
```
# -*- coding: UTF-8 -*-
```

```
"""
```

```
    Tests for the pdf to html transform.
```

```
"""
```

```
import unittest
```

```
from os.path import join, abspath, dirname
```

```
import zope.app.publisher.browser
```

```
import zope.component
```

```
from zope.component import queryUtility
```

```
from zope.component.testing import setUp, tearDown
```

```
from zope.configuration.xmlconfig import XMLConfig
```

```
from zope.testing import doctest
```

```
from zope.testing.doctestunit import DocTestSuite
```

```
import plone.transforms
```

```
from plone.transforms.interfaces import ITransform
```

```
PREFIX = abspath(dirname(__file__))
```

```
def input_file_path(name):
```

```
    return join(PREFIX, 'input', name)
```

```
def configurationSetUp(self):
```

```
    setUp()
```

```
    XMLConfig('meta.zcml', zope.component)()
```

```
    XMLConfig('meta.zcml', zope.app.publisher.browser)()
```

```
    XMLConfig('configure.zcml', plone.transforms)()
```

```
def testPDFTransform():
```

```
    """
```

```
    First we get the transform utility.
```

```
    >>> util = queryUtility(ITransform,
```

```
    ...             name='plone.transforms.text.pdf_html.PDFTransform')
```

```
    >>> util
```

```
    <plone.transforms.binary.pdf_html.PDFTransform object at ...>
```

```
    Set up some test text:
```

```
    >>> input_file_path()
```

```
>>> text = ""
>>> data = (chr for chr in text)
```

Now transform the data:

```
>>> result = util.transform(data)
```

And check the result:

```
>>> result
<plone.transforms.transform.TransformResult object at ...>

>>> result.data
<generator object at ...>

>>> 'Error: May not be a PDF file' in ''.join(result.data)
True
"""
```

```
def test_suite():
    suite = unittest.TestSuite((
        DocTestSuite('plone.transforms.binary.pdf_html'),
    ))
    suite.addTest(
        DocTestSuite(setUp=configurationSetUp,
                      tearDown=tearDown,
                      optionflags=doctest.ELLIPSIS |
                      doctest.NORMALIZE_WHITESPACE))
    return suite

if __name__ == '__main__':
    unittest.main(defaultTest="test_suite")
```