

# Package ‘yhatr’

May 9, 2017

**Type** Package

**Title** R Binder for the Yhat API

**Version** 0.15.1

**Date** 2017-05-01

**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** htr, jsonlite, stringr,

**License** FreeBSD

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-05-09 07:16:06 UTC

## R topics documented:

add.dependency . . . . .	2
capture.src . . . . .	2
check.dependencies . . . . .	3
check.image.size . . . . .	3
is.https . . . . .	3
set.model.require . . . . .	4
yhat.batchDeploy . . . . .	4
yhat.deploy . . . . .	5
yhat.get . . . . .	6
yhat.library . . . . .	6
yhat.ls . . . . .	7

yhat.post . . . . .	7
yhat.predict . . . . .	8
yhat.predict_bulk . . . . .	8
yhat.predict_raw . . . . .	9
yhat.spider.block . . . . .	10
yhat.spider.func . . . . .	10
yhat.unload . . . . .	11
yhat.verify . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

add.dependency	<i>Private function that adds a package to the list of dependencies that will be installed on the ScienceOps server</i>
----------------	---

---

### 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	<i>Private function for capturing the source code of model</i>
-------------	--

---

### Description

Private function for capturing 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

<code>model_name</code>	name of your model
<code>packages</code>	list of packages to install using apt-get
<code>confirm</code>	boolean indicating whether to prompt before deploying
<code>custom_image</code>	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	<i>Private function for performing a GET request</i>
----------	--

---

**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	<i>Import one or more libraries and add them to the Yhat model's dependency list</i>
--------------	--

---

**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	<i>Private function for determining model dependencies</i>
---------	--

---

**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	<i>Private function for performing a POST request</i>
-----------	---

---

**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	<i>Make a prediction using Yhat.</i>
--------------	--------------------------------------

---

**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	<i>Make bulk predictions using Yhat.</i>
-------------------	--

---

**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	<i>Calls Yhat's REST API and returns a JSON document containing both the prediction and associated metadata.</i>
------------------	--

---

**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	<i>Removes a library from the Yhat model's dependency list</i>
-------------	--

---

**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	<i>Private function for verifying username and apikey</i>
-------------	---

---

**Description**

Private function for verifying username and apikey

**Usage**

```
yhat.verify()
```

# Index

- \*Topic **bulk**
  - yhat.predict\_bulk, 8
- \*Topic **deploy**
  - yhat.batchDeploy, 4
  - yhat.deploy, 5
- \*Topic **import**
  - yhat.library, 6
- \*Topic **predict**
  - yhat.predict, 8
  
- add.dependency, 2
  
- capture.src, 2
- check.dependencies, 3
- check.image.size, 3
  
- is.https, 3
  
- set.model.require, 4
  
- yhat.batchDeploy, 4
- yhat.deploy, 5
- yhat.get, 6
- yhat.library, 6
- yhat.ls, 7
- yhat.post, 7
- yhat.predict, 5, 8
- yhat.predict\_bulk, 8
- yhat.predict\_raw, 9
- yhat.spider.block, 10
- yhat.spider.func, 10
- yhat.unload, 11
- yhat.verify, 11