Package 'yhatr'

May 5, 2013

Type Package

Title R binder for the Yhat API

Version 0.1

Date 2013-05-03

Author Greg Lamp

Maintainer Greg Lamp <greg@yhathq.com>

Description yhatr let's you deploy, maintain, and invoke models via the Yhat REST API.

Depends R (>= 2.12.0)

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

Imports httr, rjson, plyr

License FreeBSD

Collate 'yhatr.R'

R topics documented:

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)
```

Arguments

model_name	name of your model

Examples

```
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"))
}
yhat.login("rtest", "abcd1234")
yhat.deploy("irisModel")
```

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

Description

Private function for performing a GET request

Usage

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

yhat.login — *A function for logging into Yhat's api.*

Description

A function for logging into Yhat's api.

Usage

```
yhat.login(username, apikey)
```

Arguments

username	Your Yhat username
apikey	Your Yhat apikey

Examples

```
yhat.login("hmardukas", "abcd1234")
```

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

Description

Private function for performing a POST request

Usage

```
yhat.post(endpoint, query = c(), data)
```

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, version, data)
```

Arguments

model_name	the name of the model you want to call
version	the version number of the model you want to call
data	input data for the model

Examples

```
yhat.predict("irisModel", 1, iris)
```

`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, version, data)
```

Arguments

`model_name`	the name of the model you want to call
`version`	the version number of the model you want to call
`data`	input data for the model

Examples

```
yhat.predict_raw("irisModel", 1, iris)
```

`yhat.scaffolding` *Quick function for setting up a basic scaffolding of functions for deploying on Yhat.*

Description

Quick function for setting up a basic scaffolding of functions for deploying on Yhat.

Usage

```
yhat.scaffolding()
```

Examples

```
yhat.scaffolding()
```

yhat.show_models *Shows which models you have deployed on Yhat.*

Description

This function queries the Yhat API and finds the models that have been deployed for your account.

Usage

```
yhat.show_models()
```

Examples

```
yhat.show_models()
# some output here
#    username className                name version
# 1      greg MySMSClassifier       1
# 2      greg MySMSClassifier       2
# 3      greg MySMSClassifier       3
# 4      greg MySMSClassifier       4
```

yhatR *A package for deploying statistical models on Yhat*

Description

yhatR requires you to implement 2 functions `model.transform` and `model.predict` and optionally `model.require`. Each function should perform the actions neccessary to generate predictions for your model. You can put anything you want in each function, so long as `model.predict` returns a `data.frame`.

Details

Package:	yhatR
Type:	Package
Version:	0.1
Date:	2013-05-03
License:	FreeBSD

See http://www.yhathq.com/ for more detials.

Author(s)

Greg Lamp <greg@yhathq.com>

References

http://www.yhathq.com/

See Also

http://www.yhathq.com/docs/1 http://www.yhathq.com/examples/

Examples

```
# build a quick model
iris$Sepal.Width_sq <- iris$Sepal.Width^2
fit <- glm(I(Species)=="virginica" ~ ., data=iris)


model.require <- function() {
  require("someLibrary")
}

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"))
}
yhat.deploy("irisModel")
```

Index