'), ('ZW', 'Prešov region'), ('ZX', 'Prešov region'), ('ZY', 'Prešov region'), ('ZZ', 'Prešov region'))

http://sk.wikipedia.org/wiki/Administrat%C3%ADvne_%C4%8Dlenenie_Slovenska

7.54 Tunisia (tn)

Tunisia-specific Form helpers.

class `localflavor.tn.forms.TNGovernorateSelect` (*attrs=None*)

A Select widget that uses a list of the Tunisian governorates as its choices.

`localflavor.tn.tn_governorates.GOVERNORATE_CHOICES` = (('ariana', 'Ariana'), ('beja', 'Beja'), ('ben arous', 'Ben Arous'), ('gabès', 'Gabès'), ('gafsa', 'Gafsa'), ('jendouba', 'Jendouba'), ('kairouan', 'Kairouan'), ('kasserine', 'Kasserine'), ('medenine', 'Medenine'), ('monastir', 'Monastir'), ('nabeul', 'Nabeul'), ('nord', 'Nord'), ('nord_ouest', 'Nord-Ouest'), ('sfax', 'Sfax'), ('siliana', 'Siliana'), ('tataouine', 'Tataouine'), ('tunis', 'Tunis'), ('zaghouane', 'Zaghouane'))

All the 24 Tunisian Governorates http://en.wikipedia.org/wiki/Governorates_of_Tunisia

7.55 Turkey (tr)

7.55.1 Forms

class `localflavor.tr.forms.TRIdentificationNumberField` (*required=True*, *widget=None*, *label=None*, *initial=None*, *help_text=''*, *error_messages=None*, *show_hidden_initial=False*, *validators=[]*, *localize=False*, *disabled=False*, *label_suffix=None*)

A Turkey Identification Number number.

See: http://tr.wikipedia.org/wiki/T%C3%BCrkiye_Cumhuriyeti_Kimlik_Numaras%C4%B1

Checks the following rules to determine whether the number is valid:

- The number is 11-digits.
- First digit is not 0.
- Conforms to the following two formula: $(\text{sum}(1\text{st}, 3\text{rd}, 5\text{th}, 7\text{th}, 9\text{th}) * 7 - \text{sum}(2\text{nd}, 4\text{th}, 6\text{th}, 8\text{th})) \% 10 = 10\text{th digit}$ $\text{sum}(1\text{st to } 10\text{th}) \% 10 = 11\text{th digit}$

```
class localflavor.tr.forms.TRPhoneNumberField(max_length=None, min_length=None,
                                              strip=True, *args, **kwargs)
```

A form field that validates input as a Turkish phone number.

The correct format is 0xxx xxx xxxx. +90xxx xxx xxxx and inputs without spaces also validates. The result is normalized to xxx xxx xxxx format.

```
class localflavor.tr.forms.TRPostalCodeField(max_length=5, min_length=5, *args,
                                              **kwargs)
```

A form field that validates input as a Turkish zip code.

Valid codes consist of five digits.

```
class localflavor.tr.forms.TRProvinceSelect(attrs=None)
```

A Select widget that uses a list of provinces in Turkey as its choices.

7.55.2 Data

```
localflavor.tr.tr_provinces.PROVINCE_CHOICES = (('01', 'Adana'), ('02', 'Adiyaman'), ('03', 'Afyonkarahisar'), (
```

A list of Turkish provinces

7.56 United States of America (us)

7.56.1 Forms

USA-specific Form helpers.

```
class localflavor.us.forms.USPSSelect(attrs=None)
```

A Select widget that uses a list of US Postal Service codes as its choices.

Note: If you are looking for a form field that validates U.S. ZIP codes please use `USZipCodeField`.

```
class localflavor.us.forms.USPhoneNumberField(max_length=None, min_length=None,
                                              strip=True, *args, **kwargs)
```

A form field that validates input as a U.S. phone number.

```
class localflavor.us.forms.USSocialSecurityNumberField(max_length=None,
                                                         min_length=None, strip=True,
                                                         *args, **kwargs)
```

A United States Social Security number.

Checks the following rules to determine whether the number is valid:

- Conforms to the XXX-XX-XXXX format.
- No group consists entirely of zeroes.
- The leading group is not “666” (block “666” will never be allocated).
- The number is not in the promotional block 987-65-4320 through 987-65-4329, which are permanently invalid.
- The number is not one known to be invalid due to otherwise widespread promotional use or distribution (e.g., the Woolworth’s number or the 1962 promotional number).

New in version 1.1.

class `localflavor.us.forms.USStateField` (*required=True, widget=None, label=None, initial=None, help_text='', error_messages=None, show_hidden_initial=False, validators=[], localize=False, disabled=False, label_suffix=None*)

A form field that validates its input is a U.S. state name or abbreviation.

It normalizes the input to the standard two-letter postal service abbreviation for the given state.

class `localflavor.us.forms.USStateSelect` (*attrs=None*)

A Select widget that uses a list of U.S. states/territories as its choices.

class `localflavor.us.forms.USZipCodeField` (*max_length=None, min_length=None, *args, **kwargs*)

A form field that validates input as a U.S. ZIP code.

Valid formats are `XXXXXX` or `XXXXX-XXXX`.

Note: If you are looking for a form field with a list of U.S. Postal Service locations please use `USPSSelect`.

New in version 1.1.

Whitespace around the ZIP code is accepted and automatically trimmed.

7.56.2 Models

class `localflavor.us.models.PhoneNumberField` (**args, **kwargs*)

A `CharField` that checks that the value is a valid U.S.A.-style phone number.

(in the format `XXX-XXX-XXXX`).

class `localflavor.us.models.USPostalCodeField` (**args, **kwargs*)

A model field that stores the two-letter U.S. Postal Service abbreviation in the database.

Forms represent it as a `USPSSelect` field.

Note: If you are looking for a model field that validates U.S. ZIP codes please use `USZipCodeField`.

class `localflavor.us.models.USSocialSecurityNumberField` (**args, **kwargs*)

A model field that stores the security number in the format `XXX-XX-XXXX`.

Forms represent it as `forms.USSocialSecurityNumberField` field.

New in version 1.1.

class `localflavor.us.models.USStateField` (**args, **kwargs*)

A model field that stores the two-letter U.S. state abbreviation in the database.

Forms represent it as a `forms.USStateField` field.

class `localflavor.us.models.USZipCodeField` (**args, **kwargs*)

A model field that stores the U.S. ZIP code in the database.

Forms represent it as a `USZipCodeField` field.

Note: If you are looking for a model field with a list of U.S. Postal Service locations please use `USPostalCodeField`.

New in version 1.1.

7.56.3 Data

`localflavor.us.us_states.CONTIGUOUS_STATES = (('AL', 'Alabama'), ('AZ', 'Arizona'), ('AR', 'Arkansas'), ('CA', 'California'), ('CO', 'Colorado'), ('CT', 'Connecticut'), ('DC', 'District of Columbia'), ('DE', 'Delaware'), ('FL', 'Florida'), ('GA', 'Georgia'), ('HI', 'Hawaii'), ('IA', 'Iowa'), ('IL', 'Illinois'), ('IN', 'Indiana'), ('KS', 'Kansas'), ('KY', 'Kentucky'), ('LA', 'Louisiana'), ('MA', 'Massachusetts'), ('MD', 'Maryland'), ('ME', 'Maine'), ('MI', 'Michigan'), ('MN', 'Minnesota'), ('MO', 'Missouri'), ('MS', 'Mississippi'), ('MT', 'Montana'), ('NC', 'North Carolina'), ('ND', 'North Dakota'), ('OH', 'Ohio'), ('OK', 'Oklahoma'), ('OR', 'Oregon'), ('PA', 'Pennsylvania'), ('RI', 'Rhode Island'), ('SC', 'South Carolina'), ('SD', 'South Dakota'), ('TN', 'Tennessee'), ('TX', 'Texas'), ('UT', 'Utah'), ('VA', 'Virginia'), ('VT', 'Vermont'), ('WA', 'Washington'), ('WI', 'Wisconsin'), ('WV', 'West Virginia'), ('WY', 'Wyoming'))`

The 48 contiguous states, plus the District of Columbia.

`localflavor.us.us_states.NON_CONTIGUOUS_STATES = (('AK', 'Alaska'), ('HI', 'Hawaii'))`

Non contiguous states (Not connected to mainland USA)

`localflavor.us.us_states.US_TERRITORIES = (('AS', 'American Samoa'), ('GU', 'Guam'), ('MP', 'Northern Mariana Islands'))`

Non-state territories.

`localflavor.us.us_states.ARMED_FORCES_STATES = (('AA', 'Armed Forces Americas'), ('AE', 'Armed Forces Europe'))`

Military postal “states”. Note that ‘AE’ actually encompasses Europe, Canada, Africa and the Middle East.

`localflavor.us.us_states.COFA_STATES = (('FM', 'Federated States of Micronesia'), ('MH', 'Marshall Islands'), ('PW', 'Palau'))`

Non-US locations serviced by USPS (under Compact of Free Association).

`localflavor.us.us_states.OBSOLETE_STATES = (('CM', 'Commonwealth of the Northern Mariana Islands'), ('CZ', 'Czech Republic'))`

Obsolete abbreviations (no longer US territories/USPS service, or code changed).

`localflavor.us.us_states.US_STATES = CONTIGUOUS_STATES + NON_CONTIGUOUS_STATES`

All US states.

This tuple is lazily generated and may not work as expected in all cases due to tuple optimizations in the Python interpreter which do not account for lazily generated tuples. For example:

```
US_STATES + ('XX', _('Select a State'))
```

should work as expected, but:

```
('XX', _('Select a State')) + US_STATES
```

may throw:

```
TypeError: can only concatenate tuple (not "proxy") to tuple
```

due to a Python optimization that causes the concatenation to occur before `US_STATES` has been lazily generated. To work around these issues, you can use a slice index (`[:]`) to force the generation of `US_STATES` before any other operations are processed by the Python interpreter:

```
('XX', _('Select a State')) + US_STATES[:]
```

`localflavor.us.us_states.STATE_CHOICES = CONTIGUOUS_STATES + NON_CONTIGUOUS_STATES + US_TERRITORIES`

All US states and territories plus DC and military mail.

This tuple is lazily generated and may not work as expected in all cases due to tuple optimizations in the Python interpreter which do not account for lazily generated tuples. For example:

```
STATE_CHOICES + ('XX', _('Select a State'))
```

should work as expected, but:

```
('XX', _('Select a State')) + STATE_CHOICES
```

may throw:

```
TypeError: can only concatenate tuple (not "proxy") to tuple
```

due to a Python optimization that causes the concatenation to occur before `STATE_CHOICES` has been lazily generated. To work around these issues, you can use a slice index (`[:]`) to force the generation of `STATE_CHOICES` before any other operations are processed by the Python interpreter:

```
('XX', _('Select a State')) + STATE_CHOICES[:]
```

```
localflavor.us.us_states.USPS_CHOICES = CONTIGUOUS_STATES + NON_CONTIGUOUS_STATES + US_TERMINALS
All US Postal Service locations.
```

This tuple is lazily generated and may not work as expected in all cases due to tuple optimizations in the Python interpreter which do not account for lazily generated tuples. For example:

```
USPS_CHOICES + ('XX', _('Select a State'))
```

should work as expected, but:

```
('XX', _('Select a State')) + USPS_CHOICES
```

may throw:

```
TypeError: can only concatenate tuple (not "proxy") to tuple
```

due to a Python optimization that causes the concatenation to occur before `USPS_CHOICES` has been lazily generated. To work around these issues, you can use a slice index (`[:]`) to force the generation of `USPS_CHOICES` before any other operations are processed by the Python interpreter:

```
('XX', _('Select a State')) + USPS_CHOICES[:]
```

```
localflavor.us.us_states.STATES_NORMALIZED = {'ark': 'AR', 'hi': 'HI', 'idaho': 'ID', 'missouri': 'MO', 'us virgin islands': 'VI'}
Normalized versions of state names
```

7.57 Uruguay (uy)

7.57.1 Forms

UY-specific form helpers.

class `localflavor.uy.forms.UYCIField` (**args*, ***kwargs*)
A field that validates Uruguayan ‘Cedula de identidad’ (CI) numbers.

clean (*value*)

Validates format and validation digit.

The official format is `[X.]XXX.XXX-X` but usually dots and/or slash are omitted so, when validating, those characters are ignored if found in the correct place. The three typically used formats are supported: `[X]XXXXXXXX`, `[X]XXXXXX-X` and `[X.]XXX.XXX-X`.

class `localflavor.uy.forms.UYDepartmentSelect` (*attrs=None*)
A Select widget that uses a list of Uruguayan departments as its choices.

7.57.2 Utilities

`localflavor.uy.util.get_validation_digit` (*number*)
Calculates the validation digit for the given number.

7.57.3 Data

`localflavor.uy.uy_departments.DEPARTMENT_CHOICES = (('G', 'Artigas'), ('A', 'Canelones'), ('E', 'Cerro Largo'),`
A list of Uruguayan departments as *choices* in a formfield.

7.58 South Africa (za)

7.58.1 Forms

South Africa-specific Form helpers.

class `localflavor.za.forms.ZAIDField` (*max_length=None, min_length=None, strip=True, *args,*
***kwargs*)

A form field for South African ID numbers.

The checksum is validated using the Luhn checksum, and uses a simplistic (read: not entirely accurate) check for the birthdate

class `localflavor.za.forms.ZAPostCodeField` (*max_length=None, min_length=None, *args,*
***kwargs*)

A form field that validates input as a South African postcode.

Valid postcodes must have four digits.

class `localflavor.za.forms.ZAProvinceSelect` (*attrs=None*)

A Select widget that uses a list of South African Provinces as its choices.

7.58.2 Data

`localflavor.za.za_provinces.PROVINCE_CHOICES = (('EC', 'Eastern Cape'), ('FS', 'Free State'), ('GP', 'Gauteng'),`
A list of South African provinces as *choices* in a formfield.

7.59 Generic helpers

7.59.1 Forms

class `localflavor.generic.forms.BICFormField` (**args, **kwargs*)

A BIC consists of 8 (BIC8) or 11 (BIC11) alphanumeric characters.

BICs are also known as SWIFT-BIC, BIC code, SWIFT ID, SWIFT code or ISO 9362.

https://en.wikipedia.org/wiki/ISO_9362

New in version 1.1.

class `localflavor.generic.forms.DateField` (*input_formats=None, *args, **kwargs*)

A date input field which uses non-US date input formats by default.

class `localflavor.generic.forms.DateTimeField` (*input_formats=None, *args, **kwargs*)

A date and time input field which uses non-US date and time input formats by default.

class `localflavor.generic.forms.IBANFormField` (*use_nordea_extensions=False, include_countries=None, *args, **kwargs*)

An IBAN consists of up to 34 alphanumeric characters.

To limit validation to specific countries, set the ‘include_countries’ argument with a tuple or list of ISO 3166-1 alpha-2 codes. For example, `include_countries=(‘NL’, ‘BE’, ‘LU’)`.

A list of countries that use IBANs as part of SEPA is included for convenience. To use this feature, set `include_countries=IBAN_SEPA_COUNTRIES` as an argument to the field.

Example:

```
from django import forms
from localflavor.generic.forms import IBANFormField
from localflavor.generic.countries.sepa import IBAN_SEPA_COUNTRIES

class MyForm(forms.Form):
    iban = IBANFormField(include_countries=IBAN_SEPA_COUNTRIES)
```

In addition to validating official IBANs, this field can optionally validate unofficial IBANs that have been catalogued by Nordea by setting the `use_nordea_extensions` argument to True.

https://en.wikipedia.org/wiki/International_Bank_Account_Number

New in version 1.1.

prepare_value (*value*)

The display format for IBAN has a space every 4 characters.

class `localflavor.generic.forms.SplitDateTimeField` (*input_date_formats=None, input_time_formats=None, *args, **kwargs*)

Split date and time input fields which use non-US date and time input formats by default.

7.59.2 Models

class `localflavor.generic.models.BICField` (**args, **kwargs*)

A BIC consists of 8 (BIC8) or 11 (BIC11) alphanumeric characters.

BICs are also known as SWIFT-BIC, BIC code, SWIFT ID, SWIFT code or ISO 9362.

https://en.wikipedia.org/wiki/ISO_9362

New in version 1.1.

class `localflavor.generic.models.IBANField` (**args, **kwargs*)

An IBAN consists of up to 34 alphanumeric characters.

To limit validation to specific countries, set the ‘include_countries’ argument with a tuple or list of ISO 3166-1 alpha-2 codes. For example, `include_countries=(‘NL’, ‘BE’, ‘LU’)`.

A list of countries that use IBANs as part of SEPA is included for convenience. To use this feature, set `include_countries=IBAN_SEPA_COUNTRIES` as an argument to the field.

Example:

```
from django.db import models
from localflavor.generic.models import IBANField
from localflavor.generic.countries.sepa import IBAN_SEPA_COUNTRIES

class MyModel(models.Model):
    iban = IBANField(include_countries=IBAN_SEPA_COUNTRIES)
```

In addition to validating official IBANs, this field can optionally validate unofficial IBANs that have been catalogued by Nordea by setting the `use_nordea_extensions` argument to True.

https://en.wikipedia.org/wiki/International_Bank_Account_Number

New in version 1.1.

7.59.3 Checksums

Common checksum routines.

`localflavor.generic.checksums.luhn(candidate)`

Checks a candidate number for validity according to the Luhn algorithm.

Luhn algorithm is used in validation of, for example, credit cards. Both numeric and string candidates are accepted.

`localflavor.generic.checksums.ean(candidate)`

Checks a candidate number for validity according to the EAN checksum calculation.

Note that this validator does not enforce any length checks (usually 13 or 8).

[http://en.wikipedia.org/wiki/International_Article_Number_\(EAN\)](http://en.wikipedia.org/wiki/International_Article_Number_(EAN))

New in version 1.1.

7.59.4 Validators

class `localflavor.generic.validators.BICValidator`

A validator for SWIFT Business Identifier Codes (ISO 9362:2009).

Validation is based on the BIC structure found on wikipedia.

https://en.wikipedia.org/wiki/ISO_9362#Structure

deconstruct (*obj*)

Returns a 3-tuple of class import path, positional arguments, and keyword arguments.

class `localflavor.generic.validators.EANValidator` (*strip_nondigits=False, sage=None*) *mes-*

A generic validator for EAN like codes with the last digit being the checksum.

[http://en.wikipedia.org/wiki/International_Article_Number_\(EAN\)](http://en.wikipedia.org/wiki/International_Article_Number_(EAN))

deconstruct (*obj*)

Returns a 3-tuple of class import path, positional arguments, and keyword arguments.

class `localflavor.generic.validators.IBANValidator` (*use_nordea_extensions=False, include_countries=None*) *in-*

A validator for International Bank Account Numbers (IBAN - ISO 13616-1:2007).

deconstruct (*obj*)

Returns a 3-tuple of class import path, positional arguments, and keyword arguments.

static iban_checksum (*value*)

Returns check digits for an input IBAN number.

Original checksum in input value is ignored.

New in version 1.1.

7.59.5 Data

`localflavor.generic.countries.iso_3166.ISO_3166_1_ALPHA2_COUNTRY_CODES` = ('AD', 'AE', 'AF', 'AG', 'A
ISO 3166-1 country list. Sourced from <https://www.iso.org/obp/ui> on 2014-11-08

`localflavor.generic.countries.sepa.IBAN_SEPA_COUNTRIES` = ('AT', 'BE', 'BG', 'CH', 'CY', 'CZ', 'DE', 'DK',
European Payments Council list of SEPA scheme countries as of 2 Sep 2015.
<http://www.europeanpaymentscouncil.eu/index.cfm/knowledge-bank/epc-documents/epc-list-of-sepa-scheme-countries/>

|
localflavor, 3
localflavor.ar.forms, 21
localflavor.at.forms, 22
localflavor.au.forms, 22
localflavor.au.models, 23
localflavor.be.forms, 24
localflavor.bg.models, 25
localflavor.bg.utils, 25
localflavor.bg.validators, 25
localflavor.br.forms, 26
localflavor.ca.forms, 27
localflavor.ch.forms, 28
localflavor.cl.forms, 29
localflavor.cn.forms, 29
localflavor.co.forms, 30
localflavor.cz.forms, 31
localflavor.de.forms, 31
localflavor.dk.forms, 32
localflavor.ec.forms, 33
localflavor.ee.forms, 33
localflavor.es.forms, 34
localflavor.fi.forms, 35
localflavor.fr.forms, 36
localflavor.gb.forms, 37
localflavor.gb.gb_regions, 37
localflavor.generic.checksums, 72
localflavor.generic.forms, 70
localflavor.generic.models, 71
localflavor.generic.validators, 72
localflavor.gr.forms, 38
localflavor.hk.forms, 39
localflavor.hr.forms, 39
localflavor.hu.forms, 41
localflavor.id_.forms, 41
localflavor.ie.forms, 42
localflavor.il.forms, 42
localflavor.in_.forms, 43
localflavor.is_.forms, 44
localflavor.it.forms, 45
localflavor.it.util, 46
localflavor.jp.forms, 46
localflavor.kw.forms, 47
localflavor.lt.forms, 47
localflavor.lv.forms, 48
localflavor.mk.forms, 49
localflavor.mk.models, 50
localflavor.mt.forms, 50
localflavor.mx.forms, 50
localflavor.mx.models, 52
localflavor.nl.forms, 53
localflavor.nl.models, 53
localflavor.no.forms, 54
localflavor.nz.forms, 55
localflavor.pe.forms, 56
localflavor.pk.forms, 56
localflavor.pk.models, 57
localflavor.pl.forms, 57
localflavor.pt.forms, 59
localflavor.py_.forms, 60
localflavor.ro.forms, 60
localflavor.ru.forms, 62
localflavor.se.forms, 62
localflavor.se.utils, 63
localflavor.sg.forms, 63
localflavor.si.forms, 64
localflavor.sk.forms, 64
localflavor.tn.forms, 65
localflavor.tr.forms, 65
localflavor.us.forms, 66
localflavor.us.models, 67
localflavor.uy.forms, 69
localflavor.uy.util, 69
localflavor.za.forms, 70

A

ADMINISTRATIVE_UNIT_CHOICES (in module localflavor.pl.pl_administrativeunits), 58
 ARCBUField (class in localflavor.ar.forms), 21
 ARCUITField (class in localflavor.ar.forms), 21
 ARDNIField (class in localflavor.ar.forms), 21
 ARMED_FORCES_STATES (in module localflavor.us.us_states), 68
 ARPostalCodeField (class in localflavor.ar.forms), 22
 ARProvinceSelect (class in localflavor.ar.forms), 22
 ATSocialSecurityNumberField (class in localflavor.at.forms), 22
 ATStateSelect (class in localflavor.at.forms), 22
 ATZipCodeField (class in localflavor.at.forms), 22
 AUBusinessNumberField (class in localflavor.au.forms), 22
 AUBusinessNumberField (class in localflavor.au.models), 23
 AUPhoneNumberField (class in localflavor.au.forms), 23
 AUPhoneNumberField (class in localflavor.au.models), 23
 AUPostCodeField (class in localflavor.au.forms), 23
 AUPostCodeField (class in localflavor.au.models), 23
 AUStateField (class in localflavor.au.models), 23
 AUStateSelect (class in localflavor.au.forms), 23
 AUTaxFileNumberField (class in localflavor.au.forms), 23
 AUTaxFileNumberField (class in localflavor.au.models), 24

B

BEPhoneNumberField (class in localflavor.be.forms), 24
 BEPostalCodeField (class in localflavor.be.forms), 24
 BEProvinceSelect (class in localflavor.be.forms), 24
 BERRegionSelect (class in localflavor.be.forms), 24
 BGEGNField (class in localflavor.bg.models), 25
 BGEIKField (class in localflavor.bg.models), 25
 BICField (class in localflavor.generic.models), 71
 BICFormField (class in localflavor.generic.forms), 70
 BICValidator (class in localflavor.generic.validators), 72

BRCNPJField (class in localflavor.br.forms), 26
 BRCPFField (class in localflavor.br.forms), 26
 BRPhoneNumberField (class in localflavor.br.forms), 26
 BRProcessoField (class in localflavor.br.forms), 26
 BRStateChoiceField (class in localflavor.br.forms), 26
 BRStateSelect (class in localflavor.br.forms), 26
 BRZipCodeField (class in localflavor.br.forms), 26

C

CAPhoneNumberField (class in localflavor.ca.forms), 27
 CAPostalCodeField (class in localflavor.ca.forms), 27
 CAProvinceField (class in localflavor.ca.forms), 27
 CAProvinceSelect (class in localflavor.ca.forms), 27
 CASocialInsuranceNumberField (class in localflavor.ca.forms), 27
 CHIdentityCardNumberField (class in localflavor.ch.forms), 28
 CHPhoneNumberField (class in localflavor.ch.forms), 28
 CHSocialSecurityNumberField (class in localflavor.ch.forms), 28
 CHStateSelect (class in localflavor.ch.forms), 28
 CHZipCodeField (class in localflavor.ch.forms), 28
 clean() (localflavor.ar.forms.ARCBUField method), 21
 clean() (localflavor.ar.forms.ARCUITField method), 21
 clean() (localflavor.ar.forms.ARDNIField method), 21
 clean() (localflavor.au.forms.AUPhoneNumberField method), 23
 clean() (localflavor.br.forms.BRCNPJField method), 26
 clean() (localflavor.br.forms.BRCPFField method), 26
 clean() (localflavor.br.forms.BRProcessoField method), 26
 clean() (localflavor.cl.forms.CLRutField method), 29
 clean() (localflavor.cn.forms.CNIDCardField method), 30
 clean() (localflavor.cz.forms.CZPostalCodeField method), 31
 clean() (localflavor.jp.forms.JPPostalCodeField method), 46
 clean() (localflavor.pe.forms.PEDNIField method), 56
 clean() (localflavor.pe.forms.PERUCField method), 56
 clean() (localflavor.pk.forms.PKPhoneNumberField method), 56

- clean() (localflavor.ro.forms.ROCIFField method), 60
 - clean() (localflavor.ro.forms.ROCNPFfield method), 60
 - clean() (localflavor.ro.forms.ROPhoneNumberField method), 61
 - clean() (localflavor.sk.forms.SKPostalCodeField method), 65
 - clean() (localflavor.uy.forms.UYCIField method), 69
 - CLRegionSelect (class in localflavor.cl.forms), 29
 - CLRutField (class in localflavor.cl.forms), 29
 - CN_PROVINCE_CHOICES (in module localflavor.cn.cn_provinces), 30
 - CNCellNumberField (class in localflavor.cn.forms), 30
 - CNIDCardField (class in localflavor.cn.forms), 29
 - CNPhoneNumberField (class in localflavor.cn.forms), 30
 - CNPostCodeField (class in localflavor.cn.forms), 29
 - CNProvinceSelect (class in localflavor.cn.forms), 29
 - CODepartmentSelect (class in localflavor.co.forms), 30
 - COFA_STATES (in module localflavor.us.us_states), 68
 - CONTIGUOUS_STATES (in module localflavor.us.us_states), 68
 - COUNTIES_CHOICES (in module localflavor.ro.ro_counties), 61
 - COUNTY_CHOICES (in module localflavor.ee.ee_counties), 34
 - COUNTY_CHOICES (in module localflavor.lt.lt_choices), 48
 - COUNTY_CHOICES (in module localflavor.se.se_counties), 63
 - CURP_INCONVENIENT_WORDS (in module localflavor.mx.forms), 50
 - CZBirthNumberField (class in localflavor.cz.forms), 31
 - CZICNumberField (class in localflavor.cz.forms), 31
 - CZPostalCodeField (class in localflavor.cz.forms), 31
 - CZRegionSelect (class in localflavor.cz.forms), 31
- ## D
- DateField (class in localflavor.generic.forms), 70
 - DateTimeField (class in localflavor.generic.forms), 70
 - deconstruct() (localflavor.au.models.AUStateField method), 24
 - deconstruct() (localflavor.bg.validators.EGNValidator method), 25
 - deconstruct() (localflavor.bg.validators.EIKValidator method), 25
 - deconstruct() (localflavor.generic.validators.BICValidator method), 72
 - deconstruct() (localflavor.generic.validators.EANValidator method), 72
 - deconstruct() (localflavor.generic.validators.IBANValidator method), 72
 - deconstruct() (localflavor.nl.models.NLProvinceField method), 53
 - deconstruct() (localflavor.pk.models.PKStateField method), 57
 - DEIdentityCardNumberField (class in localflavor.de.forms), 31
 - DEPARTMENT_CHOICES (in module localflavor.co.co_departments), 30
 - DEPARTMENT_CHOICES (in module localflavor.fr.fr_department), 37
 - DEPARTMENT_CHOICES (in module localflavor.py.py_department), 60
 - DEPARTMENT_CHOICES (in module localflavor.uy.uy_departments), 70
 - DEPARTMENT_CHOICES_PER_REGION (in module localflavor.fr.fr_department), 37
 - DEPARTMENT_ROMAN_CHOICES (in module localflavor.py.py_department), 60
 - description (localflavor.au.models.AUBusinessNumberField attribute), 23
 - description (localflavor.au.models.AUPhoneNumberField attribute), 23
 - description (localflavor.au.models.AUPostCodeField attribute), 23
 - description (localflavor.au.models.AUStateField attribute), 24
 - description (localflavor.au.models.AUTaxFileNumberField attribute), 24
 - description (localflavor.nl.models.NLPhoneNumberField attribute), 53
 - description (localflavor.nl.models.NLProvinceField attribute), 53
 - description (localflavor.nl.models.NLSoFiNumberField attribute), 53
 - description (localflavor.nl.models.NLZipCodeField attribute), 54
 - description (localflavor.pk.models.PKPhoneNumberField attribute), 57
 - description (localflavor.pk.models.PKPostCodeField attribute), 57
 - description (localflavor.pk.models.PKStateField attribute), 57
 - DEStateSelect (class in localflavor.de.forms), 32
 - DEZipCodeField (class in localflavor.de.forms), 32
 - DISTRICT_CHOICES (in module localflavor.sk.sk_districts), 65
 - DK_MUNICIPALITIES (in module localflavor.dk.dk_municipalities), 33
 - DK_POSTALCODES (in module localflavor.dk.dk_postalcodes), 32
 - DKMunicipalitySelect (class in localflavor.dk.forms), 32
 - DKPhoneNumberField (class in localflavor.dk.forms), 32
 - DKPostalCodeField (class in localflavor.dk.forms), 32
- ## E
- ean() (in module localflavor.generic.checksums), 72
 - EANValidator (class in localflavor.generic.validators), 72
 - ECProvinceSelect (class in localflavor.ec.forms), 33

ee_checksum() (localflavor.ee.forms.EEPersonalIdentificationCode static method), 33

EEBusinessRegistryCode (class in localflavor.ee.forms), 33

EECountySelect (class in localflavor.ee.forms), 33

EEPersonalIdentificationCode (class in localflavor.ee.forms), 33

EEZipCodeField (class in localflavor.ee.forms), 34

EGNValidator (class in localflavor.bg.validators), 25

EIKValidator (class in localflavor.bg.validators), 25

ENGLAND_REGION_CHOICES (in module localflavor.gb.gb_regions), 37

ESCCCFIELD (class in localflavor.es.forms), 34

ESIdentityCardNumberField (class in localflavor.es.forms), 34

ESPhoneNumberField (class in localflavor.es.forms), 34

ESPostalCodeField (class in localflavor.es.forms), 35

ESProvinceSelect (class in localflavor.es.forms), 35

ESRegionSelect (class in localflavor.es.forms), 35

F

FIMunicipalitySelect (class in localflavor.fi.forms), 35

FISocialSecurityNumber (class in localflavor.fi.forms), 35

FIZipCodeField (class in localflavor.fi.forms), 35

formfield() (localflavor.au.models.AUBusinessNumberField method), 23

formfield() (localflavor.au.models.AUPhoneNumberField method), 23

formfield() (localflavor.au.models.AUPostCodeField method), 23

formfield() (localflavor.au.models.AUTaxFileNumberField method), 24

formfield() (localflavor.nl.models.NLPhoneNumberField method), 53

formfield() (localflavor.nl.models.NLSoFiNumberField method), 53

formfield() (localflavor.nl.models.NLZipCodeField method), 54

formfield() (localflavor.pk.models.PKPhoneNumberField method), 57

formfield() (localflavor.pk.models.PKPostCodeField method), 57

FRDepartmentField (class in localflavor.fr.forms), 36

FRDepartmentSelect (class in localflavor.fr.forms), 36

FRNationalIdentificationNumber (class in localflavor.fr.forms), 36

FRPhoneNumberField (class in localflavor.fr.forms), 36

FRRegion2016Select (class in localflavor.fr.forms), 36

FRRegionField (class in localflavor.fr.forms), 36

FRRegionSelect (class in localflavor.fr.forms), 36

FRSIRENField (class in localflavor.fr.forms), 36

FRSIRETField (class in localflavor.fr.forms), 36

FRZipCodeField (class in localflavor.fr.forms), 37

G

GB_NATIONS_CHOICES (in module localflavor.gb.gb_regions), 38

GB_REGION_CHOICES (in module localflavor.gb.gb_regions), 38

GBCountySelect (class in localflavor.gb.forms), 37

GBNationSelect (class in localflavor.gb.forms), 37

GBPostcodeField (class in localflavor.gb.forms), 37

get_egn_birth_date() (in module localflavor.bg.utils), 25

get_validation_digit() (in module localflavor.uy.util), 69

GOVERNORATE_CHOICES (in module localflavor.tn.tn_governorates), 65

GRMobilePhoneNumberField (class in localflavor.gr.forms), 38

GRPhoneNumberField (class in localflavor.gr.forms), 38

GRPostalCodeField (class in localflavor.gr.forms), 38

GRTaxNumberCodeField (class in localflavor.gr.forms), 38

H

has_valid_birth_date() (localflavor.pl.forms.PLPESELField method), 58

has_valid_birthday() (localflavor.cn.forms.CNIDCardField method), 30

has_valid_checksum() (localflavor.cn.forms.CNIDCardField method), 30

has_valid_checksum() (localflavor.pl.forms.PLNationalIDCardNumberField method), 58

has_valid_checksum() (localflavor.pl.forms.PLNIPField method), 57

has_valid_checksum() (localflavor.pl.forms.PLPESELField method), 58

has_valid_checksum() (localflavor.pl.forms.PLREGONField method), 58

has_valid_location() (localflavor.cn.forms.CNIDCardField method), 30

HKPhoneNumberField (class in localflavor.hk.forms), 39

HR_COUNTY_CHOICES (in module localflavor.hr.hr_choices), 40

HR_LICENSE_PLATE_PREFIX_CHOICES (in module localflavor.hr.hr_choices), 40

HR_PHONE_NUMBER_PREFIX_CHOICES (in module localflavor.hr.hr_choices), 40

HRCountySelect (class in localflavor.hr.forms), 39

HRJMBAGField (class in localflavor.hr.forms), 39

HRJMBGField (class in localflavor.hr.forms), 39

HRLicensePlateField (class in localflavor.hr.forms), 39

HRLicensePlatePrefixSelect (class in localflavor.hr.forms), 40

HROIBField (class in localflavor.hr.forms), 40

HRPhoneNumberField (class in localflavor.hr.forms), 40

HRPhoneNumberPrefixSelect (class in localflavor.hr.forms), 40

HRPostalCodeField (class in localflavor.hr.forms), 40
 HU_COUNTY_CHOICES (in module localflavor.hu.hu_counties), 41
 HUCountySelect (class in localflavor.hu.forms), 41

I

iban_checksum() (localflavor.generic.validators.IBANValidator static method), 72
 IBAN_SEPA_COUNTRIES (in module localflavor.generic.countries.sepa), 73
 IBANField (class in localflavor.generic.models), 71
 IBANFormField (class in localflavor.generic.forms), 70
 IBANValidator (class in localflavor.generic.validators), 72
 id_number_checksum() (in module localflavor.se.utils), 63
 IDLicensePlateField (class in localflavor.id_.forms), 41
 IDLicensePlatePrefixSelect (class in localflavor.id_.forms), 41
 IDNationalIdentityNumberField (class in localflavor.id_.forms), 41
 IDPhoneNumberField (class in localflavor.id_.forms), 42
 IDPostCodeField (class in localflavor.id_.forms), 42
 IDProvinceSelect (class in localflavor.id_.forms), 42
 IE_COUNTY_CHOICES (in module localflavor.ie.ie_counties), 42
 IECountySelect (class in localflavor.ie.forms), 42
 ILIDNumberField (class in localflavor.il.forms), 42
 ILMobilePhoneNumberField (class in localflavor.il.forms), 43
 ILPostalCodeField (class in localflavor.il.forms), 43
 INAadhaarNumberField (class in localflavor.in_.forms), 43
 INPhoneNumberField (class in localflavor.in_.forms), 43
 INStateField (class in localflavor.in_.forms), 44
 INStateSelect (class in localflavor.in_.forms), 44
 INZipCodeField (class in localflavor.in_.forms), 44
 IS_POSTALCODES (in module localflavor.is.is_postalcodes), 45
 ISIdNumberField (class in localflavor.is_.forms), 44
 ISO_3166_1_ALPHA2_COUNTRY_CODES (in module localflavor.generic.countries.iso_3166), 73
 ISPhoneNumberField (class in localflavor.is_.forms), 44
 ISPostalCodeSelect (class in localflavor.is_.forms), 44
 ITPhoneNumberField (class in localflavor.it.forms), 45
 ITProvinceSelect (class in localflavor.it.forms), 45
 ITRegionProvinceSelect (class in localflavor.it.forms), 45
 ITRegionSelect (class in localflavor.it.forms), 45
 ITSocialSecurityNumberField (class in localflavor.it.forms), 45
 ITVatNumberField (class in localflavor.it.forms), 45
 ITZipCodeField (class in localflavor.it.forms), 45

J

JP_PREFECTURES (in module localflavor.jp.jp_prefectures), 47
 JPPostalCodeField (class in localflavor.jp.forms), 46
 JPPrefectureCodeSelect (class in localflavor.jp.forms), 46
 JPPrefectureSelect (class in localflavor.jp.forms), 47

K

KWCivilIDNumberField (class in localflavor.kw.forms), 47
 KWGovernorateSelect (class in localflavor.kw.forms), 47

L

LICENSE_PLATE_PREFIX_CHOICES (in module localflavor.id.id_choices), 42
 localflavor (module), 1
 localflavor.ar.forms (module), 21
 localflavor.at.forms (module), 22
 localflavor.au.forms (module), 22
 localflavor.au.models (module), 23
 localflavor.be.forms (module), 24
 localflavor.bg.models (module), 25
 localflavor.bg.utils (module), 25
 localflavor.bg.validators (module), 25
 localflavor.br.forms (module), 26
 localflavor.ca.forms (module), 27
 localflavor.ch.forms (module), 28
 localflavor.cl.forms (module), 29
 localflavor.cn.forms (module), 29
 localflavor.co.forms (module), 30
 localflavor.cz.forms (module), 31
 localflavor.de.forms (module), 31
 localflavor.dk.forms (module), 32
 localflavor.ec.forms (module), 33
 localflavor.ee.forms (module), 33
 localflavor.es.forms (module), 34
 localflavor.fi.forms (module), 35
 localflavor.fr.forms (module), 36
 localflavor.gb.forms (module), 37
 localflavor.gb.gb_regions (module), 37
 localflavor.generic.checksums (module), 72
 localflavor.generic.forms (module), 70
 localflavor.generic.models (module), 71
 localflavor.generic.validators (module), 72
 localflavor.gr.forms (module), 38
 localflavor.hk.forms (module), 39
 localflavor.hr.forms (module), 39
 localflavor.hu.forms (module), 41
 localflavor.id_.forms (module), 41
 localflavor.ie.forms (module), 42
 localflavor.il.forms (module), 42
 localflavor.in_.forms (module), 43
 localflavor.is_.forms (module), 44

[localflavor.it.forms \(module\)](#), 45
[localflavor.it.util \(module\)](#), 46
[localflavor.jp.forms \(module\)](#), 46
[localflavor.kw.forms \(module\)](#), 47
[localflavor.lt.forms \(module\)](#), 47
[localflavor.lv.forms \(module\)](#), 48
[localflavor.mk.forms \(module\)](#), 49
[localflavor.mk.models \(module\)](#), 50
[localflavor.mt.forms \(module\)](#), 50
[localflavor.mx.forms \(module\)](#), 50
[localflavor.mx.models \(module\)](#), 52
[localflavor.nl.forms \(module\)](#), 53
[localflavor.nl.models \(module\)](#), 53
[localflavor.no.forms \(module\)](#), 54
[localflavor.nz.forms \(module\)](#), 55
[localflavor.pe.forms \(module\)](#), 56
[localflavor.pk.forms \(module\)](#), 56
[localflavor.pk.models \(module\)](#), 57
[localflavor.pl.forms \(module\)](#), 57
[localflavor.pt.forms \(module\)](#), 59
[localflavor.py_.forms \(module\)](#), 60
[localflavor.ro.forms \(module\)](#), 60
[localflavor.ru.forms \(module\)](#), 62
[localflavor.se.forms \(module\)](#), 62
[localflavor.se.utils \(module\)](#), 63
[localflavor.sg.forms \(module\)](#), 63
[localflavor.si.forms \(module\)](#), 64
[localflavor.sk.forms \(module\)](#), 64
[localflavor.tn.forms \(module\)](#), 65
[localflavor.tr.forms \(module\)](#), 65
[localflavor.us.forms \(module\)](#), 66
[localflavor.us.models \(module\)](#), 67
[localflavor.us.us_states.STATE_CHOICES \(in module localflavor.us.models\)](#), 68
[localflavor.us.us_states.US_STATES \(in module localflavor.us.models\)](#), 68
[localflavor.us.us_states.USPS_CHOICES \(in module localflavor.us.models\)](#), 69
[localflavor.uy.forms \(module\)](#), 69
[localflavor.uy.util \(module\)](#), 69
[localflavor.za.forms \(module\)](#), 70
[LTCountySelect \(class in localflavor.lt.forms\)](#), 47
[LTIDCodeField \(class in localflavor.lt.forms\)](#), 47
[LTMunicipalitySelect \(class in localflavor.lt.forms\)](#), 47
[LTPhoneField \(class in localflavor.lt.forms\)](#), 48
[LTPostalCodeField \(class in localflavor.lt.forms\)](#), 48
[luhn\(\) \(in module localflavor.generic.checksums\)](#), 72
[lv_checksum\(\) \(localflavor.lv.forms.LVPersonalCodeField static method\)](#), 49
[LVMunicipalitySelect \(class in localflavor.lv.forms\)](#), 48
[LVPersonalCodeField \(class in localflavor.lv.forms\)](#), 48
[LVPostalCodeField \(class in localflavor.lv.forms\)](#), 49

M

[MK_MUNICIPALITIES \(in module localflavor.mk.mk_choices\)](#), 50
[MKIdentityCardNumberField \(class in localflavor.mk.forms\)](#), 49
[MKIdentityCardNumberField \(class in localflavor.mk.models\)](#), 50
[MKMunicipalityField \(class in localflavor.mk.models\)](#), 50
[MKMunicipalitySelect \(class in localflavor.mk.forms\)](#), 49
[MUNICIPALITY_CHOICES \(in module localflavor.fi.fi_municipalities\)](#), 36
[MUNICIPALITY_CHOICES \(in module localflavor.lt.lt_choices\)](#), 48
[MUNICIPALITY_CHOICES \(in module localflavor.lv.lv_choices\)](#), 49
[MUNICIPALITY_CHOICES \(in module localflavor.no.no_municipalities\)](#), 54
[MXCLABEField \(class in localflavor.mx.forms\)](#), 50
[MXCLABEField \(class in localflavor.mx.models\)](#), 52
[MXCURPField \(class in localflavor.mx.forms\)](#), 51
[MXCURPField \(class in localflavor.mx.models\)](#), 52
[MXRFCField \(class in localflavor.mx.forms\)](#), 51
[MXRFCField \(class in localflavor.mx.models\)](#), 52
[MXSocialSecurityNumberField \(class in localflavor.mx.forms\)](#), 51
[MXSocialSecurityNumberField \(class in localflavor.mx.models\)](#), 52
[MXStateField \(class in localflavor.mx.models\)](#), 52
[MXStateSelect \(class in localflavor.mx.forms\)](#), 52
[MXZipCodeField \(class in localflavor.mx.forms\)](#), 52
[MXZipCodeField \(class in localflavor.mx.models\)](#), 52

N

[NLBankAccountNumberField \(class in localflavor.nl.models\)](#), 53
[NLPhoneNumberField \(class in localflavor.nl.forms\)](#), 53
[NLPhoneNumberField \(class in localflavor.nl.models\)](#), 53
[NLProvinceField \(class in localflavor.nl.models\)](#), 53
[NLProvinceSelect \(class in localflavor.nl.forms\)](#), 53
[NLSoFiNumberField \(class in localflavor.nl.forms\)](#), 53
[NLSoFiNumberField \(class in localflavor.nl.models\)](#), 53
[NLZipCodeField \(class in localflavor.nl.forms\)](#), 53
[NLZipCodeField \(class in localflavor.nl.models\)](#), 54
[NOMunicipalitySelect \(class in localflavor.no.forms\)](#), 54
[NON_CONTIGUOUS_STATES \(in module localflavor.us.us_states\)](#), 68
[NOPhoneNumberField \(class in localflavor.no.forms\)](#), 54
[NORTH_ISLAND_COUNCIL_CHOICES \(in module localflavor.nz.nz_councils\)](#), 56
[NORTHERN_IRELAND_REGION_CHOICES \(in module localflavor.gb.gb_regions\)](#), 38
[NOSocialSecurityNumber \(class in localflavor.no.forms\)](#), 54

NOZipCodeField (class in localflavor.no.forms), 54
 NZBankAccountNumberField (class in localflavor.nz.forms), 55
 NZNorthIslandCouncilSelect (class in localflavor.nz.forms), 55
 NZPhoneNumberField (class in localflavor.nz.forms), 55
 NZPostCodeField (class in localflavor.nz.forms), 55
 NZProvinceSelect (class in localflavor.nz.forms), 55
 NZRegionSelect (class in localflavor.nz.forms), 55
 NZSouthIslandCouncilSelect (class in localflavor.nz.forms), 55

O

OBSOLETE_STATES (in module localflavor.us.us_states), 68

P

PEDNIField (class in localflavor.pe.forms), 56
 PERRegionSelect (class in localflavor.pe.forms), 56
 PERUCField (class in localflavor.pe.forms), 56
 PhoneNumberField (class in localflavor.us.models), 67
 PKPhoneNumberField (class in localflavor.pk.forms), 56
 PKPhoneNumberField (class in localflavor.pk.models), 57
 PKPostCodeField (class in localflavor.pk.forms), 57
 PKPostCodeField (class in localflavor.pk.models), 57
 PKStateField (class in localflavor.pk.models), 57
 PKStateSelect (class in localflavor.pk.forms), 57
 PLCountySelect (class in localflavor.pl.forms), 57
 PLNationalIDCardNumberField (class in localflavor.pl.forms), 58
 PLNIPField (class in localflavor.pl.forms), 57
 PLPESELField (class in localflavor.pl.forms), 58
 PLPostalCodeField (class in localflavor.pl.forms), 58
 PLProvinceSelect (class in localflavor.pl.forms), 58
 PLREGIONField (class in localflavor.pl.forms), 58
 prepare_value() (localflavor.au.forms.AUBusinessNumberField method), 23
 prepare_value() (localflavor.au.forms.AUTaxFileNumberField method), 23
 prepare_value() (localflavor.generic.forms.IBANFormField method), 71
 PROVINCE_CHOICES (in module localflavor.ar.ar_provinces), 22
 PROVINCE_CHOICES (in module localflavor.be.be_provinces), 25
 PROVINCE_CHOICES (in module localflavor.ca.ca_provinces), 27
 PROVINCE_CHOICES (in module localflavor.ec.ec_provinces), 33

PROVINCE_CHOICES (in module localflavor.es.es_provinces), 35
 PROVINCE_CHOICES (in module localflavor.id.id_choices), 42
 PROVINCE_CHOICES (in module localflavor.it.it_province), 46
 PROVINCE_CHOICES (in module localflavor.nl.nl_provinces), 54
 PROVINCE_CHOICES (in module localflavor.nz.nz_provinces), 56
 PROVINCE_CHOICES (in module localflavor.tr.tr_provinces), 66
 PROVINCE_CHOICES (in module localflavor.za.za_provinces), 70
 PROVINCE_REGIONS (in module localflavor.it.it_province), 46
 PROVINCES_NORMALIZED (in module localflavor.ca.ca_provinces), 28
 PTCitizenCardNumberField (class in localflavor.pt.forms), 59
 PTPhoneNumberField (class in localflavor.pt.forms), 59
 PTRRegionSelect (class in localflavor.pt.forms), 59
 PTSocialSecurityNumberField (class in localflavor.pt.forms), 59
 PTZipCodeField (class in localflavor.pt.forms), 60
 PyDepartmentSelect (class in localflavor.py_.forms), 60
 PyNumberedDepartmentSelect (class in localflavor.py_.forms), 60

R

REGION_2016_CHOICES (in module localflavor.fr.fr_region), 37
 REGION_CHOICES (in module localflavor.be.be_regions), 25
 REGION_CHOICES (in module localflavor.cl.cl_regions), 29
 REGION_CHOICES (in module localflavor.cz.cz_regions), 31
 REGION_CHOICES (in module localflavor.es.es_regions), 35
 REGION_CHOICES (in module localflavor.fr.fr_region), 37
 REGION_CHOICES (in module localflavor.it.it_region), 46
 REGION_CHOICES (in module localflavor.nz.nz_regions), 56
 REGION_CHOICES (in module localflavor.pe.pe_region), 56
 REGION_CHOICES (in module localflavor.pt.pt_regions), 60
 REGION_CHOICES (in module localflavor.sk.sk_regions), 65
 REGION_HOVEDSTADEN (in module localflavor.dk.dk_municipalities), 32

- REGION_MIDTJYLLAND (in module localflavor.dk.dk_municipalities), 32
- REGION_NORDJYLLAND (in module localflavor.dk.dk_municipalities), 32
- REGION_PROVINCE_CHOICES (in module localflavor.it.it_region), 46
- REGION_PROVINCES (in module localflavor.it.it_region), 46
- REGION_SJAELLAND (in module localflavor.dk.dk_municipalities), 32
- REGION_SYDDANMARK (in module localflavor.dk.dk_municipalities), 33
- RFC_INCONVENIENT_WORDS (in module localflavor.mx.forms), 52
- ROCIFField (class in localflavor.ro.forms), 60
- ROCNPFfield (class in localflavor.ro.forms), 60
- ROCountyField (class in localflavor.ro.forms), 61
- ROCountySelect (class in localflavor.ro.forms), 61
- ROIBANField (class in localflavor.ro.forms), 61
- ROPhoneNumberField (class in localflavor.ro.forms), 61
- ROPostalCodeField (class in localflavor.ro.forms), 61
- RU_COUNTY_CHOICES (in module localflavor.ru.ru_regions), 62
- RU_REGIONS_CHOICES (in module localflavor.ru.ru_regions), 62
- RUAlienPassportNumberField (class in localflavor.ru.forms), 62
- RUCountySelect (class in localflavor.ru.forms), 62
- RUPassportNumberField (class in localflavor.ru.forms), 62
- RUPostalCodeField (class in localflavor.ru.forms), 62
- RURegionSelect (class in localflavor.ru.forms), 62
- S**
- SCOTTISH_REGION_CHOICES (in module localflavor.gb.gb_regions), 38
- SECountySelect (class in localflavor.se.forms), 62
- SEOrganisationNumberField (class in localflavor.se.forms), 62
- SEPersonalIdentityNumberField (class in localflavor.se.forms), 63
- SEPostalCodeField (class in localflavor.se.forms), 63
- SI_POSTALCODES (in module localflavor.si.si_postalcodes), 64
- SIEMSOField (class in localflavor.si.forms), 64
- SIPhoneNumberField (class in localflavor.si.forms), 64
- SIPostalCodeField (class in localflavor.si.forms), 64
- SIPostalCodeSelect (class in localflavor.si.forms), 64
- SITaxNumberField (class in localflavor.si.forms), 64
- SKDistrictSelect (class in localflavor.sk.forms), 64
- SKPostalCodeField (class in localflavor.sk.forms), 64
- SKRegionSelect (class in localflavor.sk.forms), 65
- SOUTH_ISLAND_COUNCIL_CHOICES (in module localflavor.nz.nz_councils), 56
- SplitDateTimeField (class in localflavor.generic.forms), 71
- ssn_check_digit() (in module localflavor.it.util), 46
- ssn_validation() (in module localflavor.it.util), 46
- STATE_CHOICES (in module localflavor.at.at_states), 22
- STATE_CHOICES (in module localflavor.au.au_states), 24
- STATE_CHOICES (in module localflavor.br.br_states), 27
- STATE_CHOICES (in module localflavor.ch.ch_states), 29
- STATE_CHOICES (in module localflavor.de.de_states), 32
- STATE_CHOICES (in module localflavor.in.in_states), 44
- STATE_CHOICES (in module localflavor.mx.mx_states), 52
- STATE_CHOICES (in module localflavor.pk.pk_states), 57
- STATES_NORMALIZED (in module localflavor.in.in_states), 44
- STATES_NORMALIZED (in module localflavor.us.us_states), 69
- T**
- TNGovernorateSelect (class in localflavor.tn.forms), 65
- to_python() (localflavor.au.models.AUBusinessNumberField method), 23
- to_python() (localflavor.au.models.AUTaxFileNumberField method), 24
- to_python() (localflavor.nl.models.NLZipCodeField method), 54
- TRIdentificationNumberField (class in localflavor.tr.forms), 65
- TRPhoneNumberField (class in localflavor.tr.forms), 65
- TRPostalCodeField (class in localflavor.tr.forms), 66
- TRProvinceSelect (class in localflavor.tr.forms), 66
- U**
- UMCNField (class in localflavor.mk.forms), 49
- UMCNField (class in localflavor.mk.models), 50
- US_TERRITORIES (in module localflavor.us.us_states), 68
- USPhoneNumberField (class in localflavor.us.forms), 66
- USPostalCodeField (class in localflavor.us.models), 67
- USPSSelect (class in localflavor.us.forms), 66
- USSocialSecurityNumberField (class in localflavor.us.forms), 66
- USSocialSecurityNumberField (class in localflavor.us.models), 67
- USStateField (class in localflavor.us.forms), 66
- USStateField (class in localflavor.us.models), 67
- USStateSelect (class in localflavor.us.forms), 67

USZipCodeField (class in localflavor.us.forms), 67
USZipCodeField (class in localflavor.us.models), 67
UYCIField (class in localflavor.uy.forms), 69
UYDepartmentSelect (class in localflavor.uy.forms), 69

V

valid_date() (localflavor.lt.forms.LTIDCodeField method), 47
validate_id_birthday() (in module localflavor.se.utils), 63
validator (localflavor.nl.models.NLPhoneNumberField attribute), 53
validators (localflavor.au.models.AUBusinessNumberField attribute), 23
validators (localflavor.au.models.AUTaxFileNumberField attribute), 24
validators (localflavor.nl.models.NLSoFiNumberField attribute), 53
validators (localflavor.nl.models.NLZipCodeField attribute), 54
vat_number_check_digit() (in module localflavor.it.util), 46
vat_number_validation() (in module localflavor.it.util), 46
VOIVODESHIP_CHOICES (in module localflavor.pl.pl_voivodeships), 58

W

WALES_REGION_CHOICES (in module localflavor.gb.gb_regions), 38
widget (localflavor.fr.forms.FRDepartmentField attribute), 36
widget (localflavor.fr.forms.FRRegionField attribute), 36

Z

ZAIDField (class in localflavor.za.forms), 70
ZAPostCodeField (class in localflavor.za.forms), 70
ZAProvinceSelect (class in localflavor.za.forms), 70