

XML Objectifier

This package is hosted at:

https://svn.mrc-epid.cam.ac.uk/private/MRC_Webforms_Python/xml-objectifier/

This package does the basic conversion of XML into an object structure of sections, question groups and questions. It was separated out from the main project so it could be used in other projects. It is used in the XML Editor and the Forms System Renderer extension.

The structure is

- Application class - The main class is a container object for the sections, it represents the complete XML file.
- Section class - This class contains question groups and properties about itself.
- Question Group class - This contains Question and TextNodes as well as having properties relevant to itself.
- Question class
- TextNode - These are just instances of a Bunch. A Bunch is a class that extends a simple python dictionary so that keys can be accessed using standard python object notation e.g. `my_thing.myproperty` rather than the standard dictionary key `mything['myproperty']`. This allows looping through a mixture of textnodes and other objects without having to use other notations.
- MethodMixin - a support class that provides shared functionality and some extra passing functions that allow looping through mixed objects without errors.

Installation

This is best installed inside a virtualenv where you are running your Django project but it will install anywhere.

```
pip install git+git://github.com/cam-mrc-epid/xml_objectifier
```

OR

```
pip install  
svn+https://svn.mrc-epid.cam.ac.uk/private/MRC_Webforms_Python/xml-objectifier/
```

Dependencies

XML Objectifier's install process will install lxml and bunch.

Usage

To load an XML file as objects

```
from xml_objectifier.objectifier import Application
my_app = Application('some name', 'somepath/somewhere/FamHist.xml')
section_obj = my_app.get_section(section_number) # uses position
qg = section_obj.get_question_group(question_group_number) # uses position
q = question_group.get_question(question_number) # uses position
```

The main purpose of this structure is to build a structure that can be passed to the templates and iterated over. The brief was that a section, question_group or question must be able to be rendered separately. So templates are rendered to use this structure in the Forms System Renderer extension. See `fs_renderer`'s views for more details.

Project Structure

This is a simple Python package with `setup.py` in the root.

`xml_objectifier/objectifier.py` contains all the classes for conversion of the XML file to objects.

Dependencies

`lxml` # provides the objectify functions for the first stage in conversion of XML to our object structure.

`bunch` # provides a dictionary adaption that allows dictionary keys to be accessed and set using standard object property notation e.g. `myDict.some_key`