

Re: dynamic class instantiation

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-01/msg05063.html>

- *From:* Larry Bates <larry.bates@xxxxxxxxxxx>
 - *Date:* Mon, 30 Jan 2006 16:19:13 -0600
-

Ognen Duzlevski wrote:

```
> Larry Bates <larry.bates@xxxxxxxxxxx> wrote:
>>> Now I want to use something like xml.dom.minidom to "parse" the
>>> .xml file into a set of classes defined according to the "language
>>> definition" file. The parse() method from the xml.dom.minidom
>>> package will return a document instance and I can get the node
>>> name from it. Say I got "page" as a string. How do I go about
>>> instantiating a class from this piece of information? To make it
>>> more obvious how do I create the page() class based on the "page"
>>> string I have? I want to make this generic so for me the language
>>> definition file will contain pages, functions, datasets etc. but
>>> for someone else mileage will vary.
>>>
>>> Thanks,
>>> Ognen
>
>> I think you should rethink this approach. Why have 3 different
>> files instead of just defining everything as Python classes?
>> Get good baseclass definitions and use OOP inheritance to
>> create your specific class instances. I think mixing XML,
>> language definition file (LDF), LDF to python parser is a LOT
>> harder (and slower) than just writing the classes outright.
>
> Hi Larry,
>
> here is the whole story. I have a database (postgres) xml "transformer"
> and substitution engine. It is written entirely in plpgsql and the whole
> environment works something like this. Someone writes an xml file that
> defines a web page. This xml file is then fed into the database at the
> time of development. Now at runtime a mod_python apache webapp makes a
> request for a certain page. Its xml form is retrieved from the database
> and a lot of data substitution goes on. Also parts of the page are "masked"
> out by the database engine based on the "level" of the person requesting
> the page. The "transformed and substituted" xml is then spit back to the
> webapp which now applies an xml->html transformation and presents the
> final result to the user. So, I want people to be able to "define" their
> own "language" when it comes to page definitions. And instead of rewriting
> the xml file parser I want to be able to generate the parser.
>
```

> Thanks,
> Ognen

Well you certainly can do that, but why invent a new language to define the pages instead of just using Python directly? I did an application a few years ago that sounds a little like what you describe and now wish I had not spent so much time doing Text File->Python transformations. The logic got pretty ugly and extending things was rather involved. It wasn't until I started using ReportLab that I began to "see the light" on just keeping everything in Python and inheriting from base classes and extending.

If you want to define a Python class, the best way to do that is to use Python from the outset. I probably don't completely understand what you want to do, but thought I'd at least put in my 2-cents worth.

Regards,
Larry

• **References:**

- ◆ ***dynamic class instantiation***
 - ◇ *From: Ognen Duzlevski*
 - ◆ ***Re: dynamic class instantiation***
 - ◇ *From: Larry Bates*
 - ◆ ***Re: dynamic class instantiation***
 - ◇ *From: Ognen Duzlevski*
-
- Prev by Date: ***Re: Deterministic destruction and RAI idioms in Python***
 - Next by Date: ***Re: "Intro to Pyparsing" Article at ONLamp***
 - Previous by thread: ***Re: dynamic class instantiation***
 - Next by thread: ***Re: dynamic class instantiation***
 - Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***