Getting started

What is pyntree?

Overview

pyntree is a python package which allows you to easily and syntactically save your data. Not only that, it also lets you save in multiple formats, and even serialize and compress data by merely changing a few characters.

History

The first iteration of what is now known as pyntree came about when its sole developer decided that typing SQL commands into strings and running them was not a very pythonic way of doing things. Instead of dealing with this frustration, he created his own "database" system - DataManager. Although this system was never publicly released, pyndb, its successor, was (This is why pyndb was released on version 2 to start with). pyndb introduced a "node" system, in which each key of a dictionary was represented by a python object called a Node. pyntree keeps this structure, but its main difference is that nodes are created on the fly, via the `__getattr__` magic method.

A basic program with pyntree

Let's take a simple, yet practical example.

Let's say you've written a web service, and need to save user data and be able to send it to clients quickly. You also need to load several config files.

Doing so with pyntree is easy and syntactic:

```
from pyntree import Node
from datetime import datetime

config = Node('config.json')
secrets_config = Node('secrets_config.json')
user_db = Node('users.pyn', autosave=True)  # No need to call db.save() all the time!

# On event
user_db.get(user).data = new_data
user_db.get(user).data.time = datetime.now()  # You can save objects directly to file!

# On user requesting data
send(user_db.get(user)())  # Calling the Node returns its value
```

Terminology

Node

A pythonic representation of data - each key in a dictionary is represented by a Node object. Node objects representing a dictionary "contain" (Nodes are created on-demand) other Nodes representing any keys in that dictionary.

Root node

This is your primary node - the one you initialize with file parameters and location (or a dictionary). The root node represents your main dictionary.

Child node

The root node spawns child nodes to represent all of the keys it contains. These child nodes can, again, spawn more nodes - but that does not make them root nodes.

Installing

pyntree is available via pip:

```
pip install pyntree
```

That's it!