FastscoreRSDK

April 15, 2019

Type Package

Title Fastscore R SDK (GoVersion)

Version 0.1.0

Discription package for Fastscore R integration, using Fastscore GoSDK

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

License TBD

Suggests knitr,
      rmarkdown

VignetteBuilder knitr

R topics documented:

Attachment_download   *Download an attachment to a local path*

Description

Download an attachment to a local path

Usage

```
Attachment_download(model_name, att_name, att_path)
```

Arguments

model_name	model name
att_name	attachment name
att_path	path of the attachment (inluding file name)

Value

operation message or error message

Attachment_list *List all attchments of a model*

Description

List all attchments of a model

Usage

```
Attachment_list(model_name)
```

Arguments

model_name   name of the model to check for attachments

Value

list of names or error message

Attachment_remove *Remove an attachment from a model*

Description

Remove an attachment from a model

Usage

```
Attachment_remove(model_name, att_name)
```

Arguments

model_name   model name

att_name   attachment name

Value

operation message or error message

Attachment_upload   *Upload an attqchment to a model from local path*

Description

Upload an attqchment to a model from local path

Usage

```
Attachment_upload(model_name, att_name, att_path)
```

Arguments

`model_name`	model name
`att_name`	attachment name
`att_path`	path of the attachment (inluding file name)

Value

operation message or error message

Composer_config_set   *Set composer configuration from a local file*

Description

Set composer configuration from a local file

Usage

```
Composer_config_set(path)
```

Arguments

`path`	path of the composer configuration file

Value

operation message or error message

Composer_config_show   *Show composer configuration*

Description

Show composer configuration

Usage

```
Composer_config_show()
```

Value

instance info or error message

Composer_create   *Create a composer from a local file*

Description

Create a composer from a local file

Usage

```
Composer_create(name, path)
```

Arguments

name	composer name
path	path of the composer file (including file name)

Value

operation message or error message

Composer_delete	*Delete a composer*

Description

Delete a composer

Usage

```
Composer_delete(name)
```

Arguments

name   name of the composer to delete

Value

operation message or error message

Composer_list	*List all composers*

Description

List all composers

Usage

```
Composer_list()
```

Value

list of names or error message

Composer_show	*Show content of a composer*

Description

Show content of a composer

Usage

```
Composer_show(name)
```

Arguments

name   composer name

Value

instance info or error message

Config_set   *Set configuration from a local file*

Description

Set configuration from a local file

Usage

```
Config_set(path)
```

Arguments

path   path of the configuration file

Value

operation message or error message

Config_show   *Show configuration of Fastscore*

Description

Show configuration of Fastscore

Usage

```
Config_show()
```

Value

instance info or error message

Connect   *Connect to Fastscore, set proxy_prefix*

Description

Connect to Fastscore, set proxy_prefix

Usage

```
Connect(proxy_prefix)
```

Arguments

proxy_prefix   prefix of connect

Value

proxy prefix + built date or error message

Engine_inspect   *Inspect an engine*

Description

Inspect an engine

Usage

```
Engine_inspect(name)
```

Arguments

name   engine name

Value

instance info or error message

Engine_pause   *Pause an engine*

Description

Pause an engine

Usage

```
Engine_pause(name)
```

Arguments

name   engine name

Value

operation message or error message

Engine_reset   *Reset an engine, or reset all engines*

Description

Reset an engine, or reset all engines

Usage

```
Engine_reset(name_or_all)
```

Arguments

name   engine name, "all" to reset all engines

Value

operation message or error message

Engine_unpause   *Unpause an engine*

Description

Unpause an engine

Usage

```
Engine_unpause(name)
```

Arguments

name   engine name

Value

operation message or error message

Get_Fleet   *Show all fleets and health*

Description

Show all fleets and health

Usage

```
Get_Fleet()
```

Value

list of name + health or error message

import	*import a list of certain instances from a given directory*

Description

import a list of certain instances from a given directory

Usage

```
import(namelist, type, dir)
```

Arguments

namelist	a list of instance names
type	instance type ("model", "schema", "stream")
dir	directory path (example: ".../fastscore/library/")

Value

operation message or error message

Examples

```
import(list("array-double", "string"), "schema", "fastscore/library/")
```

Model_add	*Add a model from local file*

Description

Add a model from local file

Usage

```
Model_add(name, path)
```

Arguments

name	model name
path	model path (including file name)

Value

operation message or error message

Model_inspect   *Inspect a model*

Description

Inspect a model

Usage

```
Model_inspect(name)
```

Arguments

name   model name

Value

instance info or error message

Model_list   *List all models*

Description

List all models

Usage

```
Model_list()
```

Value

list of names or error message

Model_load   *Load a model to an engine*

Description

Load a model to an engine

Usage

```
Model_load(model_name, engine_name)
```

Arguments

model_name   model name
engine_name   model name

Value

operation message or error message

Model_remove *Remove a model*

Description

Remove a model

Usage

```
Model_remove(name)
```

Arguments

name   model name

Value

operation message or error message

Model_show *Show content of a model*

Description

Show content of a model

Usage

```
Model_show(name)
```

Arguments

name   model name

Value

instance info or error message

remove_all   *remove all model/stream/schema loaded*

Description

remove all model/stream/schema loaded

Usage

```
remove_all(type)
```

Arguments

type   type of instance to clear ("model", "schema", "stream")

Value

operation message or error message

Examples

```
remove_all("model")
```

Run   *Run a model in an engine, need to specify input and output streams*

Description

Run a model in an engine, need to specify input and output streams

Usage

```
Run(engine_name, model_name, stream0_name, stream1_name)
```

Arguments

engine_name   engine name

model_name   model name

stream0_name input stream name

stream1_name output stream name

Value

operation message or error message

Schema_add   *Add a schema from local path*

Description

Add a schema from local path

Usage

```
Schema_add(name, path)
```

Arguments

name   schema name

path   path of file (including file name)

Value

operation message or error message

Schema_list   *List all schemas*

Description

List all schemas

Usage

```
Schema_list()
```

Value

list of names or error message

Schema_remove   *Remove a schema*

Description

Remove a schema

Usage

```
Schema_remove(name)
```

Arguments

name	schema name

Value

operation message or error message

Schema_show   *Show content of a schema*

Description

Show content of a schema

Usage

```
Schema_show(p0)
```

Value

instance info or error message

Sensor_add   *Add a sensor from local path*

Description

Add a sensor from local path

Usage

```
Sensor_add(name, path)
```

Arguments

name	sensor name
path	sensor file path (including file name)

Value

operation message or error message

Sensor_inspect   *Inspect a sensor*

Description

Inspect a sensor

Usage

```
Sensor_inspect(name)
```

Arguments

name   sensor name

Value

instance info or error message

Sensor_install   *Install a sensor*

Description

Install a sensor

Usage

```
Sensor_install(name)
```

Arguments

name   sensor name

Value

operation message or error message

Sensor_list *List all sensors*

Description

List all sensors

Usage

```
Sensor_list()
```

Value

list of names or error message

Sensor_points *List sensor points*

Description

List sensor points

Usage

```
Sensor_points()
```

Value

list of names or error message

Sensor_remove *Remove a sensor*

Description

Remove a sensor

Usage

```
Sensor_remove(name)
```

Arguments

name   sensor name

Value

operation message or error message

Sensor_show   *Show content of a sensor*

Description

Show content of a sensor

Usage

```
Sensor_show(name)
```

Arguments

name   schema name

Value

instance info or error message

Sensor_uninstall   *Uninstall a sensor*

Description

Uninstall a sensor

Usage

```
Sensor_uninstall(name)
```

Arguments

name   sensor name

Value

operation message or error message

Stream_add *Add a stream from local file*

Description

Add a stream from local file

Usage

```
Stream_add(name, path)
```

Arguments

name	stream name
path	stream file path (including file name)

Value

operation message or error message

Stream_attach *Attach a stream to an engine, specify slot*

Description

Attach a stream to an engine, specify slot

Usage

```
Stream_attach(slot, stream_name, engine_name)
```

Arguments

slot	attach slot number
stream_name	stream name
engine_name	engine name

Value

operation message or error message

Stream_inspect   *Inspect a stream*

Description

Inspect a stream

Usage

```
Stream_inspect(stream_name, engine_name)
```

Arguments

stream_name   stream name

engine_name   engine name

Value

instance info or error message

Stream_list   *List all streams*

Description

List all streams

Usage

```
Stream_list()
```

Value

list of names or error message

Stream_remove *Remove a stream*

Description

Remove a stream

Usage

```
Stream_remove(name)
```

Arguments

name   stream name

Value

operation message or error message

Stream_sample *Get samples from a stream, specify count*

Description

Get samples from a stream, specify count

Usage

```
Stream_sample(stream_name, engine_name, count)
```

Arguments

stream_name   stream name

engine_name   engine name

count   number of samples

Value

samples or error message

Stream_show   *Show content of a stream*

Description

Show content of a stream

Usage

```
Stream_show(name)
```

Arguments

name   stream name

Value

instance info or error message

Use   *Use a certain engine*

Description

Use a certain engine

Usage

```
Use(engine_name)
```

Arguments

engine_name   engine name

Value

operation message or error message

Index