Package: yhatr (via r-universe)

October 13, 2024

Type Package

Title R Binder for the Yhat API

Version 0.15.1

Date 2016-08-29

Author Greg Lamp <greg@yhathq.com>, Stevie Smith <dev@yhathq.com>, Ross Kippenbrock <ross@yhathq.com>

Maintainer Greg Lamp <greg@yhathq.com>

Description Deploy, maintain, and invoke models via the Yhat REST API.

Depends R (>= 2.12.0)

URL https://github.com/yhat/yhatr

Imports httr, jsonlite, stringr,

License FreeBSD

RoxygenNote 6.0.1

Repository https://yhat.r-universe.dev

RemoteUrl https://github.com/yhat/yhatr

RemoteRef HEAD

RemoteSha 198d0444bd43d3777f68c6eb1e11074fe1ce0b90

Contents

`add.dependency` *Private function that adds a package to the list of dependencies that will be installed on the ScienceOps server*

Description

Private function that adds a package to the list of dependencies that will be installed on the ScienceOps server

Usage

```
add.dependency(name, importName, src, version, install)
```

Arguments

`name`	name of the package to be installed
`importName`	name under which the package is imported (for a github package, this may be different from the name used to install it)
`src`	source that the package is installed from (CRAN or github)
`version`	version of the package
`install`	whether or not the package should be installed in the model image

---

`capture.src` *Private function for catpuring the source code of model*

Description

Private function for catpuring the source code of model

Usage

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

Arguments

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

---

check.dependencies *Checks dependencies and makes sure all are installed.*

Description

Checks dependencies and makes sure all are installed.

Usage

```
check.dependencies()
```

---

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

Description

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

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

Arguments

x	is a url string

`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

```
set.model.require()
```

`yhat.batchDeploy` *Deploy a batch model to Yhat servers*

Description

This function will deploy your batch model to the yhat servers

Usage

```
yhat.batchDeploy(job_name, confirm = TRUE)
```

Arguments

`job_name`	name of batch job
`confirm`	boolean indicating whether to prompt before deploying

Examples

```
yhat.config <- c(
 username = "your username",
 apikey = "your apikey",
 env = "http://sandbox.yhathq.com/"
)
yhat.batch <- function() {
  name <- "ross"
  greeting <- paste("Hello", name)
  print(greeting)
}
## Not run:
yhat.batchDeploy("helloworld")

## End(Not run)
```

yhat.deploy *Deploy a model to Yhat's servers*

Description

This function takes model.transform and model.predict and creates a model on Yhat's servers which can be called from any programming language via Yhat's REST API (see yhat.predict).

Usage

```
yhat.deploy(model_name, packages = c(), confirm = TRUE,
  custom_image = NULL)
```

Arguments

model_name	name of your model
packages	list of packages to install using apt-get
confirm	boolean indicating whether to prompt before deploying
custom_image	name of the image you'd like your model to use

Examples

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

model.require <- function() {
 # require("randomForest")
}

model.transform <- function(df) {
 df$Sepal.Width_sq <- df$Sepal.Width^2
 df
}
model.predict <- function(df) {
 data.frame("prediction"=predict(fit, df, type="response"))
}
## Not run:
yhat.deploy("irisModel")
yhat.deploy("irisModelCustomImage", custom_image="myImage:latest")

## End(Not run)
```

yhat.get — *Private function for performing a GET request*

Description

Private function for performing a GET request

Usage

```
yhat.get(endpoint, query = c())
```

Arguments

endpoint	/path for REST request
query	url parameters for request

yhat.library — *Import one or more libraries and add them to the Yhat model's dependency list*

Description

Import one or more libraries and add them to the Yhat model's dependency list

Usage

```
yhat.library(name, src = "CRAN", version = NULL, user = NULL,
  install = TRUE)
```

Arguments

name	name of the package to be added
src	source from which the package will be installed on ScienceOps (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 ScienceOps server; this is typically set to False when the package has already been added to the ScienceOps base image.

Examples

```
## Not run:
yhat.library("MASS")
yhat.library(c("wesanderson", "stringr"))
yhat.library("cats", src="github", user="hilaryparker")
yhat.library("hilaryparker/cats")
yhat.library("my_proprietary_package", install=FALSE)

## End(Not run)
```

`yhat.ls` *Private function for determining model dependencies*

Description

List all object names which are dependencies of ‘model.transform‘ and ‘model.predict‘ or ‘yhat.batch‘ if this is a batch mode deploy

Usage

```
yhat.ls(batchMode = FALSE)
```

Arguments

`batchMode`	boolean to capture yhat.batch code for a batch job

`yhat.post` *Private function for performing a POST request*

Description

Private function for performing a POST request

Usage

```
yhat.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`.

yhat.predict *Make a prediction using Yhat.*

Description

This function calls Yhat's REST API and returns a response formatted as a data frame.

Usage

```
yhat.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 yhat.post) be silenced? Default is FALSE.

Examples

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

## End(Not run)
```

yhat.predict_bulk *Make bulk predictions using Yhat.*

Description

This function calls Yhat's bulk API and returns a response formatted as a data frame.

Usage

```
yhat.predict_bulk(model_name, data, model_owner, raw_input = FALSE,
  silent = TRUE)
```

Arguments

model_name	the name of the model you want to call
data	input rows of data to be scored
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 yhat.post) be silenced? Default is FALSE.

Examples

```
yhat.config <- c(
 username = "your username",
 apikey = "your apikey",
 env = "http://sandbox.yhathq.com/"
)
## Not run:
yhat.predict_bulk("irisModel", iris)

## End(Not run)
```

---

yhat.predict_raw *Calls Yhat's REST API and returns a JSON document containing both the prediction and associated metadata.*

Description

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

Usage

```
yhat.predict_raw(model_name, data, model_owner, raw_input = FALSE,
  silent = TRUE, bulk = FALSE)
```

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 yhat.post) be silenced? Default is FALSE.
bulk	should the bulk api be used Default is FALSE.

Examples

```
yhat.config <- c(
 username = "your username",
 apikey = "your apikey"
)
## Not run:
yhat.predict_raw("irisModel", iris)

## End(Not run)
```

`yhat.spider.block` *Private function for recursively looking for variables*

Description

Private function for recursively looking for variables

Usage

```
yhat.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

---

`yhat.spider.func` *Private function for spidering function source code*

Description

Private function for spidering function source code

Usage

```
yhat.spider.func(func.name)
```

Arguments

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

yhat.unload *Removes a library from the Yhat model's dependency list*

Description

Removes a library from the Yhat model's dependency list

Usage

```
yhat.unload(name)
```

Arguments

name	of the package to be removed

Examples

```
## Not run:
yhat.unload("wesanderson")

## End(Not run)
```

yhat.verify *Private function for verifying username and apikey*

Description

Private function for verifying username and apikey

Usage

```
yhat.verify()
```

Index