Introducing Rblpapi

Dirk Eddelbuettel

2015-08-13

Introduction

The Rblpapi package connects the R language and environment for statistical computing to the Bloomberg services supported by the official Bloomberg APIs.

The Rblpapi package is provided in both source and binary (for Windows and OS X) via the CRAN network for R. Source code is available at the corresponding GitHub repo; the blp repo contains the Bloomberg API components required during the build.

Usage

The next few section illustrate key functions within the package. All functions also have proper help pages for fuller details.

Package Load

Use

```
library(Rblpapi)
```

to load the package. You can also set options which will automatically connect at package load; see the next section.

Connecting

Use

```
blpConnect()
```

which will connect to the Bloomberg backend. Default values for the IP address (127.0.0.1) and port (8194) are used and can be overridden both as function arguments and via global options `blpHost` and `blpPort`. Moreover, if option `blpAutoConnect` is set to `TRUE`, this connection is done at package load.

The `appName` argument is optional and is for Application-Name authentication (via B-PIPE or SAPI, see `blpAuthenticate()`).

The resulting connection object is stored in a per-package environment providing a default value. If needed it can be overridden in each accessor function.

As the connection object is being held onto, the `blpDisconnect()` function is implemented as an empty function without side-effects. The internal function `defaultConnection()` returns the default connection object.

Default arguments and auto-connect can be set via `options()`

```
options("blpAutoConnect" = TRUE)
options("blpHost" = "x.x.x.x")
options("blpPort" = 8194)
options("blpAppName" = "yyy")
```

Optionally use

```
blpAuthenticate()
```

`blpAuthenticate()` is usually needed for SAPI & B-PIPE sessions. DAPI (Desktop) sessions do not usually require calling `blpAuthenticate()`, and will function with a default NULL `identity` object.

`blpAuthenticate()` connects to SAPI/B-PIPE server, and authenticates via UUID/login-location or Application-Name. It can set a default identity object for future calls (`bdp()`, `bds()`, etc). Or it can return an identity object to explicitly pass to future calls.

Additionally, if `blpAutoAuthenticate` is true, `blpAuthenticate()` will be called at package load, storing a default identity object. The internal function `defaultAuthentication()` returns the default identity object.

If needed, the default identity object can be overridden in each accessor function.

Default arguments and auto-authentication can be set via `options()`

```
options("blpAutoAuthenticate" = TRUE)
options("blpUUID" = "xxx")
options("blpLoginHostname" = "yyy")
options("blpLoginIP" = "z.z.z.z")
```

For application-id authentication, first use `blpConnect()` with an `appName` argument. Then call `blpAuthenticate()` with no arguments.

```
blpConnect( ... , appName = "appName")
blpAuthenticate()
```

For UUID authentication, connect with `blpConnect()`, then `blpAuthenticate()` with the UUID and last hostname/IP the UUID logged in from. Usually this is the IP of your Bloomberg Terminal. For your UUID, `IAM <GO>` in Bloomberg Terminal. Note, if you supply both IP and host arguments, the IP will be used, and the host ignored.

```
blpConnect( ... )
blpAuthenticate( uuid="UUID", ip.address="x.x.x.x")
```

bdp: Bloomberg Data Point

Current (or most recent) values of market-related instruments can be accessed via the `bdp()` function:

```
bdp(c("ESA Index", "SPY US Equity"), c("PX_LAST", "VOLUME"))
```

bds: Bloomberg Data Set

The `bds()` function can retrieve data of a more static nature:

```
bds("GOOG US Equity", "TOP_20_HOLDERS_PUBLIC_FILINGS")
```

bdh: Bloomberg Data History

Historical data (at a daily granularity) can be retrieved with `bdh()`:

```
bdh("SPY US Equity", c("PX_LAST", "VOLUME"), start.date=Sys.Date()-31)
```

getBars: OHLCV Aggregates

The `getBars()` function retrieves aggregated Open / High / Low / Close / Volume data. For example, in

```
getBars("ES1 Index")
```

the default values of six hourly bars for the lead ES future.

getTicks: Transactional Tick Data

The `getTicks()` function retrieves tickdata, albeit with timestamps at a minute granularity. For example, the call

```r
getTicks("ES1 Index")
```

retrieves all ticks for the given security over the last hour.

fieldSearch: Query For Fields

The `fieldSearch()` helper function can search for available fields which can be used with the `bdp()`, `bdh()` or `bds()` functions. For example,

```r
res <- fieldSearch("vwap")
```

search for fields describing volume-weighted average price fields

beqs: Bloomberg EQS Queries

The `beqs()` function (which was contribued by Rademeyer Vermaak) can access EQS functionality:

```r
beqs("Global Oil Companies YTD Return","GLOBAL")
```

Acknowledgements

Most valuable contributions from both Jonathan Owen (who kick-started builds on Windows when we considered these to be impossible) and Jeroen Ooms (who took care of builds on OS X) are gratefully acknowledged.

Legal

All trademarks and registered trademarks are the property of their respective owners.

All code of the Rblpapi package (ie directories `src/`, `R/`, ...) is released under the GNU GPL-3.

All code retrieved from the blp repository during build is released by Bloomberg, available at the Bloomberg API site and released under the license included below.

```
Copyright 2012. Bloomberg Finance L.P.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this proprietary software and associated documentation files (the "Software"),
to use, publish, or distribute copies of the Software, and to permit persons to
whom the Software is furnished to do so.

Any other use, including modifying, adapting, reverse engineering, decompiling,
or disassembling, is not permitted.

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
```