Package ‘promote’

October 14, 2022

Type Package

Title Client for the 'Alteryx Promote' API

Version 1.1.1

Date 2019-02-05

Author Paul E. Promote <promotedev@alteryx.com>

Maintainer Paul E. Promote <promotedev@alteryx.com>

Description Deploy, maintain, and invoke predictive models using the 'Alteryx Promote' REST API. 'Alteryx Promote' is available at the URL: <https://www.alteryx.com/products/alteryx-promote>.

Depends R (>= 3.2.0)

URL https://github.com/alteryx/promote-r-client

Imports httr, jsonlite, stringr,

License FreeBSD

RoxygenNote 6.1.0

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-06 14:30:03 UTC

R topics documented:

- add.dependency .......................................................... 2
- add.metadata .............................................................. 3
- capture.src .............................................................. 3
- check.image.size ........................................................ 3
- is.https ................................................................. 4
- promote.deploy ........................................................ 4
- promote.get ............................................................ 5
- promote.library ......................................................... 5
- promote.ls ............................................................... 6
Private function that adds a package to the list of dependencies that will be installed on the Promote server

Usage

add.dependency(name, importName, src, version, install, auth_token, ref, subdir)

Arguments

name
importName
src
version
install
auth_token
ref
subdir

name of the package to be installed
name under which the package is imported (for a github package, this may be different from the name used to install it)
source that the package is installed from (CRAN or github)
version of the package
whether or not the package should be installed in the model image
a personal access token for github or gitlab repositories
The git branch, tag, or SHA of the package to be installed
The path to the repo subdirectory holding the package to be installed
add.metadata

Private function that adds metadata about the model that will be installed on the Promote server. The metadata is arranged as key-value pairs.

Usage

add.metadata(key, value)

Arguments

key : key name for the metadata entry
value : value for the metadata entry

capture.src

Private function for capturing the source code of the model.

Usage

capture.src(funcs, capture.model.require = TRUE)

Arguments

funcs : functions to capture, defaults to required promote model functions
capture.model.require : flag to capture the model.require function

check.image.size

Private function for checking the size of the user's image.

Usage

check.image.size()
is.https  
*Private predicate function that checks if the protocol of a url is https.*

**Description**

Private predicate function that checks if the protocol of a url is https.

**Usage**

```r
is.https(x)
```

**Arguments**

- `x`  
  is a url string

---

promote.deploy  
*Deploy a model to promote's servers*

**Description**

This function takes `model.predict` and creates a model on promote’s servers which can be called from any programming language via promote’s REST API (see `promote.predict`).

**Usage**

```r
promote.deploy(model_name, confirm = TRUE, custom_image = NULL)
```

**Arguments**

- `model_name`  
  name of your model
- `confirm`  
  boolean indicating whether to prompt before deploying
- `custom_image`  
  name of the image you’d like your model to use

**Examples**

```r
promote.config <- c(
  username = "your username",
  apikey = "your apikey",
  env = "http://sandbox.promotehq.com/"
)
iris$Sepal.Width_sq <- iris$Sepal.Width^2
fit <- glm(I(Species) == "virginica" ~ ., data = iris)

model.predict <- function(df) {
  data.frame("prediction" = predict(fit, df, type = "response"))
}
```
### Description

Private function for performing a GET request

### Usage

```r
promote.get(endpoint, query = c())
```

### Arguments

- **endpoint**: /path for REST request
- **query**: url parameters for request

### Description

Import one or more libraries and add them to the promote model’s dependency list

### Usage

```r
promote.library(name, src = "version", version = NULL, user = NULL,
                install = TRUE, auth_token = NULL, url = NULL, ref = "master",
                subdir = NULL)
```

### Arguments

- **name**: name of the package to be added
- **src**: source from which the package will be installed on Promote (github or CRAN)
- **version**: version of the package to be added
- **user**: Github username associated with the package
- **install**: Whether the package should also be installed into the model on the Promote server; this is typically set to False when the package has already been added to the Promote base image.
auth_token  Personal access token string associated with a private package’s repository
url          A valid URL pointing to a remote hosted git repository
ref          The git branch, tag, or SHA of the package to be installed
subdir       The path to the repo subdirectory holding the package to be installed

Examples

```r
## Not run:
promote.library("MASS")
promote.library(c("wesanderson", "stringr"))
promote.library("hilaryparker/cats")
promote.library("cats", src="github", user="hilaryparker")
promote.library("my_public_package", install=FALSE)
promote.library("my_public_package", src="git", url="https://gitlab.com/username/rpkg.git")
promote.library("my_private_package", src="github", auth_token=<yourToken>)
promote.library("testPkg", src="github", user="emessess", auth_token=<yourToken>)
promote.library("priv_pkg", src="git",
               url="https://x-access-token:<PersonalAccessToken>@github.com/username/rpkg.git")
promote.library("priv_pkg", src="git",
               url="https://x-access-token:<PersonalAccessToken>@gitlab.com/username/rpkg.git",
               ref="stage")
promote.library("my_package", src="github", auth_token=<yourToken> subdir="/pathToSubdir/"

## End(Not run)
```

---

**promote.ls**  

*Private function for determining model dependencies*

Description

List all object names which are dependencies of and `model.predict`.

Usage

`promote.ls()`
promote.metadata  Add metadata to the deployment of your promote model

Description
Add metadata to the deployment of your promote model

Usage
promote.metadata(name, value)

Arguments
name  key name for the metadata entry
value  value for the metadata entry

Examples
## Not run:
promote.metadata("key", "value")
promote.metadata("R_squared", summary(fit)$r.squared)
promote.metadata("R_squared_adj", summary(fit)$adj.r.squared)
promote.metadata("deploy_node", Sys.info()[["nodename"]])
## End(Not run)

promote.post  Private function for performing a POST request

Description
Private function for performing a POST request

Usage
promote.post(endpoint, query = c(), data, silent = TRUE,
bulk = FALSE)

Arguments
endpoint  /path for REST request
query  url parameters for request
data  payload to be converted to raw JSON
silent  should output of url to console be silenced? Default is FALSE.
bulk  is this a bulk style request? Default is FALSE.
promote.predict Make a prediction using promote.

Description

This function calls promote’s REST API and returns a response formatted as a data frame.

Usage

promote.predict(model_name, data, model_owner, raw_input = FALSE, silent = TRUE)

Arguments

- model_name: the name of the model you want to call
- data: input data for the model
- model_owner: the owner of the model [optional]
- raw_input: when true, incoming data will NOT be coerced into data.frame
- silent: should output of url to console (via promote.post) be silenced? Default is FALSE.

Examples

```r
promote.config <- c(
  username = "your username",
  apikey = "your apikey",
  env = "http://sandbox.promotehq.com/"
)
## Not run:
promote.predict(irisModel, iris)
## End(Not run)
```

promote.predict_raw Calls promote’s REST API and returns a JSON document containing both the prediction and associated metadata.

Description

Calls promote’s REST API and returns a JSON document containing both the prediction and associated metadata.

Usage

promote.predict_raw(model_name, data, model_owner, raw_input = FALSE, silent = TRUE)
promote.spider.block

Arguments

- model_name: the name of the model you want to call
- data: input data for the model
- model_owner: the owner of the model [optional]
- raw_input: when true, incoming data will NOT be coerced into data.frame
- silent: should output of url to console (via promote.post) be silenced? Default is FALSE.

Examples

```r
promote.config <- c(
  username = "your username",
  apikey = "your apikey",
  env="http://ip_of_alteryx_promote.com"
)
## Not run:
promote.predict_raw("irisModel", iris)
## End(Not run)
```

promote.spider.block  Private function for recursively looking for variables

Description

Private function for recursively looking for variables

Usage

```r
promote.spider.block(block, defined.vars = c())
```

Arguments

- block: code block to spider
- defined.vars: variables which have already been defined within the scope of the block. e.g. function argument
### promote.spider.func
*Private function for spidering function source code*

**Description**

Private function for spidering function source code

**Usage**

`promote.spider.func(func.name)`

**Arguments**

- `func.name` name of function you want to spider

---

### promote.unload
*Removes a library from the promote model’s dependency list*

**Description**

Removes a library from the promote model’s dependency list

**Usage**

`promote.unload(name)`

**Arguments**

- `name` of the package to be removed

**Examples**

```r
## Not run:
promote.unload("wesanderson")
## End(Not run)
```
set.model.require

| set.model.require | Private function that generates a model.require function based on the libraries that have been imported in this session. |

**Description**

Private function that generates a model.require function based on the libraries that have been imported in this session.

**Usage**

```python
set.model.require()
```
Index

* **deploy**
  promote.deploy, 4

* **import**
  promote.library, 5

* **metadata**
  promote.metadata, 7

* **predict**
  promote.predict, 8

add.dependency, 2
add.metadata, 3

capture.src, 3
check.image.size, 3

is.https, 4
promote.deploy, 4
promote.get, 5
promote.library, 5
promote.ls, 6
promote.metadata, 7
promote.post, 7
promote.predict, 4, 8
promote.predict_raw, 8
promote.spider.block, 9
promote.spider.func, 10
promote.unload, 10

set.model.require, 11