poplar$_{i}soccDocumentation$

Release 1.1.0

Chris Binckly

Mar 27, 2020

Contents

This package enforces the use of ISO-3166-1 compliant country codes in any field in the Sage Desktop. It contains validation code as well as unit and acceptance tests.

CHAPTER 1

poplar_isocc

This package implements a method to validate a string against a reference list of country codes.

poplar_isocc.**is_valid_iso_cc**(*code*)

Is code a valid ISO-3166 code?

Parameters **code** (*str*) – country code to validate

Return type (bool, list)

Returns (True, []) if code is valid, (False, [matches]) otherwise

New in version 0.3.0: Fuzzy matches returned on no match.

```
from poplar_isocc import is_valid_iso_cc

# Check if a code is valid.
is_valid_iso_cc("CA")
    # returns True
is_valid_iso_cc("ZZ")
    # returns False
```

CHAPTER 2

Testing

This package includes both Python and Extender Unit testing as well as an Extender Acceptance Test.

class poplar_isocc.tests.test_poplar_isocc.**IsoCountryCodeTestCase**(*methodName='runTest'*)
Bases: unittest.case.TestCase

Unit tests for ISO country code validation.

test_is_valid_iso_cc()
Test that the ISO code validation function works as expected.

1. For each code in a list of valid codes, verify that the validation passes.
2. For each code in a list of invalid codes, including overlength and empty values, verify that the validation fails.

class poplar_isocc.tests.extest_poplar_isocc.**IsoCountryCodeTestCase**(*log_level=15*)
Bases: extools.extest.ExTestCase

Unit tests for ISO country code validation.

test_is_valid_iso_cc()
Passes iff the validation works for a valid, invalid and empty country code in SAMINC.

1. Open a record with a valid country code, verify it validates.
2. Open a record with an invalid country code, verify validation fails.
3. Open a record with an empty country code, verify validation fails.

class poplar_isocc.tests.extest_poplar_isocc.**IsoCountryCodeAcceptanceTestCase**(*log_level=15*)
Bases: extools.extest.ExTestCase

Acceptance tests for ISO country code validation.

test_is_valid_iso_enforced_in_view()
Verify that a ISO code validation is enforced at the view.

1. Navigate to a record with an empty country code.
2. Put an invalid country code in the field (raises ExViewError).

3. Put a valid country code in the field (succeeds).

`poplar_isocc` part of a general demonstration of how Python Packaging can be used with Orchid Extender. Best consumed with the accompanying presentation, Python Packaging for Extender - ISO Country Codes and a cold beer.

The code in this package is very simple, relying on the iso3166 package for reference data and performing only a simple validation. It demonstrates the key concepts in improving code reuse and distribution for Extender, including:

- leveraging the extools library for testing
- inclusion of unit and acceptance testing using ExTestCase
- how packaging can make installation, upgrade, and backport a breeze

This package can be used to demonstrate the upgrade and feature backport. The relevant versions are:

- v0.1.3: enforces two character ISO country codes on A/R Customer Country
- v0.2.0: enforces three character ISO country codes on A/R Customer Country
- v0.3.7: enforces three character ISO country codes on A/R Customer Country with fuzzy matching and recommendatons on invalid code.

To demo with this package:

1. Install `expip`, the Package Manager for Orchid Extender.
2. Open A/R Customers, input the value "CDN" into a new Customer Record.
    - The put is successful.
3. Open `expip` and install `poplar_isocc==0.1.3`. This version enforces two character country codes.
4. Open A/R Customers, input the value "CDN" into a new Customer Record.
    - The put fails with a message to the user.
5. Input the value "CA" into the new Customer Record.
    - The put succeeds silently.
6. Open `expip` and install `poplar_isocc==0.2.0`. This version enforces three character country codes.
7. Open A/R Customers, input the value "CA" into a new Customer Record.
    - The put fails with a message to the user.
8. Input the value "CAN" into the new Customer Record.
    - The put succeeds silently.
9. Open `expip`, select `poplar_isocc` from the drop-down, and click `Upgrade`. This will install the latest version, which includes fuzzy matching to make recommendations on invalid input.
10. Open A/R Customers, input the value "CDN" into a new Customer Record.
    - The put fails with a message to the user recommending "CAN"
11. Input the value "CAN" into the new Customer Record.
    - The put succeeds silently.

Done.

This package, the Python Package Manager for Orchid Extender, and the extools library were created and are maintained by 2665093 Ontario Inc. Comments and quesions are always welcome, send an email.

Python Module Index

p

Index

I

P

T