Json Justify Documentation

Release 1.0.0

Akash Chaudhari

Oct 15, 2018

Contents

Json justify is used to create a structure of json python classes and then that structure can be used to validate json, track that all keys of structure are available or not, all keys have data as per structure created, render json etc.

CHAPTER 1

Contents

1.1 Installation

This project only supports python3 version python2 support may come in upcomming versions

1.1.1 via pip

```
# if under linux machiene
pip3 install justify

# if under windows
pip install justify
```

1.1.2 via github

```
#cloning git repo
git clone git@github.com:AngrySoilder/json-justify.git
cd json-justify
python3 setup.py install
```

1.2 Basic Usage

The basic usage of json_justify is shown here which is used to validate data from source

```
from json_justify import JsonManager
from json_justify.fields import String,Number,Boolean,Array

class Js(JsonManager):
```

(continues on next page)

(continued from previous page)

```
        name = String("name")
        age = Number("age")
        male = Boolean("male")
        friends = Array("friends")

data = {
        "name" : "john doe",
        "age" : 120,
        "male" : False,
        "friends" : ["Jelly","Kelly"]
        }
# This will return True
js = Js(data = data)
data.is_valid()
```

1.3 Json Manager

class json_justify.jason.**JsonManager**(*data*, *allow_extra=False*, *_child_hook=False*)

This is the main json field which ultimately any progarm will subclass to make it possible to create Json classes

Parameters

- **data** – data to validate
- **allow_extra** – If extra fields allowed or not
- **hook** (*_child*) – If it is Child or not

Object

alias of __builtin__.dict

add_render_machiene(*func*)

This is used to regester function for rendering if param is not callable then it will raise InvalidContainer

Parameters **func** –

Returns None

child

This property is used to check that if it is child json or not

Returns True or False

generate_otk_token()

This function is a dummy function which may be used to crete dummy functions for another projects

Returns

integral_types(*data*)

This is used to check if field data is in integral types or not

Parameters **data** (*any*) – value to chaeck

is_object(*value*)

This class is used to check if it is dir which is standard key value pair or not

Parameters **value** – directory

Returns True or False

is_valid()
This is used to check if data provided to JsonManager is actually valid or not Following Things Will be checked – Data type corrospondence – Good with validators

Returns True or False

items()
This method should be used to get all of the Field class keys inside JsonManager Class

Returns a list of Field

json_or_error()
json_or_error function should be to get json or error -json and returned to system and then rendered accordingly

Returns dict of error or render

regester_attris(*func*)
This is used to regester function which will be called on creation of class if param is not callable then it will raise InvalidMachiene

Parameters **func** – Callable function which returns tuple of key value pair or return value if function name you want to be key name

Returns None

regester_error(*name*, *value*)
This is used to regester error to the object and used to regester validation Error

Parameters

- **name** – name of error
- **value** – value of error

render_json()
This function is used as a master key to create json with registered rendered function and send it back as response

Returns dict(Kind of Json)

setup_fields()
This method filters out and setup field dictionary to work creates field dictionary and returns it

Returns a dictionary of fields

setup_json()
This is function which will be used to setup form if data and if not provided in any of its instances then it will register error

Raises Invalid if not valid data

1.4 Fields

1.4.1 Field

class json_justify.fields.**Field**(*field_name*, *validators=None*)
This Field Class is Core of all the Fields and should be instanciated For Creating New Fields This should only be used with JsonManager Field Following Things should be done in order to Create Field – Instanciate Field – Should implement _validate() function to which data of field will be provided Also one can change whole functionality by instanciating Field object as well

Parameters

- **field_name** (*str*) – Name of the field same as JsonManager Class Field
- **validators** (*list of callable take one param data*) – list of validators

data

Returns data associated with particular field

Raises Invalid Exception on delete of these property

register_error(*key*, *value*)

This Function is used to register error of the fields which is usually done by the validaors to register errors

Parameters

- **key** – key of error for json
- **value** – custom error message inside json

Returns None but registors error

1.4.2 String

class json_justify.fields.**String**(*field_name*, *validators=None*)

This is simple String Field of json Which should be used with json manager class

This field is automacally called inside JsonManager Class and validated if Invalid data is raised it will automatically catched by JsonManager Class

Parameters

- **field_name** (*str*) – Name of the field same as JsonManager Class Field
- **validators** (*list of callable take one param data*) – list of validators

1.4.3 Number

class json_justify.fields.**Number**(*field_name*, *validators=None*)

This is Number Field and Should be Used inside JsonManager Class to create numbers

This field will raise Invalid and automatically catched by JsonManager if Data to key is not int of float

Parameters

- **field_name** (*str*) – Name of the field same as JsonManager Class Field
- **validators** (*list of callable take one param data*) – list of validators

1.4.4 Boolean

class json_justify.fields.**Boolean**(*field_name*, *validators=None*)

This is Boolean Field and Should be Used inside JsonManager Class to create numbers

This field will raise Invalid and automatically catched by JsonManager if Data to key is not Boolean

Parameters

- **field_name** (*str*) – Name of the field same as JsonManager Class Field
- **validators** (*list of callable take one param data*) – list of validators

1.4.5 Array

class json_justify.fields.**Array**(*field_name*, *min_len=-1*, *max_len=-1*, *js_model=None*, *validators=None*, *seq_validators=None*)

This is simple Array Field and should be used with JsonManager Class to create simple Arrays and also objects inside array

Only one of three paremeters from js_model, validators, seq_validators can be choosed at a time Two Dimentional or N-Dimentional Array May be Implemented inside later Version of This

Parameters

- **field_name** (*str*) – Name of the field same as JsonManager Class Field
- **validators** (*list of callable take one param data*) – list of validators
- **js_model** – a JsonManager Class that should be used to create array of key json
- **seq_validators** – A sequence of validators to validate each sequence of validator

if choosed length of array is seq_validator

1.5 Validators

1.5.1 Validator

class json_justify.validators.**Validator**(*message=None*)

This is core of each validator class

Parameters **message** (*str*) – Message to be raised when any invalidation occours

1.5.2 Data

class json_justify.validators.**Data**(*message=None*)

This is Data field to check that weather data is provided or not

Parameters **message** (*str*) – Message to be raised when any invalidation occours

1.5.3 Length

class json_justify.validators.**Length**(*min_val=-1*, *max_val=-1*, *message=None*)

This validator is used to check the minimum and maximum Length of data This will work with str, int , float not with array length

Parameters

- **min_val** (*int*) – minimum length
- **max_val** (*int*) – maximum length
- **message** (*str*) – Message to be raised when any invalidation occours

1.5.4 Email

class json_justify.validators.**Email**(*message=None*)

The Email Validator is used to validate Email which use dependency of python email_validator to validate email if not installed on machiene it will used standard regex to validate email

Parameters **message** (*str*) – Message to be raised when any invalidation occours

1.5.5 URL

class json_justify.validators.**URL**(*message=None*)

The url validator is used to validate url's But it dones not instanciate from Regex class here this is another implementation

Parameters **message** (*str*) – Message to be raised when any invalidation occours

1.5.6 EqualTo

class json_justify.validators.**EqualTo**(*field*, *message=None*)

The EqualTo Validator is used to check if data in our field is same as other field

Parameters

- **field** (*Field class*) – Other field Pass the whole Field
- **message** (*str*) – Message to be raised when any invalidation occours

1.5.7 Date

class json_justify.validators.**Date**(*min_date=datetime.datetime(1, 1, 1, 0, 0)*, *max_date=datetime.datetime(1, 1, 1, 0, 0)*, *field_format=None*, *message=None*)

The date field validator is used to validate date field under specific range of min_date and max_date

Parameters

- **min_date** (*str of year-month-date*) – minimum date
- **max_date** (*str of year-month-date*) – maximum date
- **field_format** (*Field format %Y-%M-%d is default*) – format of field
- **message** (*str*) – message

1.5.8 Regex

class json_justify.validators.**Regex**(*reg*, *message=None*)

This is used to validate data in format of regex

Parameters

- **regex** (*str*) – regex
- **message** (*str*) – Message to be raised when Error occours

1.5.9 Right

class json_justify.validators.**Right**(*message=None*)
This Validator is specific to boolean Field which validates that data is True only

Parameters **message** (*str*) – message to be raise when error occours

1.5.10 Wrong

class json_justify.validators.**Wrong**(*message=None*)
This Validator is specific to boolean Field which validates that data is False only

Parameters **message** (*str*) – message to be raise when error occours

1.6 Utilities

1.6.1 Json Manager Utilities

json_justify.jason.**keymapper**(*dict_like*)
This is used to create Json From rendered functions Internally

Parameters **dict_like** (*list*) – list of functions

Returns dict of rendered functions

json_justify.jason.**addupdict**(**args*)
This will be used to make summiton of dictionary keys

Parameters **args** – dicts

Returns Added dictionaries

json_justify.jason.**render_factory**(*js*, *render_tup*)
This is used to register tuples of function to js rendering

Parameters

- **js** – JsonManager class
- **render_tup** – Tuple of Renderjs

Returns None

class json_justify.utils.**ErrorResponse**(*response=None*)
This class will be used to create error response but it will be used in later implementation of module

class json_justify.utils.**Level**(*level*)
This class will be used to set Level of runtime :Example: level = Level(Level.DEBUG)

Parameters **level** – int

level
This is used to set level to work with default levels are as follow DEGUG PRODUCTION

Returns Int

class json_justify.utils.**PlaceHolder**
This is placeholder which is used when no data is set

1.7 Errors

1.7.1 Invalid

class json_justify.validators.**Invalid**(*message*, **args*, ***kwargs*)

This Exception is General Exception used to raise when Data is invalid, Not a valid Type etc.

Parameters

- **message** (*str*) – Message you want to print in error
- **args** – any
- **kwargs** – any

1.7.2 InvalidMachiene

class json_justify.jason.**InvalidMachiene**(*message*)

This is InvalidMachiene class which is used to raise Exception when Invalid function is registered to attris

Parameters **message** (*str*) – message to be raise when error occours

1.7.3 InvalidContainer

class json_justify.jason.**InvalidContainer**(*message*)

This is InvalidMachiene class which is used to raise Exception when Invalid function is registered to render

Parameters **message** (*str*) – message to be raise when error occours

1.8 License

1.9 Changelog

1.9.1 Currently in 1.0.0

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

j

Index

U

V

W