

Package ‘protolite’

September 1, 2017

Type Package

Title Fast and Simple Object Serialization to Protocol Buffers

Author Jeroen Ooms

Maintainer Jeroen Ooms <jeroen@berkeley.edu>

Description Optimized C++ implementations for reading and writing protocol-buffers. Currently supports 'rexp.proto' for serializing R objects and 'geobuf.proto' for geojson data. This lightweight package is complementary to the much larger 'RProtoBuf' package which provides a full featured toolkit for working with protocol-buffers in R.

Version 1.7

License MIT + file LICENSE

URL <https://github.com/jeroen/protolite#readme>

BugReports <https://github.com/jeroen/protolite/issues>

SystemRequirements libprotobuf and protobuf-compiler

LinkingTo Rcpp

Imports Rcpp (>= 0.12.12), jsonlite

Suggests testthat, RProtoBuf

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-09-01 11:14:17

R topics documented:

geobuf	2
serialize_pb	2

Index	4
--------------	----------

`geobuf`*Geobuf*

Description

The `geobuf` format is an optimized binary format for storing geojson data with protocol buffers. These functions are compatible with the `geobuf2json` and `json2geobuf` utilities from the `geobuf` [npm package](#).

Usage

```
read_geobuf(x, as_data_frame = TRUE)
```

```
geobuf2json(x, pretty = FALSE)
```

```
json2geobuf(json, decimals = 6)
```

Arguments

<code>x</code>	file path or raw vector with the serialized <code>geobuf.proto</code> message
<code>as_data_frame</code>	simplify geojson data into data frames
<code>pretty</code>	indent json, see jsonlite::toJSON
<code>json</code>	a text string with geojson data
<code>decimals</code>	how many decimals (digits behind the dot) to store for numbers

`serialize_pb`*Serialize to Protocol Buffers*

Description

Serializes R objects to a general purpose protobuf message. It uses the same `rexp.proto` descriptor and mapping between R objects and protobuf messages as `RHIPE` and the `RProtoBuf` package.

Usage

```
serialize_pb(object, connection = NULL, skip_native = FALSE)
```

```
unserialize_pb(msg)
```

Arguments

object	an R object to serialize
connection	a connection, file, or NULL for a raw vector
skip_native	do not serialize 'native' (non-data) R objects. Setting to TRUE will only serialize <i>data</i> types (numeric, boolean, string, raw, list). The default behavior is to fall back on base R serialize for non-data objects.
msg	raw vector with the serialized rexp.proto message

Details

The `serialize_pb` and `unserialize_pb` reimplement the identically named functions from the RProtoBuf package in pure C++. This makes the function faster and simpler, but the output should be identical.

Examples

```
# Serialize and unserialize an object
buf <- serialize_pb(iris)
out <- unserialize_pb(buf)
stopifnot(identical(iris, out))

## Not run: #Fully compatible with RProtoBuf
buf <- RProtoBuf::serialize_pb(iris, NULL)
out <- protolite::unserialize_pb(buf)
stopifnot(identical(iris, out))

# Other way around
buf <- protolite::serialize_pb(mtcars, NULL)
out <- RProtoBuf::unserialize_pb(buf)
stopifnot(identical(mtcars, out))

## End(Not run)
```

Index

`geobuf`, [2](#)
`geobuf2json` (`geobuf`), [2](#)

`json2geobuf` (`geobuf`), [2](#)
`jsonlite::toJSON`, [2](#)

`protolite` (`serialize_pb`), [2](#)

`read_geobuf` (`geobuf`), [2](#)
`RProtoBuf`, [2](#)

`serialize`, [3](#)
`serialize_pb`, [2](#)

`unserialize_pb` (`serialize_pb`), [2](#)