NAME

User::Identity::Location − physical location of a person

INHERITANCE

```
User::Identity::Location
  is a User::Identity::Item
```

SYNOPSIS

```
use User::Identity;
use User::Identity::Location;
my $me   = User::Identity−>new(...);
my $addr = User::Identity::Location−>new(...);
$me−>add(location => $addr);

# Simpler

use User::Identity;
my $me   = User::Identity−>new(...);
my $addr = $me−>add(location => ...);
```

DESCRIPTION

The `User::Identity::Location` object contains the description of a physical location of a person: home, work, travel. The locations are collected by a User::Identity::Collection::Locations object.

Nearly all methods can return `undef`. Some methods produce language or country specific output.

Extends "DESCRIPTION" in User::Identity::Item.

METHODS

Extends "METHODS" in User::Identity::Item.

Constructors

Extends "Constructors" in User::Identity::Item.

User::Identity::Location−>**new**([$name], `%options`)

Create a new location. You can specify a name as first argument, or in the OPTION list. Without a specific name, the organization is used as name.

```
−Option      −−Defined in          −−Default
 country                             undef
 country_code                        undef
 description   User::Identity::Item  undef
 fax                                 undef
 name          User::Identity::Item  <required>
 organization                        undef
 parent        User::Identity::Item  undef
 pc                                  undef
 phone                               undef
 pobox                               undef
 pobox_pc                            undef
 postal_code                         <value of option pc>
 state                               undef
 street                              undef
```

country => STRING
country_code => STRING
description => STRING
fax => STRING|ARRAY

name => STRING
organization => STRING
parent => OBJECT
pc => STRING
Short name for `postal_code`.

phone => STRING|ARRAY
pobox => STRING
pobox_pc => STRING
postal_code => STRING
state => STRING
street => STRING

Attributes

Extends "Attributes" in User::Identity::Item.

`$obj->`**city**()
The city where the address is located.

`$obj->`**country**()
The country where the address is located. If the name of the country is not known but a country code is defined, the name will be looked-up using Geography::Countries (if installed).

`$obj->`**countryCode**()
Each country has an ISO standard abbreviation. Specify the country or the country code, and the other will be filled in automatically.

`$obj->`**description**()
Inherited, see "Attributes" in User::Identity::Item

`$obj->`**fax**()
One or more fax numbers, like **phone()**.

`$obj->`**fullAddress**()
Create an address to put on a postal mailing, in the format as normal in the country where it must go to. To be able to achieve that, the country code must be known. If the city is not specified or no street or pobox is given, undef will be returned: an incomplete address.

example:

```
 print $uil->fullAddress;
 print $user->find(location => 'home')->fullAddress;
```

`$obj->`**name**([$newname])
Inherited, see "Attributes" in User::Identity::Item

`$obj->`**organization**()
The organization (for instance company) which is related to this location.

`$obj->`**phone**()
One or more phone numbers. Please use the international notation, which starts with `'+'`, for instance `+31-26-12131`. In scalar context, only the first number is produced. In list context, all numbers are presented.

`$obj->`**pobox**()
Post Office mail box specification. Use `"P.O.Box 314"`, not simple `314`.

`$obj->`**poboxPostalCode**()
The postal code related to the Post-Office mail box. Defined by **new()** option `pobox_pc`.

`$obj->`**postalCode**()
The postal code is very country dependent. Also, the location of the code within the formatted string is country dependent.

`$obj`−>**state**()

The state, which is important for some countries but certainly not for the smaller ones. Only set this value when you state has to appear on printed addresses.

`$obj`−>**street**()

Returns the address of this location. Since Perl 5.7.3, you can use unicode in strings, so why not format the address nicely?

Collections

Extends ‘‘Collections’’ in User::Identity::Item.

`$obj`−>**add**($collection, `$role`)

Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`−>**addCollection**(`$object` | <[$type], `%options`>)

Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`−>**collection**($name)

Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`−>**parent**([$parent])

Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`−>**removeCollection**($object|$name)

Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`−>**type**()
User::Identity::Location−>**type**()

Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`−>**user**()

Inherited, see ‘‘Collections’’ in User::Identity::Item

Searching

Extends ‘‘Searching’’ in User::Identity::Item.

`$obj`−>**find**($collection, `$role`)

Inherited, see ‘‘Searching’’ in User::Identity::Item

DIAGNOSTICS

Error: `$object` is not a collection.

The first argument is an object, but not of a class which extends User::Identity::Collection.

Error: Cannot load collection module for `$type` ($class).

Either the specified `$type` does not exist, or that module named `$class` returns compilation errors. If the type as specified in the warning is not the name of a package, you specified a nickname which was not defined. Maybe you forgot the ’require’ the package which defines the nickname.

Error: Creation of a collection via `$class` failed.

The `$class` did compile, but it was not possible to create an object of that class using the options you specified.

Error: Don’t know what type of collection you want to add.

If you add a collection, it must either by a collection object or a list of options which can be used to create a collection object. In the latter case, the type of collection must be specified.

Warning: No collection `$name`

The collection with `$name` does not exist and can not be created.

SEE ALSO

This module is part of User-Identity distribution version 1.01, built on February 11, 2022. Website: *http://perl.overmeer.net/CPAN/*

LICENSE