Package 'ajv'

April 24, 2017

Title Another JSON Schema Validator

Version 1.0.0

Maintainer Jason Thorpe <jdthorpe@gmail.com>

Description A thin wrapper around the 'ajv' JSON validation package for JavaScript. See <http://epoberezkin.github.io/ajv/> for details.

License GPL-2

LazyData true

URL https://github.com/jdthorpe/ajvr

BugReports https://github.com/jdthorpe/ajvr/issues

Imports V8, yaml, RJSONIO

Suggests knitr, rmarkdown, testthat

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Jason Thorpe [aut, cre]

Repository CRAN

Date/Publication 2017-04-24 16:23:22 UTC

R topics documented:

`Ajv` *Create an Ajv instnace.*

Description

Create an Ajv instnace (the equivalent of calling `new Ajv()` in javascript)

Usage

```
Ajv(options = NULL)
```

Arguments

`options`	Optional; see the ajv github page for details

`ajv.addFormat` *A wrapper for the Ajv.addFormat method*

Description

Add a string format to an Ajv instance.

Usage

```
ajv.addFormat(this, key, format)
```

Arguments

`this`	An AJV instance, provided implicitly when called via `my_instance$addFormat(...)`
`key`	String; the name with format to add.
`format`	the format to be added. Note that JavaScript object literals should be enclosed in a call to `JS`. (i.e. `ajv$addFormat("numbers",JS("/^\\d+$/"))`)

Value

```
invisible(NULL)
```

See Also

Other AJV.Instance.Methods: `ajv.addKeyword`, `ajv.addSchema`, `ajv.errorsText`, `ajv.keyword`, `ajv.removeSchema`, `ajv.validateSchema`, `ajv.validate`

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$keyword(key,object)

## End(Not run)
```

`ajv.addKeyword` *A wrapper for the Ajv.addKeyword method*

Description

The add a schema to an Ajv instance

Usage

```
ajv.addKeyword(this, name, definition)
```

Arguments

`this`	An AJV instance, provided implicitly when called via `my_instance$addSchema(...)`
`name`	The name of the keyword to be added.
`definition`	A string encoding of a javascript object to be used as to define the keyword.

Value

```
invisible(NULL)
```

See Also

Other AJV.Instance.Methods: `ajv.addFormat`, `ajv.addSchema`, `ajv.errorsText`, `ajv.keyword`, `ajv.removeSchema`, `ajv.validateSchema`, `ajv.validate`

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$addSchema

## End(Not run)
```

`ajv.addSchema` *A wrapper for the Ajv.addSchema method*

Description

The add a schema to an Ajv instance

Usage

```
ajv.addSchema(this, schema, key)
```

Arguments

`this`	An AJV instance, provided implicitly when called via `my_instance$addSchema(...)`
`schema`	The schema to be added. `schema` may be a valid JSON string, an R object (i.e. 'list(...)'), a connection to a JSON file, or the name of a JSON or YAML file. YAML files are parsed via [js-yaml](https://www.npmjs.com/package/js-yaml)'s 'safeLoad()' method.
`key`	String; the name with which to store the schema

Value

```
invisible(NULL)
```

See Also

Other AJV.Instance.Methods: `ajv.addFormat`, `ajv.addKeyword`, `ajv.errorsText`, `ajv.keyword`, `ajv.removeSchema`, `ajv.validateSchema`, `ajv.validate`

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$addSchema

## End(Not run)
```

ajv.compile *The Ajv.compile method*

Description

Create an Ajv validator function from a schema

Usage

```
ajv.compile(this, schema)
```

Arguments

this	An AJV instance, provided implicitly when called via `my_instance$compile(...)`
schema	The Schema with which to validate the `data`. `schema` may be a valid JSON string, an R object (i.e. 'list(...)'), a connection to a JSON file, or the name of a JSON or YAML file. YAML files are parsed via [js-yaml](https://www.npmjs.com/package/js-yaml)'s 'safeLoad()' method.

Value

an AJV validation function

ajv.errorsText *A wrapper for the Ajv.errorsText method*

Description

Extracts the errors object from

Usage

```
ajv.errorsText(this)
```

Arguments

this	An AJV instance, provided implicitly when called via `my_instance$errorsText(...)`

Value

JSON encoded object containing the error message (if any), with class "AJV-errorsText" for pretty printing via `print.errorsText`

See Also

Other AJV.Instance.Methods: `ajv.addFormat`, `ajv.addKeyword`, `ajv.addSchema`, `ajv.keyword`, `ajv.removeSchema`, `ajv.validateSchema`, `ajv.validate`

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$errorsText

## End(Not run)
```

`ajv.getSchema` *The Ajv.compile method*

Description

Create an Ajv validator function from a schema

Usage

```
ajv.getSchema(this, key)
```

Arguments

`this`	An AJV instance, provided implicitly when called via `my_instance$getSchema(...)`
`key`	String; the name of the schema to fetch from the Ajv instancd.

Value

an AJV validation function

`ajv.keyword` *A wrapper for the Ajv.addFormat method*

Description

Add a string format to an Ajv instance.

Usage

```
ajv.keyword(this, key, object)
```

Arguments

`this`	An AJV instance, provided implicitly when called via `my_instance$keyword(...)`
`key`	String; the name with keyword to add.
`object`	the format to be added. Must be enclosed in a call to `JS`.

Value

invisible(NULL)

See Also

Other AJV.Instance.Methods: ajv.addFormat, ajv.addKeyword, ajv.addSchema, ajv.errorsText, ajv.removeSchema, ajv.validateSchema, ajv.validate

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$keyword(key,object)

## End(Not run)
```

---

ajv.removeSchema *A wrapper for the Ajv.removeSchema method*

Description

The remove a schema from an Ajv instance

Usage

```
ajv.removeSchema(this, key)
```

Arguments

this	An AJV instance, provided implicitly when called via my_instance$removeSchema(...)
key	String; the name with schema to remove

Value

invisible(NULL)

See Also

Other AJV.Instance.Methods: ajv.addFormat, ajv.addKeyword, ajv.addSchema, ajv.errorsText, ajv.keyword, ajv.validateSchema, ajv.validate

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$removeSchema

## End(Not run)
```

ajv.validate *A wrapper for the Ajv.validate method*

Description

The equivalent of calling `var ajv = new Ajv(); ajv.validate(...)` in javascript.

Usage

```
ajv.validate(this, schema, data)
```

Arguments

this	An AJV instance, provided implicitly when called via `my_instance$validate(...)`
schema	The Schema with which to validate the `data`. `schema` may be a valid JSON string, an R object (i.e. 'list(...)'), a connection to a JSON file, or the name of a JSON or YAML file. YAML files are parsed via [js-yaml](https://www.npmjs.com/package/js-yaml)'s 'safeLoad()' method.
data	The data to be validated. may be any of the above foremats.

See Also

Other AJV.Instance.Methods: `ajv.addFormat`, `ajv.addKeyword`, `ajv.addSchema`, `ajv.errorsText`, `ajv.keyword`, `ajv.removeSchema`, `ajv.validateSchema`

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$validate

## End(Not run)
```

ajv.validateSchema *A wrapper for the Ajv.validateSchema method*

Description

The validate a json schema

Usage

```
ajv.validateSchema(this, schema)
```

Arguments

this	An AJV instance, provided implicitly when called via my_instance$ajv(...)
schema	The Schema to be validated. schema may be a valid JSON string, an R object (i.e. 'list(...)'), a connection to a JSON file, or the name of a JSON or YAML file. YAML files are parsed via [js-yaml](https://www.npmjs.com/package/js-yaml)'s 'safeLoad()' method.

Value

boolean

See Also

Other AJV.Instance.Methods: ajv.addFormat, ajv.addKeyword, ajv.addSchema, ajv.errorsText, ajv.keyword, ajv.removeSchema, ajv.validate

Examples

```
## Not run:
my_ajv_instance = Ajv()
my_ajv_instance$validateSchema

## End(Not run)
```

Index