Test

February 8, 2023

[1]:
```
import picaso.justdoit as jdi
import picaso.references as pref
```

[2]:
```
opa = jdi.opannection(wave_range=[0.3,1]) #lets just use all defaults
planet=jdi.inputs()
planet.phase_angle(0) #radians
planet.gravity(gravity=25, gravity_unit=jdi.u.Unit('m/(s**2)')) #any astropy␣
 ↪units available
planet.star(opa, 5000,0,4.0) #opacity db, pysynphot database, temp,␣
 ↪metallicity, logg
planet.atmosphere(filename=jdi.jupiter_pt(), delim_whitespace=True)
full_output=planet.spectrum(opa, full_output=True)
```

[3]:
```
refs = pref.References()
opa_latex, bibdb = refs.get_opa(full_output=full_output['full_output'])
```

```
---------------------------------------------------------------------------
JSONDecodeError                           Traceback (most recent call last)
Input In [3], in <cell line: 1>()
----> 1 refs = pref.References()
      2 opa_latex, bibdb = refs.get_opa(full_output=full_output['full_output'])

File ~/Documents/Exo_Surface/picasso/picaso/picaso/references.py:21, in␣
 ↪References.__init__(self)
     18     bib_database=bibtexparser.bparser.BibTexParser(
     19     common_strings=True).parse_file(bibtex_file)
     20 self.bib_dict = {i['ID']:i for i in bib_database.entries}
---> 21 self.reflist = json.load(open(reflist))

File ~/opt/anaconda3/envs/env_picaso/lib/python3.9/json/__init__.py:293, in␣
 ↪load(fp, cls, object_hook, parse_float, parse_int, parse_constant,␣
 ↪object_pairs_hook, **kw)
    274 def load(fp, *, cls=None, object_hook=None, parse_float=None,
    275         parse_int=None, parse_constant=None, object_pairs_hook=None,␣
 ↪**kw):
    276     """Deserialize ``fp`` (a ``.read()``-supporting file-like object␣
 ↪containing
    277     a JSON document) to a Python object.
```

```
    278
   (…)
    291     kwarg; otherwise ``JSONDecoder`` is used.
    292     """
--> 293     return loads(fp.read(),
    294         cls=cls, object_hook=object_hook,
    295         parse_float=parse_float, parse_int=parse_int,
    296␣
 ↪        parse_constant=parse_constant, object_pairs_hook=object_pairs_hook, **kw)

File ~/opt/anaconda3/envs/env_picaso/lib/python3.9/json/__init__.py:346, in␣
 ↪loads(s, cls, object_hook, parse_float, parse_int, parse_constant,␣
 ↪object_pairs_hook, **kw)
    341     s = s.decode(detect_encoding(s), 'surrogatepass')
    343 if (cls is None and object_hook is None and
    344         parse_int is None and parse_float is None and
    345         parse_constant is None and object_pairs_hook is None and not kw :
--> 346     return _default_decoder.decode(s)
    347 if cls is None:
    348     cls = JSONDecoder

File ~/opt/anaconda3/envs/env_picaso/lib/python3.9/json/decoder.py:337, in␣
 ↪JSONDecoder.decode(self, s, _w)
    332 def decode(self, s, _w=WHITESPACE.match):
    333     """Return the Python representation of ``s`` (a ``str`` instance
    334     containing a JSON document).
    335
    336     """
--> 337     obj, end = self.raw_decode(s, idx=_w(s, 0).end())
    338     end = _w(s, end).end()
    339     if end != len(s):

File ~/opt/anaconda3/envs/env_picaso/lib/python3.9/json/decoder.py:353, in␣
 ↪JSONDecoder.raw_decode(self, s, idx)
    344 """Decode a JSON document from ``s`` (a ``str`` beginning with
    345 a JSON document) and return a 2-tuple of the Python
    346 representation and the index in ``s`` where the document ended.
   (…)
    350
    351 """
    352 try:
--> 353     obj, end = self.scan_once(s, idx)
    354 except StopIteration as err:
    355     raise JSONDecodeError("Expecting value", s, err.value) from None

JSONDecodeError: Expecting property name enclosed in double quotes: line 55␣
 ↪column 1 (char 2374)
```

[ ]: