

Mystery Case Files: Ravenhearst - French Torrent Download Fix [key Serial Number]

Trifle Wikipedia, the free encyclopedia The designer mystery holiday. Mystery Case Files : Return to ravenhearst [Mac OS X] Serial [key] Not [a] RequiQ: Mock PyPI upload using unittest.mock I am writing a unit test for PyPI, and I want to mock the entire upload process. I have used the unittest.mock library in Python to mock the urllib2.urlopen() function, but I can't figure out how to mock the PIP_INDEXES functionality. I tried: import PyPI import unittest.mock as mock class MockPyPI: index = None _check_releases = None @classmethod def index(cls, *args, **kwargs): if cls.index is not None: return cls.index cls.index = MockPyPI._check_releases return cls.index(*args, **kwargs) class TestMockingPyPIIndexes(unittest.TestCase): def setUp(self): self.unmocked_releases = test_data['releases'] self.mocked_releases = self.index(*self.unmocked_releases) self.mocked_index = MockPyPI.index(*self.mocked_releases) def test_index(self): results = self.mocked_index.results self.assertFalse(len(results)) The index() method calls _check_releases, and if that is not None, it calls index() on it. My test is failing because PyPI has released some packages since the test was written, but _check_releases is never set to None. So I'm not sure how to create a mock PyPI instance that has a real index and _check_releases that is actually set to None? A:



