commentjson Documentation

Release 0.4

Vaidik Kapoor

Jun 19, 2018

Contents

`commentjson` is a Python library that lets you have Python and JavaScript style inline comments in your JSON files. Its API is very similar to the Python Standard library's json module.

CHAPTER 1

What does commentjson do?

`commentjson` allows you to deserialize JSON files with Python and JavaScript style comments in them. `commentjson`'s API is the same as the standard library's json module. If you are using or like using JSON for configuration files, `commentjson` can be very handy.

You may put comments in your JSON files like so:

```
{
    "name": "Vaidik Kapoor", # Person's name
    "location": "Delhi, India", // Person's location

    # Section contains info about
    // person's appearance
    "appearance": {
        "hair_color": "black",
        "eyes_color": "black",
        "height": "6"
    }
}
```

CHAPTER 2

Installation

```
pip install commentjson
```

CHAPTER 3

Basic Usage

Since commentjson's API is the same as standard libraries json module, it is extremely simple to use commentjson. You don't have to do anything new to use commentjson. Here is what you have to do:

1. Write your JSON files by hand or use standard library's json module or commentjson to create a new JSON file.
2. Open the JSON file in your text editor and add comments the same way you would in Python (using # and not docstrings) or the same way you would in JavaScript (using // and not multi-line comments using /** */).
3. Use commentjson's loads or load method to deserialize the file just like how you would use json module to parse a normal JSON string without comments. load and loads will return you a Python object.

```
>>> import commentjson
>>>
>>> json_string = """{
...     "name": "Vaidik Kapoor", # Person's name
...     "location": "Delhi, India", // Person's location
...
...     # Section contains info about
...     // person's appearance
...     "appearance": {
...         "hair_color": "black",
...         "eyes_color": "black",
...         "height": "6"
...     }
... }"""
>>>
>>> json_loaded = commentjson.loads(json_string)
>>> print json_loaded
{u'appearance': {u'eyes_color': u'black', u'hair_color': u'black', u'height': u'6'}, u
↪'name': u'Vaidik Kapoor', u'location': u'Delhi, India'}
```

CHAPTER 4

API Documentation

`commentjson.`**`loads`**(*text*, ***kwargs*)

Deserialize *text* (a *str* or *unicode* instance containing a JSON document with Python or JavaScript like comments) to a Python object.

Parameters

- **`text`** – serialized JSON string with or without comments.
- **`kwargs`** – all the arguments that json.loads accepts.

Raises commentjson.JSONLibraryException

Returns dict or list.

`commentjson.`**`dumps`**(*obj*, ***kwargs*)

Serialize *obj* to a JSON formatted *str*. Accepts the same arguments as *json* module in stdlib.

Parameters

- **`obj`** – a JSON serializable Python object.
- **`kwargs`** – all the arguments that json.dumps accepts.

Raises commentjson.JSONLibraryException

Returns str serialized string.

`commentjson.`**`load`**(*fp*, ***kwargs*)

Deserialize *fp* (a *.read()*-supporting file-like object containing a JSON document with Python or JavaScript like comments) to a Python object.

Parameters

- **`fp`** – a *.read()*-supporting file-like object containing a JSON document with or without comments.
- **`kwargs`** – all the arguments that json.load accepts.

Raises commentjson.JSONLibraryException

Returns dict or list.

`commentjson.`**`dump`**(*obj*, *fp*, ***kwargs*)

Serialize *obj* as a JSON formatted stream to *fp* (a *.write()*-supporting file-like object). Accepts the same arguments as *json* module in stdlib.

Parameters

- **`obj`** – a JSON serializable Python object.
- **`fp`** – a *.read()*-supporting file-like object containing a JSON document with or without comments.
- **`kwargs`** – all the arguments that json.dump accepts.

Raises commentjson.JSONLibraryException

`exception` `commentjson.`**`JSONLibraryException`**(*exc*)

Exception raised when the JSON library in use raises an exception i.e. the exception is not caused by *commentjson* and only caused by the JSON library *commentjson* is using.

Note: As of now, `commentjson` supports only standard library's `json` module. It might start supporting other widely-used contributed JSON libraries in the future.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Index