

Package ‘trackr’

November 7, 2018

Type Package

Title Semantic Annotation and Discoverability System for R-Based Artifacts

Version 0.10.5

Date 2018-11-05

Copyright Genentech, Inc.

Maintainer Gabriel Becker <gabembecker@gmail.com>

Description Automatically annotates R-based artifacts with relevant descriptive and provenance-related and provides a backend-agnostic storage and discoverability system for organizing, retrieving, and interrogating such artifacts.

License Artistic-2.0

Depends histry (>= 0.2.1)

Imports fastdigest, ggplot2, lattice, gridGraphics, methods, CodeDepends (>= 0.6.2), RJSONIO, rsolr, htmltools, miniUI, shiny, rmarkdown, roprov (>= 0.1.1), rlang

Suggests proto, png, httr, knitr, MASS, MEMSS, mlmRev, dplyr, gridExtra, rstudioapi, switchr (>= 0.12.99)

VignetteBuilder knitr

Collate 'Classes.R' 'DocCollectionRef-methods.R' 'Generics.R' 'PlotWrapper.R' 'ObjFeatureSet.R' 'Trackr-main-API.R' 'abbrev.R' 'accessor-methods.R' 'autotrack.R' 'backend-impls.R' 'coercerender.R' 'convenience_constructors.R' 'ggplotversions.R' 'knitr.R' 'makeRStudioExtras.R' 'parseCode.R' 'recordFile.R' 'shiny.R' 'showMethods.R'

RoxygenNote 6.0.1

NeedsCompilation no

Author Gabriel Becker [aut, cre],
Sara Moore [aut]

Repository CRAN

Date/Publication 2018-11-07 00:10:03 UTC

R topics documented:

annotationText	3
as.list	4
codeInfo<-	4
coordSystem	5
dataLabels<-	6
dataNames	7
dataTypes	8
defaultTDB	9
describePackage	9
DocCollectionRef	10
FeatureSet-class	10
fullData	12
generateTags	13
geomObject<-	13
ggplot-class	14
graphSys	15
groupInfo	15
hasLegend	16
JSONBackend	17
jsonTDB	18
knit_and_record	19
makeFeatureSet	20
make_image_files	22
manifestFromRecord	22
ndoc,TrackrDB-method	23
nLayers<-	23
nObs	24
objCode<-	25
ObjFeatureSet	26
parseCode	29
plot,PlotFeatureSet,missing-method	29
position<-	30
print.gTree	31
record	31
regDateTime<-	33
RStudioExtras	34
saveBasicPlot	34
statTransform<-	35
summary,PlotFeatureSet-method	36
tags<-	37
titles<-	38
trackrAddin	39
TrackrDB-class	40
TrackrOptions-class	40
trackropts	41
trackr_knit_env	41

annotationText 3

trackr_lookup	42
trackr_options	43
uniqueID<-	44
user<-	44

Index 46

<code>annotationText</code>	<i>Accessor method for the annotation text (slot "annotationtext") of the plot object.</i>
-----------------------------	--

Description

Get the text annotations for a plot object.

Usage

```
annotationText(object)

## S4 method for signature 'PlotFeatureSet'
annotationText(object)

## S4 method for signature 'ggplot'
annotationText(object)

## S4 method for signature 'trellis'
annotationText(object)

## S4 method for signature 'gTree'
annotationText(object)

## S4 method for signature 'PlotFeatureSet'
annotationText(object)
```

Arguments

`object` An object of class `PlotFeatureSet`, `GGplotFeatureSet`, `TrellisFeatureSet`, `ggplot`, or `trellis`.

Value

A character vector of text annotations appearing on the plot.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
annotationText(fs)
```

<code>as.list</code>	<i>Convert objects to lists</i>
----------------------	---------------------------------

Description

Convert objects to lists.

Usage

```
## S3 method for class 'FeatureSet'
as.list(x, ...)

## S3 method for class 'ScriptNodeInfo'
as.list(x, ...)
```

Arguments

<code>x</code>	The object to convert to a list.
<code>...</code>	Other named arguments (currently unused).

Value

A list.

<code>codeInfo<-</code>	<i>Accessors for the code analysis information of FeatureSet objects</i>
----------------------------	--

Description

Get or set the analysis information for the code associated with a FeatureSet object. This should not be called directly.

Usage

```
codeInfo(object) <- value

codeInfo(object)

## S4 replacement method for signature 'FeatureSet'
codeInfo(object) <- value

## S4 method for signature 'FeatureSet'
codeInfo(object)
```

Arguments

object	An object of (super)class PlotFeatureSet. For accession, also plot objects of class ggplot or trellis.
value	A CodeDepends::ScriptInfo object.

Value

A modified object of (super)class PlotFeatureSet.

Examples

```
mt = datasets::mtcars
fs = makeFeatureSet(mt)
codeInfo(fs)
```

coordSystem	<i>Accessor method for the coordinate system (slot "coordsys") of the plot object.</i>
-------------	--

Description

Get the coordinate system for a plot or PlotFeatureSet.

Usage

```
coordSystem(object)

## S4 method for signature 'PlotFeatureSet'
coordSystem(object)

## S4 method for signature 'ggplot'
coordSystem(object)

## S4 method for signature 'trellis'
coordSystem(object)

## S4 method for signature 'gTree'
coordSystem(object)

## S4 method for signature 'PlotFeatureSet'
coordSystem(object)
```

Arguments

object	An object of class PlotFeatureSet, GGplotFeatureSet, TrellisFeatureSet, ggplot, or trellis.
--------	---

Value

A character vector describing the coordinate system employed in the plot.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
coordSystem(fs)
```

dataLabels<- *Accessors for the axis labels PlotFeatureSet objects*

Description

Get or set the axis label information associated with a PlotFeatureSet object. This should not be called directly.

Usage

```
dataLabels(object) <- value

dataLabels(object)

## S4 method for signature 'PlotFeatureSet'
dataLabels(object)

## S4 method for signature 'ggplot'
dataLabels(object)

## S4 method for signature 'trellis'
dataLabels(object)

## S4 method for signature 'gTree'
dataLabels(object)

## S4 replacement method for signature 'PlotFeatureSet'
dataLabels(object) <- value

## S4 method for signature 'PlotFeatureSet'
dataLabels(object)
```

Arguments

object An object of (super)class PlotFeatureSet. For accession, also plot objects of class ggplot or trellis.

value A named list of variable labels of the form `list(x = "X axis label", y = "Y axis label", ...)`.

Value

A modified object of (super)class `PlotFeatureSet`.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mtcars$mpg, mtcars$cyl)
fs = makeFeatureSet(plt)
dataLabels(fs)
```

dataNames	<i>Accessor method for the variable names of the plot object. Not an exported method.</i>
-----------	---

Description

Get the variable names from a `PlotFeatureSet`

Usage

```
dataNames(object)
```

Arguments

object An object of (super)class `PlotFeatureSet`. For accession, also plot objects of class `ggplot` or `trellis`.

Value

A named list of variable names of the form `list(x = "x.name", y = "y.name", groups = list(...))`.

dataTypes	<i>Accessor method for the variable types (slot "vartypes") of the plot object.</i>
-----------	---

Description

Get the variable types from a plot feature set

Usage

```
dataTypes(object)

## S4 method for signature 'PlotFeatureSet'
dataTypes(object)

## S4 method for signature 'ggplot'
dataTypes(object)

## S4 method for signature 'trellis'
dataTypes(object)

## S4 method for signature 'gTree'
dataTypes(object)

## S4 method for signature 'PlotFeatureSet'
dataTypes(object)
```

Arguments

object An object of class PlotFeatureSet, GGplotFeatureSet, TrellisFeatureSet, ggplot, or trellis.

Value

A named list of variable types of the form list(x = "numeric", y = "factor", ...).

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
dataTypes(fs)
```

defaultTDB	<i>default TrackrDB</i>
------------	-------------------------

Description

Get or set the default TrackrDB in use (ie the one that is used when 'record' is used with no trackrdb specified).

Usage

```
defaultTDB(trackrdb)
```

Arguments

trackrdb	TrackrDB. The TrackrDB to which objects will be recorded by default. If missing, the current default is returned.
----------	---

describePackage	<i>Accessor method for ObjFeatureSet "analysispkg" slot.</i>
-----------------	--

Description

The analysis package, if any, that was 'in use' (i.e., that contained the working directory that was active) when the result was recorded.

Usage

```
describePackage(object)
```

```
## S4 method for signature 'FeatureSet'
describePackage(object)
```

Arguments

object	An object of (super)class ObjFeatureSet.
--------	--

Value

A list holding the name, title, and description of any packages with description files on the current path.

Examples

```
mt = datasets::mtcars
fs = makeFeatureSet(mt)
describePackage(fs)
```

DocCollectionRef	<i>Reference to a DocCollection</i>
------------------	-------------------------------------

Description

A reference class which carries around a DocCollection, suitable for use as a trackr backend.

Usage

```
ListBackend(lst = list())
```

Arguments

lst List of documents to populate the list backend with.

FeatureSet-class	<i>FeatureSet (and Sub)-Classes</i>
------------------	-------------------------------------

Description

Metadata inferred about R objects or dynamic documents is stored in FeatureSet objects specific. Specific types of featuresets have specific additional metadata they contain, beyond the standard metadata inferred about all results, and represented by the core FeatureSet class.

Slots

user character The user who submitted the result

regdate POSIXct The date/time the result was recorded

analysispkg list The R package associated with the result (because the working directory was within the package's directory structure).

uniqueid character The uniqueid of the result

tags character Additional tags associated with the result

analysisfile character The .R file active when there result was recorded (RStudio IDE only)

rstudioproject character The RStudio project active when the result was recorded (RStudio IDE only)

generatedin character The uniqueid of the Rmd file the result was generated in, if applicable

code character The code used to generate the result (by default, as captured by histry)

codeinfo ScriptInfo the ScriptInfo for the code

sessioninfo sinfoOrList The session info at the time the result was recorded

isplot logical Whether the result is a plot

fsetklass character The FeatureSet subclass for the result

`trackrversion` character The exact version of the trackr package used to record the result.

`clineargs` character The commandline arguments passed to R when starting the session the result was recorded from

`resultURI` character The URI associated with the result, see `featureset` constructor documentation.

`extrametadata` list Any extra metadata associated with the result.

`titles` Title and subtitle of the plot object; a named list of the form `list(main = "My title", sub = "My subtitle")`.

`klass` character The R object class of the result

`object` ANY The object itself, or NULL if the object is not available.

`vars` character The variable names for a `data.frame` result

`varclasses` character the variable classes for a `data.frame` result

`varsummaries` list summaries for

`object` ANY The object itself, or NULL if the object is not available.

`data` A list of `data.frames` containing the variables and observations used in the plot.

`varlabels` Variable labels of the plot object; a named list of the form `list(x = "X axis label", y = "Y axis label", groups = list(...))`. Note that non-empty labels are character vectors and may contain more than one entry.

`annotation.text` Annotation text of the plot object.

`vartypes` Variable types of the plot object; a named list of the form `list(x = "numeric", y = "factor", ...)`.

`grouping` Grouping information on the plot object; a named list.

`coordsys` A character vector describing the coordinate system employed in the plot.

`nobs` An integer representing the number of observations in the plotted data.

`haslegend` A boolean indicating whether or not a legend is displayed in the plot.

`tags` A character vector of user-defined tags.

`code` R code to reproduce the plot, as a `CodeDepends::Script` object. May be empty.

`codeinfo` Information about the R code to reproduce the plot, as a `CodeDepends::ScriptInfo` object. May be empty.

`geom` A named list of parameters for geometric objects in each layer of the `ggplot`.

`stat` A named list of parameters for statistical transforms in each layer of the `ggplot`.

`position` A named list of positioning information in each layer of the `ggplot`.

`num.layers` An integer representing the number of layers in the plot.

fullData	<i>Accessor method for the data (slot "data") that is used in plotting.</i>
----------	---

Description

Get full data associated with a plot or model fit, if possible.

Usage

```
fullData(object, quiet = FALSE)

## S4 method for signature 'PlotFeatureSet'
fullData(object)

## S4 method for signature 'ggplot'
fullData(object, quiet = FALSE)

## S4 method for signature 'trellis'
fullData(object, quiet = FALSE)

## S4 method for signature 'gTree'
fullData(object)

## S4 method for signature 'PlotFeatureSet'
fullData(object)
```

Arguments

object	An object of (super)class PlotFeatureSet. For accession, also plot objects of class ggplot or trellis.
quiet	Should warnings about plotted data be suppressed?

Value

A list of data.frame containing the variables used in plotting.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
fullData(fs)
```

generateTags	<i>customizing-metadata</i>
--------------	-----------------------------

Description

The generateTags is called when extracting metadata from an object. It is one mechanism by which custom metadata can be defined on a class-by-class basis without the more heavy-weight solution of defining an entirely new FeatureSet subclass.

Usage

```
generateTags(object)
```

Arguments

object	The object to generate tags for
--------	---------------------------------

Details

The upsides of this mechanism is that it is easier to use and lighter weight than defining new FeatureSet classes and methods to generate them. The downsides are that the metadata are tags, rather than proper key-value pairs, as far as trackr is concerned. (A backend could be engineered such that it interpreted tags of the form 'key:value' as key-value pairs, but this won't occur without extra work, and thus those implied fields will not be queriable via the trackr api specifically. The values will be included in the metadata generally though, so non-field-specific queries will work.

Value

A character vector of tags to associate with object during the recording process

Examples

```
generateTags(mtcars) #character(0)
```

geomObject<-	<i>Accessors for the geom of GGplotFeatureSet objects</i>
--------------	---

Description

Get or set the geom associated with a GGplotFeatureSet object. This should not be called directly.

Usage

```

geomObject(object) <- value

geomObject(object)

## S4 method for signature 'PlotFeatureSet'
geomObject(object)

## S4 method for signature 'ggplot'
geomObject(object)

## S4 method for signature 'trellis'
geomObject(object)

## S4 method for signature 'gTree'
geomObject(object)

## S4 replacement method for signature 'GGplotFeatureSet'
geomObject(object) <- value

## S4 method for signature 'GGplotFeatureSet'
geomObject(object)

```

Arguments

object	An object of class GGplotFeatureSet. For accession, also plot objects of class ggplot.
value	A named list of parameters for geometric objects in each layer of the plot.

Value

A modified object of class GGplotFeatureSet.

Examples

```

library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
geomObject(fs)

```

ggplot-class

ggplot S4 classes

Description

formal S4 classes for S3 classes defined in other packages.

graphSys	<i>Accessor method for PlotFeatureSet "package" slot.</i>
----------	---

Description

Retrieve the graphics system from a PlotFeatureSet object

Usage

```
graphSys(object)

## S4 method for signature 'PlotFeatureSet'
graphSys(object)
```

Arguments

object An object of (super)class PlotFeatureSet.

Value

A character vector representing the originating R package of the plot object, i.e. base, ggplot, or lattice.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
graphSys(fs)
```

groupInfo	<i>Accessor method for the panel and other grouping information (slot "grouping") of the plot object.</i>
-----------	---

Description

Get the grouping info for a plot or PlotFeatureSet.

Usage

```
groupInfo(object)

## S4 method for signature 'PlotFeatureSet'
groupInfo(object)

## S4 method for signature 'ggplot'
groupInfo(object)

## S4 method for signature 'trellis'
groupInfo(object)

## S4 method for signature 'gTree'
groupInfo(object)

## S4 method for signature 'PlotFeatureSet'
groupInfo(object)
```

Arguments

`object` An object of class `PlotFeatureSet`, `GGplotFeatureSet`, `TrellisFeatureSet`, `ggplot`, or `trellis`.

Value

A named list of grouping information parameters.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
groupInfo(fs)
```

<code>hasLegend</code>	<i>Accessor method for the legend indicator (slot "haslegend") of the plot object.</i>
------------------------	--

Description

Get whether or not a plot or `PlotFeatureSet`.

Usage

```
hasLegend(object)

## S4 method for signature 'PlotFeatureSet'
hasLegend(object)

## S4 method for signature 'ggplot'
hasLegend(object)

## S4 method for signature 'trellis'
hasLegend(object)

## S4 method for signature 'gTree'
hasLegend(object)

## S4 method for signature 'PlotFeatureSet'
hasLegend(object)
```

Arguments

object An object of class PlotFeatureSet, GGplotFeatureSet, TrellisFeatureSet, ggplot, or trellis.

Value

A boolean indicating whether or not a legend is displayed in the plot.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
hasLegend(fs)
```

JSONBackend

JSON backend for trackr

Description

A JSON-file based backend. This is the default backend in trackr.

Usage

```
JSONBackend(file = normalizePath("../trackr_db_data.json"), data = list())
```

Arguments

file	character. The json "database" to use as a trackr backend
data	A list of records to pre-populate the backend with.

Value

A JSONBackend object, for use in creating a TrackrDB object.

Slots

data	list. An in-memory list representation of the data in the db
file	character. The file containing the db (to read from and write to)
last_load	POSIXct. The last time data was updated from disk.

Note

This is a reference class, which does NOT have standard copy-on-write semantics
 This function should generally not be called directly by end-users. See instead [jsonTDB](#)

 jsonTDB

Convenience constructors for specific trackr backends

Description

Convenience constructors for JSON and Solr-based trackr backends.

Usage

```
jsonTDB(file = "~/trackr/objdb.json", opts = TrackrOptions(img_dir =
  img_dir, ...), img_dir = file.path(dirname(file), "images"), ...)

solrTDB(core, requestHandler = "search", opts = TrackrOptions(...), ...)

listTDB(data = list(), opts = TrackrOptions(...), ...)
```

Arguments

file	character. The json file to use
opts	TrackrOptions. The options for the DB
img_dir	character. The directory plot images should be saved into. Passed to default construction of opts. Defaults to <directory of file>/images. ignored if opts is specified explicitly.
...	ANY. Used to construct opts. Ignored if opts is specified explicitly.
core	character. The URI for the solr core to use
requestHandler	character. Passed to SolrList constructor
data	list. Data the ListBackend should be prepopulated with.

Value

A TrackrDB object

Examples

```
tdb = jsonTDB(tempfile())
```

```
tdb2 = listTDB()
```

knit_and_record	<i>Knit and record an Rmd, Rnw, etc file</i>
-----------------	--

Description

This function wraps knitr's knit function in a way that captures and records some or all values generated by code within the report, as well as the report itself.

This means that many records will generally be added to the trackr db for a single call to this function.

Usage

```
knit_and_record(input, ..., verbose = FALSE, tmptdb = TrackrDB(backend =
  ListBackend(), img_dir = img_dir(defaultTDB())), recvars = NULL,
  dryrun = FALSE)
```

Arguments

input	The input argument exactly as knitr's knit function accepts it
...	Passed directly to knit
verbose	passed to (multiple) record calls for report and its outputs
tmptdb	A TrackrDB in which to temporarily record results which are printed within the dynamic document. Generally this should not need to be changed, as it is only used to collect the records so they can be associated with the result for the whole document (in the defaultTDB).
recvars	character or NULL. The names of variables generated by the code within input which should be automatically recorded, or NULL.
dryrun	logical. Should a dryrun be performed?

Details

When recvars is NULL, any objects which are visibly printed within the report are also recorded. Otherwise, only the values of the listed variables (after all code has been evaluated) are recorded, regardless of visibility

Note

as with all knitr support in the histry and trackr packages, manually tracing certain functions within the knitr and evaluate packages will break this function.

makeFeatureSet	<i>Construct an ObjFeatureSet.</i>
----------------	------------------------------------

Description

The generic function for transforming an R object into the appropriate subclass of FeatureSet. This includes most introspection-based metadata extraction from the object. Metadata extraction can be customized at the package/R session level for specific classes of objects by defining methods for this generic.

Usage

```
makeFeatureSet(object, ...)

## S4 method for signature 'ggplot'
makeFeatureSet(object, ...)

## S4 method for signature 'trellis'
makeFeatureSet(object, ...)

## S4 method for signature 'gTree'
makeFeatureSet(object, ...)

## S4 method for signature 'recordedplot'
makeFeatureSet(object, ...)

## S4 method for signature 'expression'
makeFeatureSet(object, ...)

## S4 method for signature 'call'
makeFeatureSet(object, ...)

## S4 method for signature 'FeatureSet'
makeFeatureSet(object, ...)

## S4 method for signature 'ANY'
makeFeatureSet(object, ...)

## S4 method for signature 'data.frame'
makeFeatureSet(object, ...)
```

Arguments

object The plot to summarize via metadata in an ObjFeatureSet (or subclass)
 ... Other named arguments that become slots in the new PlotFeatureSet.

Value

PlotFeatureSet S4 object containing a ObjFeatureSet object and extracted plot features

Methods (by class)

- `ggplot`: Construct a ObjFeatureSet from an object of class `ggplot`.
- `trellis`: Construct a ObjFeatureSet from an object of class `trellis`.
- `gTree`: Construct a ObjFeatureSet from an object of class `gTree`.
- `recordedplot`: Construct a ObjFeatureSet from an object of class `recordedplot`.
- `expression`: Construct a ObjFeatureSet from an unevaluated expression.
- `call`: Construct a ObjFeatureSet from a call.
- `FeatureSet`: No-op if we already have a ObjFeatureSet
- `ANY`: Catch-all for attempted construction of a ObjFeatureSet from an object not of class `ggplot` or `trellis`.
- `data.frame`: Construct a DFFeatureSet from a `data.frame`

Examples

```
library(ggplot2)
mt = datasets::mtcars
pg <- ggplot(mt, aes(wt, mpg)) + geom_point()
pfs <- makeFeatureSet(pg)

library(lattice)
titan <- datasets::Titanic
pl <- barchart(Class ~ Freq | Sex + Age, data = as.data.frame(titan),
              groups = Survived, stack = TRUE, layout = c(4, 1),
              auto.key = list(title = "Survived", columns = 2))
pfs <- makeFeatureSet(pl)

dffs <- makeFeatureSet(as.data.frame(titan))

## Not run:
plot(1:10, 1:10)
pb <- recordPlot()
pfs <- makeFeatureSet(pb)

## End(Not run)
```

make_image_files	<i>Make image files for a featureset</i>
------------------	--

Description

Make image files for a featureset

Usage

```
make_image_files(object, opts)

## S4 method for signature 'PlotFeatureSet'
make_image_files(object, opts)

## S4 method for signature 'RmdFeatureSet'
make_image_files(object, opts)

## S4 method for signature 'ObjFeatureSet'
make_image_files(object, opts)

## S4 method for signature 'FeatureSet'
make_image_files(object, opts)

## S4 method for signature 'ANY'
make_image_files(object, opts)
```

Arguments

object	A FeatureSet object
opts	Options

Value

A named list with two entries: `preview.path`, and `image.path`. These should be paths to (now) existing iamge fiels for thumbnail and main display, respectively

manifestFromRecord	<i>Generate a (switchr) Seed Manifest from a Record</i>
--------------------	---

Description

Extracts the session info information within the record and uses it to generate a switchr manifest which can be used to reinstall exact versions of the R packages associated with the result

Usage

```
manifestFromRecord(lst)
```

Arguments

lst The record in the form of a list (eg an element of the list returned by findRecords)

Value

A switchr SessionManifest object

ndoc, TrackrDB-method *Number of docs in a TrackrDB*

Description

Query the backend for the number of docs it contains

Usage

```
## S4 method for signature 'TrackrDB'
ndoc(x, ...)
```

Arguments

x TrackrDB.
... not used.

nLayers<- *Accessors for the number of layers of GGplotFeatureSet objects*

Description

Get or set the number of layers associated with a GGplotFeatureSet object. This should not be called directly.

Usage

```

nLayers(object) <- value

nLayers(object)

## S4 method for signature 'PlotFeatureSet'
nLayers(object)

## S4 method for signature 'ggplot'
nLayers(object)

## S4 method for signature 'trellis'
nLayers(object)

## S4 method for signature 'gTree'
nLayers(object)

## S4 replacement method for signature 'GGplotFeatureSet'
nLayers(object) <- value

## S4 method for signature 'GGplotFeatureSet'
nLayers(object)

```

Arguments

object	An object of class GGplotFeatureSet. For accession, also plot objects of class ggplot.
value	An integer representing the number of layers in the plot.

Value

A modified object of class GGplotFeatureSet.

Examples

```

library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
nLayers(fs)

```

nObs	<i>Accessor method for the number of observations (slot "nobs") of the plot object.</i>
------	---

Description

Number of observations plotted in a plot or associated PlotFeatureSet.

Usage

```
nObs(object)

## S4 method for signature 'PlotFeatureSet'
nObs(object)

## S4 method for signature 'trellis'
nObs(object)

## S4 method for signature 'gTree'
nObs(object)

## S4 method for signature 'PlotFeatureSet'
nObs(object)

## S4 method for signature 'ggplot'
nObs(object)
```

Arguments

object An object of class PlotFeatureSet, GGplotFeatureSet, TrellisFeatureSet, ggplot, or trellis.

Value

An integer representing the number of observations in the plotted data.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
nObs(fs)
```

objCode<-

Accessors for the code of FeatureSet objects

Description

Get or set the code associated with a FeatureSet object. This should not be called directly.

Usage

```
objCode(object) <- value

objCode(object)

## S4 replacement method for signature 'FeatureSet'
objCode(object) <- value

## S4 method for signature 'FeatureSet'
objCode(object)
```

Arguments

object An object of (super)class PlotFeatureSet. For accession, also plot objects of class ggplot or trellis.

value A CodeDepends::Script object.

Value

A modified object of (super)class PlotFeatureSet.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
objCode(fs)
```

ObjFeatureSet

FeatureSet constructors

Description

Constructors for different typed FeatureSet subclasses. These should only be used when customizing metadata extraction, i.e., in custom makeFeatureSet methods. Not intended to be called directly by end users. Because the arguments listed here unavoidably map to class slots, and because most of them should NOT be set directly even in customization code, they are semi-internal implementation details and are subject to change.

Arguments described as 'Do not manually set' have default values that should be used in virtually all cases. Overriding these in custom makeFeatureSet methods can lead to undefined behavior by the trackr system.

Usage

```
ObjFeatureSet(object, code = as.character(parseCode(object)),
  codeinfo = CodeDepends::getInputs(parseCode(code)),
  klass = getTopS3Class(object), uniqueid = gen_hash_id(object),
  tags = character(), user = unname(Sys.info()["user"]),
  regdate = Sys.time(), analysispkg = scrape_descr(),
  analysisfile = .analysisFileOrNA(), rstudioproject = .rstudioProjOrNA(),
  fsetklass = "ObjFeatureSet", isplot = FALSE, generatedin = character(),
  clineargs = commandArgs(), resultURI = character(),
  provtable = ProvStoreDF(), ...)

PlotFeatureSet(object, fsetklass = "PlotFeatureSet",
  package = NA_character_, ...)

GGplotFeatureSet(object, fsetklass = "GGplotFeatureSet", ...)

TrellisFeatureSet(object, fsetklass = "TrellisFeatureSet", ...)

GraphicsFeatureSet(object, fsetklass = "GraphicsFeatureSet", ...)

DFFeatureSet(object, fsetklass = "DFFeatureSet", vars = names(object),
  varclasses = .makeClassesTab(object),
  varsummaries = structure(lapply(object, .makeSummaryTable), names =
  names(object)), nobs = nrow(object), ...)

RmdFeatureSet(rmdfile, outputfile, uniqueid,
  rmdfileid = gen_hash_id(readLines(rmdfile)), chunks,
  numouts = length(trackr_backend(objtdb)),
  numplots = sum(sapply(objrecords, function(x) x$isplot)), titles = "",
  author = "", textkeywords = character(), codekeywords = character(),
  outputids = sapply(objrecords, function(x) x$uniqueid, USE.NAMES = FALSE),
  tags = character(), user = unname(Sys.info()["user"]),
  regdate = Sys.time(), analysispkg = scrape_descr(),
  analysisfile = .analysisFileOrNA(), rstudioproject = .rstudioProjOrNA(),
  fsetklass = "RmdFeatureSet", objrecords = findRecords(".", db = objtdb),
  objtdb, figurefiles = NA_character_, resultURI = "", ...)
```

Arguments

object	object to extract metadata from
code	The code which generated the object. Do not manually set
codeinfo	Do not manually set
klass	The class of the object. Do not manually set
uniqueid	The uniqueID for the result. Do not manually set. EVER.
tags	Tags to associate with the object
user	The user who recorded the object. Do not manually set

regdate	The registration date/time. Do not manually set
analysispkg	The analysis R package in which the session was run. Do not manually set
analysisfile	The .R file code was executed from to create the object. Do not manually set
rstudioproject	The RStudio project in which the object was created. Do not manually set
fsetclass	The FeatureSet subclass being created. This should be overridden with custom FeatureSet subclasses
isplot	Is the object a plot. Do not manually set.
generatedin	The uniqueID of a parent result (e.g. an RMD report the object was generated within). Do not manually set
clineargs	The command-line arguments passed to the R session in which the object was recorded. Do not manually set.
resultURI	An optional character value which defines a location within a hierarchical grouping for results tracked by trackr. E.g. '/groups/Becker/HousingData/analysis3'
provtable	A ProvStoreDF object containing "value lineage" provenance information
...	For ObjFeatureSet and RmdFeatureSet, unused. For Other constructors, passed to the parent constructor.
package	The plotting package used to create a plot.
vars	Do not manually set
varclasses	Do not manually set
varsummaries	Do not manually set
nobs	Do not manually set
rmdfile	The (input) RMD file
outputfile	the path to the woven report
rmdfileid	Id associated with the input .Rmd file. Do not manually set. EVER.
chunks	The code and text chunks of the dynamic document
numouts	Do not manually set
numplots	Do not manually set
titles	Do not manually set
author	Do not manually set
textkeywords	Keywords extracted from the text
codekeywords	Keywords extracted from the code
outputids	Do not manually set
objrecords	Do not manually set. EVER.
objtdb	The (temporary) trackerdb where individual displayed outputs were recorded during the weaving process.
figurefiles	image files of plot as figures for woven report. Do not manually set.

Value

An object of a class that inherits from FeatureSet

parseCode	<i>Parse result creation code</i>
-----------	-----------------------------------

Description

parse code into a CodeDepends::Script

Usage

```
parseCode(code)

## S4 method for signature 'expression'
parseCode(code)

## S4 method for signature 'call'
parseCode(code)

## S4 method for signature 'character'
parseCode(code)

## S4 method for signature 'ggplot'
parseCode(code)

## S4 method for signature 'trellis'
parseCode(code)

## S4 method for signature 'ANY'
parseCode(code)
```

Arguments

code	An expression, call or character vector, or a PlotFeatureSet or plot object. If missing, this method will attempt to extract code out of the plot object.
------	---

Value

A CodeDepends::Script object.

plot, PlotFeatureSet, missing-method	<i>Display the plot object owned by PlotFeatureSet objects and subclasses thereof.</i>
--------------------------------------	--

Description

(re)draw the plot associated with a PlotFeatureSEt record.

Usage

```
## S4 method for signature 'PlotFeatureSet,missing'
plot(x, y, ...)
```

Arguments

<code>x</code>	An object of (super)class <code>PlotFeatureSet</code> .
<code>y</code>	Should be left empty. Included only because it is a mandatory signature element.
<code>...</code>	Other named arguments (currently ignored).

Value

The plot object, plotted through its native print method.

<code>position<-</code>	<i>Accessors for the position of GGplotFeatureSet objects</i>
----------------------------	---

Description

Get or set the position associated with a `GGplotFeatureSet` object. This should not be called directly.

Usage

```
position(object) <- value

position(object)

## S4 method for signature 'PlotFeatureSet'
position(object)

## S4 method for signature 'ggplot'
position(object)

## S4 method for signature 'trellis'
position(object)

## S4 method for signature 'gTree'
position(object)

## S4 replacement method for signature 'GGplotFeatureSet'
position(object) <- value

## S4 method for signature 'GGplotFeatureSet'
position(object)
```

Arguments

object	An object of class GGplotFeatureSet. For accession, also plot objects of class ggplot.
value	A named list of positioning information in each layer of the plot.

Value

A modified object of class GGplotFeatureSet.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
position(fs)
```

print.gTree	<i>Print method for an object of class gTree.</i>
-------------	---

Description

print a gTree object.

Usage

```
## S3 method for class 'gTree'
print(x, ...)
```

Arguments

x	An object of class gTree.
...	Other named arguments (currently unused).

record	<i>Primary high-level API functions for Trackr Databases</i>
--------	--

Description

These functions define the high-level, user-facing trackr API.

Usage

```

record(object, db = defaultTDB(), resultURI = "",
       code = history::history_tracker(), force = FALSE, verbose = FALSE,
       symorpos = NULL, dryrun = FALSE)

rmRecord(object, db = defaultTDB(), verbose = FALSE)

findRecords(pattern, db = defaultTDB(), fields = NULL,
            ret_type = c("doclist", "id", "backend"), verbose = FALSE)

recordFiles(object, ingestfun = NULL, db = defaultTDB(), resultURI = "",
            code = history_tracker(), force = FALSE, verbose = FALSE,
            symorpos = NULL, dryrun = FALSE)

```

Arguments

object	ANY. An object to record. Or (for recordFiles) a vector of one or more files to record or the path of a single directory whose contents will be recorded. Or (for rmPlot) the unique ID of an object in the database to remove.
db	TrackrDB. The database
resultURI	A URI which can be used to define a grouping/hierarchy of results recorded via trackr. Currently unused by trackr itself except as additional metadata to search across. Defaults to an empty string.
code	ANY. Code/evaluation history to be associated with object
force	logical. Overwrite any existing entry matching object. (default: FALSE)
verbose	logical. Should extra informative messages be displayed (default: FALSE)
symorpos	The symbol or position corresponding to object in code. For normal usage this will not be required.
dryrun	logical. Should a "dry run" be performed. If true, the insert_record and trackr_write steps of the record process are <i>not</i> run, and the result of prep_for_backend is immediately returned.
pattern	character. A regular expression to match against the text in fields
fields	character or NULL. The fiends in which to match, or NULL to include all fields.
ret_type	character. Format in which to return the response. Options are: "id" - id of matching documents (default), "list" - A list containing the matching documents represnted as R lists, and "backend" - a backend specific representation of the set of matching documents
ingestfun	function or NULL. A function which must accept only the paths argument which will take object and return an R object to be recorded and linked to the raw files indicated by object

Details

These functions allow end-users to interact with trackr databases. Each function does what its name suggests.

When more than one file is passed to recordFiles, either via a vector of paths or the path to a directory, the files will be zipped up into a single file which will be recorded.

Examples

```
prevtdb = defaultTDB()
defaultTDB(listTDB())
mt = datasets::mtcars
record(mt)

res = findRecords("mtcars")
stopifnot(length(res) == 1)
rmRecord(mt)

res = findRecords("mtcars")
stopifnot(length(res) == 0)

## Not run:
f = function(paths) readLines(paths)
fil = system.file("test_docs", "knitr_test.Rmd", package = "trackr")
recordFiles(fil, ingestfun = f)

res= findRecords("test_docs")

## End(Not run)
defaultTDB(prevtdb)
```

regDateTime<-

Accessors for the registration Date/Time of FeatureSet objects

Description

Get or set the registration time associated with a FeatureSet object. This should not be called directly.

Usage

```
regDateTime(object) <- value

regDateTime(object)

## S4 replacement method for signature 'FeatureSet'
regDateTime(object) <- value

## S4 method for signature 'FeatureSet'
regDateTime(object)
```

Arguments

object An object of (super)class ObjFeatureSet.
 value A POSIXct timestamp representing the moment when the plot was registered.

Value

A modified object of (super)class ObjFeatureSet.

Examples

```
mt = datasets::mtcars
fs = makeFeatureSet(mt)
regDateTime(fs)
```

RStudioExtras

RStudioExtras class

Description

A class to contain information about the current session when working inside the RStudio IDE.

saveBasicPlot

Save the plot object owned by an object of class/superclass PlotFeatureSet as an image.

Description

Save a plot to an image file as part of the record process. This generally shouldn't need to be overwritten.

Usage

```
saveBasicPlot(object, filename, ...)
```

```
## S4 method for signature 'PlotFeatureSet'
saveBasicPlot(object, filename, type = c("png",
    "jpeg", "jpg", "tiff", "tif", "bmp"), width = 7, height = 7, dpi = 300)
```

Arguments

object	An object of (super)class PlotFeatureSet or plot object of class ggplot or trellis.
filename	The full filename, including path, where the plot should be saved.
...	Other named arguments passed on to the class-specific save method.
type	The type of image to save. Default is "png"; should be one of "jpeg", "jpg", "tiff", "tif", "png", or "bmp". In future versions, this should be automatically determined from the filename.
width	Width of output, in inches
height	Height of output, in inches
dpi	Resolution of output, in pixels per inch

Value

A boolean indicating success of the save operation.

Methods (by class)

- PlotFeatureSet: Save the plot object owned by an object of class/superclass PlotFeatureSet as an image.

statTransform<- *Accessors for the stat(s) of GGplotFeatureSet objects*

Description

Get or set the stat transform(s) associated with a GGplotFeatureSet object. This should not be called directly.

Usage

```
statTransform(object) <- value

statTransform(object)

## S4 method for signature 'PlotFeatureSet'
statTransform(object)

## S4 method for signature 'ggplot'
statTransform(object)

## S4 method for signature 'trellis'
statTransform(object)

## S4 method for signature 'gTree'
statTransform(object)
```

```
## S4 replacement method for signature 'GGplotFeatureSet'
statTransform(object) <- value

## S4 method for signature 'GGplotFeatureSet'
statTransform(object)
```

Arguments

object An object of class GGplotFeatureSet. For accession, also plot objects of class ggplot.

value A named list of parameters for statistical transforms in each layer of the plot.

Value

A modified object of class GGplotFeatureSet.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
statTransform(fs)
```

summary,PlotFeatureSet-method

Summarize an object of class/superclass PlotFeatureSet

Description

Summary methods for PlotFeatureSet objects

Usage

```
## S4 method for signature 'PlotFeatureSet'
summary(object)
```

Arguments

object An object of (super)class PlotFeatureSet.

`tags<-`*Accessors for the tags of FeatureSet objects*

Description

Get or set the tags associated with a FeatureSet object. This should not be called directly.

Usage

```
tags(object) <- value

editTags(object, value, option)

tags(object)

## S4 replacement method for signature 'FeatureSet'
tags(object) <- value

## S4 replacement method for signature 'ggplot'
tags(object) <- value

## S4 replacement method for signature 'trellis'
tags(object) <- value

## S4 replacement method for signature 'gTree'
tags(object) <- value

## S4 method for signature 'FeatureSet'
editTags(object, value, option = c("replace", "add",
  "remove"))

## S4 method for signature 'FeatureSet'
tags(object)

## S4 method for signature 'ggplot'
tags(object)

## S4 method for signature 'trellis'
tags(object)

## S4 method for signature 'gTree'
tags(object)

## S4 method for signature 'ANY'
tags(object)
```

Arguments

object	An object of (super)class FeatureSet
value	A character vector of tags.
option	One of c("replace", "add", "remove"), with default "replace" (equivalent to "tags<-" with an additional uniqueness check), describing the action to be taken with the tag vector provided.
...	Additional named arguments.

Value

A modified object of (super)class PlotFeatureSet.

A modified object of (super)class PlotFeatureSet.

Examples

```
library(ggplot2)
mt = datasets::mtcars
plt = qplot(mt$mpg, mt$cyl)
fs = makeFeatureSet(plt)
tags(fs)
```

titles<- *Accessors for the title(s) of FeatureSet objects*

Description

Get or set the titles associated with a FeatureSet object. This should not be called directly.

Usage

```
titles(object) <- value

titles(object)

## S4 method for signature 'PlotFeatureSet'
titles(object)

## S4 method for signature 'ggplot'
titles(object)

## S4 method for signature 'trellis'
titles(object)

## S4 method for signature 'gTree'
titles(object)
```

```
## S4 replacement method for signature 'PlotFeatureSet'  
titles(object) <- value  
  
## S4 method for signature 'FeatureSet'  
titles(object)
```

Arguments

object An object of (super)class PlotFeatureSet. For accession, also plot objects of class ggplot or trellis.

value A named list of titles of the form list(main = "My title", sub = "My subtitle").

Value

A modified object of (super)class PlotFeatureSet.

Examples

```
library(ggplot2)  
mt = datasets::mtcars  
plt = qplot(mtcars$mpg, mtcars$cyl)  
fs = makeFeatureSet(plt)  
titles(fs)
```

trackrAddin

RStudio addin/Shiny app for artifact discovery

Description

This function initiates a shiny server which allows users to search a trackr database and view the results. It can be used as a stand-alone Shiny application or from within the RStudio IDE as an addin

Usage

```
trackrAddin(tdb = defaultTDB())
```

Arguments

tdb The database to search. Defaults to the current default database

TrackrDB-class *Trackr database*

Description

A TrackrDB object is a combination of a backend and a TrackrOptions object controlling trackr's behavior.

Usage

```
TrackrDB(opts = TrackrOptions(...), backend = JSONBackend(), ...)
```

Arguments

opts	TrackrOptions object.
backend	ANY. The backend to use.
...	ignored.

TrackrOptions-class *Trackr configuration options*

Description

TrackrOptions objects dictate various behavior by trackr when interacting with or preparing to interact with backends. See individual parameters for behaviors controlled in this manner.

Usage

```
TrackrOptions(insert_delay = 0, img_dir = "./images", img_ext = "png",
  backend_opts = list(...), ...)
```

Arguments

insert_delay	numeric. delay in seconds between insertions.
img_dir	character. Directory to save image files in. This will be normalized via normalizePath
img_ext	character. extension to give image files.
backend_opts	list. list of options specific to the backend. Currently ignored by trackr machinery.
...	additional arguments, which are collected into a list for the default backend_opts value.

trackropts	<i>trackr options environment</i>
------------	-----------------------------------

Description

An environment where the default trackrdb and history tracker are stored. Users should not modify the contents of this environment directly.

Usage

trackropts

Format

An object of class environment of length 3.

trackr_knit_env	<i>Internal details</i>
-----------------	-------------------------

Description

This environment should never be manipulated or interacted with directly by end users. It is exported only due the vagaries of how parts of trackr are implemented.

Usage

trackr_knit_env

Format

An object of class environment of length 0.

 trackr_lookup

Backend-interface API

Description

This page describes the API which must be implemented by all Trackr backends. Any class with methods for these generic can be used in the backend slot of a TrackrDB object. These functions should not be called directly by an end-user.

Usage

```

trackr_lookup(object, target, opts, exist = FALSE)

insert_record(object, id, target, opts, verbose = FALSE)

prep_for_backend(object, target, opts, verbose = FALSE)

remove_record(object, target, opts, verbose = FALSE)

trackr_write(target, opts, verbose = FALSE)

trackr_search(pattern, target, opts, fields = NULL, ret_type = c("doclist",
  "id", "backend"), verbose = TRUE)

```

Arguments

object	ANY. The object to lookup, add, remove, etc.
target	ANY. The backend of the TrackrDB instance.
opts	TrackrOptions. The trackr-level options. Typically extracted from target in a TrackrDB method and passed down.
exist	logical. Return TRUE/FALSE rather than the looked-up object. (default: FALSE)
id	character. The id to assign to the recorded result within the target backend.
verbose	logical. Should extra informative messages be displayed (default: FALSE)
pattern	character. A regular expression to match against the text in fields
fields	character or NULL. The fiends in which to match, or NULL to include all fields.
ret_type	character. Format in which to return the response. Options are: "id" - id of matching documents (default), "doclist" - A list containing the matching documents represented as R lists, and "backend" - a backend specific representation of the set of matching documents (generally the same class as backend).

Details

Methods should be written to be dispatched on `target`, and, where appropriate on `codeobject/codedoc`. These methods should be endomorphic with respect to the `target` argument. That is, they should return an object of the same class as was passed to `target`, which reflect the requested change to the backend state.

`insert_record` may or may not involve writing to disk, which can alternatively occur during `trackr_write`. Writing, if any is desired, must occur within at one and only one of these methods. If `insert_record` performs the writing, `trackr_write` should be a no-op.

Value

for `trackr_lookup`: If `exist` is `TRUE`, a logical indicating whether object was found in db. Otherwise, the object stored in the database (or `NULL` if it was not found).

for `insert_record` and `remove_record`: The `TrackrDB` (`db` parameter) after the plot has been added or removed.

For `prep_for_backend`, `object`, representend in the form that the `insert_record` method for backend expects.

Note

`remove_record` should have the same writing behavior as `insert_record`

`remove_record` should have the same writing behavior as `insert_record`

<code>trackr_options</code>	<i>options extraction</i>
-----------------------------	---------------------------

Description

These functions extract the trackr options associated with a backend.

Usage

```
trackr_options(db)
```

```
img_dir(db)
```

```
img_ext(db)
```

Arguments

`db` Object to extract information from

uniqueID<- *Accessor methods for "uniqueid" of FeatureSet objects*

Description

Get or set the uniqueID field of a FeatureSet. This should not be called directly by end users.

Usage

```
uniqueID(object) <- value

uniqueID(object)

## S4 method for signature 'FeatureSet'
uniqueID(object)

## S4 replacement method for signature 'FeatureSet'
uniqueID(object) <- value
```

Arguments

object	An object of (super)class ObjFeatureSet.
value	A character vector serving as the unique ID of the plot object in the database.

Value

A modified object of (super)class ObjFeatureSet.

user<- *Accessors for the "user" of FeatureSet objects*

Description

Get or set the user associated with a FeatureSet object. This should not be called directly.

Usage

```
user(object) <- value

user(object)

## S4 replacement method for signature 'FeatureSet'
user(object) <- value

## S4 method for signature 'FeatureSet'
user(object)
```

Arguments

<code>object</code>	An object of (super)class <code>ObjFeatureSet</code> .
<code>value</code>	A character vector serving as the username associated with the plot object in the database.

Value

A modified object of (super)class `ObjFeatureSet`.

Examples

```
mt = datasets::mtcars
fs = makeFeatureSet(mt)
user(fs)
```

Index

*Topic **datasets**

- trackr_knit_env, 41
- trackropts, 41
- .DFFeatureSet (FeatureSet-class), 10
- .FeatureSet (FeatureSet-class), 10
- .GGplotFeatureSet (FeatureSet-class), 10
- .GraphicsFeatureSet (FeatureSet-class), 10
- .ObjFeatureSet (FeatureSet-class), 10
- .PlotFeatureSet (FeatureSet-class), 10
- .RmdFeatureSet (FeatureSet-class), 10
- .TrellisFeatureSet (FeatureSet-class), 10
- .docrefclass (DocCollectionRef), 10
- annotationText, 3
- annotationText, ggplot-method (annotationText), 3
- annotationText, gTree-method (annotationText), 3
- annotationText, PlotFeatureSet-method (annotationText), 3
- annotationText, trellis-method (annotationText), 3
- as.list, 4
- as.list, FeatureSet-method (as.list), 4
- as.list.FeatureSet (as.list), 4
- as.list.ScriptNodeInfo (as.list), 4
- codeInfo (codeInfo<-), 4
- codeInfo, FeatureSet-method (codeInfo<-), 4
- codeInfo<-, 4
- codeInfo<-, FeatureSet-method (codeInfo<-), 4
- coordSystem, 5
- coordSystem, ggplot-method (coordSystem), 5
- coordSystem, gTree-method (coordSystem), 5
- coordSystem, PlotFeatureSet-method (coordSystem), 5
- coordSystem, trellis-method (coordSystem), 5
- dataLabels (dataLabels<-), 6
- dataLabels, ggplot-method (dataLabels<-), 6
- dataLabels, gTree-method (dataLabels<-), 6
- dataLabels, PlotFeatureSet-method (dataLabels<-), 6
- dataLabels, trellis-method (dataLabels<-), 6
- dataLabels<-, 6
- dataLabels<-, PlotFeatureSet-method (dataLabels<-), 6
- dataNames, 7
- dataTypes, 8
- dataTypes, ggplot-method (dataTypes), 8
- dataTypes, gTree-method (dataTypes), 8
- dataTypes, PlotFeatureSet-method (dataTypes), 8
- dataTypes, trellis-method (dataTypes), 8
- defaultTDB, 9
- describePackage, 9
- describePackage, FeatureSet-method (describePackage), 9
- DFFeatureSet (ObjFeatureSet), 26
- DFFeatureSet-class (FeatureSet-class), 10
- DocCollectionRef, 10
- DocCollectionRef-class (DocCollectionRef), 10
- editTags (tags<-), 37
- editTags, FeatureSet-method (tags<-), 37
- FeatureSet-class, 10
- findRecords (record), 31

- fullData, [12](#)
- fullData, ggplot-method (fullData), [12](#)
- fullData, gTree-method (fullData), [12](#)
- fullData, PlotFeatureSet-method (fullData), [12](#)
- fullData, trellis-method (fullData), [12](#)

- generateTags, [13](#)
- generateTags, ANY-method (generateTags), [13](#)
- geomObject (geomObject<-), [13](#)
- geomObject, ggplot-method (geomObject<-), [13](#)
- geomObject, GGplotFeatureSet-method (geomObject<-), [13](#)
- geomObject, gTree-method (geomObject<-), [13](#)
- geomObject, PlotFeatureSet-method (geomObject<-), [13](#)
- geomObject, trellis-method (geomObject<-), [13](#)
- geomObject<-, [13](#)
- geomObject<-, GGplotFeatureSet-method (geomObject<-), [13](#)
- gg (ggplot-class), [14](#)
- gg-class (ggplot-class), [14](#)
- ggplot-class, [14](#)
- GGplotFeatureSet (ObjFeatureSet), [26](#)
- GGplotFeatureSet-class (FeatureSet-class), [10](#)
- GraphicsFeatureSet (ObjFeatureSet), [26](#)
- GraphicsFeatureSet-class (FeatureSet-class), [10](#)
- graphSys, [15](#)
- graphSys, PlotFeatureSet-method (graphSys), [15](#)
- groupInfo, [15](#)
- groupInfo, ggplot-method (groupInfo), [15](#)
- groupInfo, gTree-method (groupInfo), [15](#)
- groupInfo, PlotFeatureSet-method (groupInfo), [15](#)
- groupInfo, trellis-method (groupInfo), [15](#)
- gTree (ggplot-class), [14](#)
- gTree-class (ggplot-class), [14](#)

- hasLegend, [16](#)
- hasLegend, ggplot-method (hasLegend), [16](#)
- hasLegend, gTree-method (hasLegend), [16](#)
- hasLegend, PlotFeatureSet-method (hasLegend), [16](#)
- hasLegend, trellis-method (hasLegend), [16](#)

- img_dir (trackr_options), [43](#)
- img_dir, TrackrDB-method (trackr_options), [43](#)
- img_dir, TrackrOptions-method (trackr_options), [43](#)
- img_ext (trackr_options), [43](#)
- img_ext, TrackrDB-method (trackr_options), [43](#)
- img_ext, TrackrOptions-method (trackr_options), [43](#)
- insert_record (trackr_lookup), [42](#)
- insert_record, ANY-method (trackr_lookup), [42](#)
- insert_record, DocCollection-method (trackr_lookup), [42](#)
- insert_record, DocCollectionRef-method (trackr_lookup), [42](#)
- insert_record, TrackrDB-method (trackr_lookup), [42](#)

- JSONBackend, [17](#)
- jsonbackend (JSONBackend), [17](#)
- JSONBackend-class (JSONBackend), [17](#)
- jsonTDB, [18](#), [18](#)

- knit_and_record, [19](#)

- ListBackend (DocCollectionRef), [10](#)
- listbackend (DocCollectionRef), [10](#)
- ListBackend-class (DocCollectionRef), [10](#)
- listTDB (jsonTDB), [18](#)

- make_image_files, [22](#)
- make_image_files, ANY-method (make_image_files), [22](#)
- make_image_files, FeatureSet-method (make_image_files), [22](#)
- make_image_files, ObjFeatureSet-method (make_image_files), [22](#)
- make_image_files, PlotFeatureSet-method (make_image_files), [22](#)
- make_image_files, RmdFeatureSet-method (make_image_files), [22](#)
- makeFeatureSet, [20](#)
- makeFeatureSet, ANY-method (makeFeatureSet), [20](#)

- makeFeatureSet, call-method
(makeFeatureSet), 20
- makeFeatureSet, data.frame-method
(makeFeatureSet), 20
- makeFeatureSet, expression-method
(makeFeatureSet), 20
- makeFeatureSet, FeatureSet-method
(makeFeatureSet), 20
- makeFeatureSet, ggplot-method
(makeFeatureSet), 20
- makeFeatureSet, gTree-method
(makeFeatureSet), 20
- makeFeatureSet, recordedplot-method
(makeFeatureSet), 20
- makeFeatureSet, trellis-method
(makeFeatureSet), 20
- manifestFromRecord, 22
- names<-, DocCollectionRef, ANY-method
(DocCollectionRef), 10
- ndoc, DocCollectionRef-method
(DocCollectionRef), 10
- ndoc, TrackrDB-method, 23
- nLayers (nLayers<-), 23
- nLayers, ggplot-method (nLayers<-), 23
- nLayers, GGplotFeatureSet-method
(nLayers<-), 23
- nLayers, gTree-method (nLayers<-), 23
- nLayers, PlotFeatureSet-method
(nLayers<-), 23
- nLayers, trellis-method (nLayers<-), 23
- nLayers<-, 23
- nLayers<-, GGplotFeatureSet-method
(nLayers<-), 23
- nObs, 24
- nObs, ggplot-method (nObs), 24
- nObs, gTree-method (nObs), 24
- nObs, PlotFeatureSet-method (nObs), 24
- nObs, trellis-method (nObs), 24
- objCode (objCode<-), 25
- objCode, FeatureSet-method (objCode<-),
25
- objCode<-, 25
- objCode<-, FeatureSet-method
(objCode<-), 25
- ObjFeatureSet, 26
- ObjFeatureSet-class (FeatureSet-class),
10
- parseCode, 29
- parseCode, ANY-method (parseCode), 29
- parseCode, call-method (parseCode), 29
- parseCode, character-method (parseCode),
29
- parseCode, expression-method
(parseCode), 29
- parseCode, ggplot-method (parseCode), 29
- parseCode, trellis-method (parseCode), 29
- plot, PlotFeatureSet, missing-method, 29
- PlotFeatureSet (ObjFeatureSet), 26
- PlotFeatureSet-class
(FeatureSet-class), 10
- position (position<-), 30
- position, ggplot-method (position<-), 30
- position, GGplotFeatureSet-method
(position<-), 30
- position, gTree-method (position<-), 30
- position, PlotFeatureSet-method
(position<-), 30
- position, trellis-method (position<-), 30
- position<-, 30
- position<-, GGplotFeatureSet-method
(position<-), 30
- prep_for_backend (trackr_lookup), 42
- prep_for_backend, FeatureSet, ANY-method
(trackr_lookup), 42
- prep_for_backend, FeatureSet, TrackrDB-method
(trackr_lookup), 42
- prep_for_backend, ObjFeatureSet, ANY-method
(trackr_lookup), 42
- print.gTree, 31
- record, 31
- recordFiles (record), 31
- regDateTime (regDateTime<-), 33
- regDateTime, FeatureSet-method
(regDateTime<-), 33
- regDateTime<-, 33
- regDateTime<-, FeatureSet-method
(regDateTime<-), 33
- remove_record (trackr_lookup), 42
- remove_record, ANY, TrackrDB-method
(trackr_lookup), 42
- remove_record, character, ANY-method
(trackr_lookup), 42
- remove_record, character, DocCollectionRef-method
(trackr_lookup), 42

- remove_record, character, TrackrDB-method
(trackr_lookup), 42
- RmdFeatureSet (ObjFeatureSet), 26
- RmdFeatureSet-class (FeatureSet-class),
10
- rmRecord (record), 31
- RStudioExtras, 34
- RStudioExtras-class (RStudioExtras), 34
- saveBasicPlot, 34
- saveBasicPlot, PlotFeatureSet-method
(saveBasicPlot), 34
- show, FeatureSet-method
(FeatureSet-class), 10
- show, GGplotFeatureSet-method
(FeatureSet-class), 10
- show, PlotFeatureSet-method
(FeatureSet-class), 10
- show, RmdFeatureSet-method
(FeatureSet-class), 10
- solrTDB (jsonTDB), 18
- statTransform (statTransform<-), 35
- statTransform, ggplot-method
(statTransform<-), 35
- statTransform, GGplotFeatureSet-method
(statTransform<-), 35
- statTransform, gTree-method
(statTransform<-), 35
- statTransform, PlotFeatureSet-method
(statTransform<-), 35
- statTransform, trellis-method
(statTransform<-), 35
- statTransform<-, 35
- statTransform<-, GGplotFeatureSet-method
(statTransform<-), 35
- summary, PlotFeatureSet-method, 36
- tags (tags<-), 37
- tags, ANY-method (tags<-), 37
- tags, FeatureSet-method (tags<-), 37
- tags, ggplot-method (tags<-), 37
- tags, gTree-method (tags<-), 37
- tags, trellis-method (tags<-), 37
- tags<-, 37
- tags<-, FeatureSet-method (tags<-), 37
- tags<-, ggplot-method (tags<-), 37
- tags<-, gTree-method (tags<-), 37
- tags<-, trellis-method (tags<-), 37
- titles (titles<-), 38
- titles, FeatureSet-method (titles<-), 38
- titles, ggplot-method (titles<-), 38
- titles, gTree-method (titles<-), 38
- titles, PlotFeatureSet-method
(titles<-), 38
- titles, trellis-method (titles<-), 38
- titles<-, 38
- titles<-, PlotFeatureSet-method
(titles<-), 38
- trackr_knit_env, 41
- trackr_lookup, 42
- trackr_lookup, ANY, ANY-method
(trackr_lookup), 42
- trackr_lookup, character, ANY-method
(trackr_lookup), 42
- trackr_lookup, character, DocCollectionRef-method
(trackr_lookup), 42
- trackr_lookup, character, TrackrDB-method
(trackr_lookup), 42
- trackr_options, 43
- trackr_options, ANY-method
(trackr_options), 43
- trackr_options, TrackrDB-method
(trackr_options), 43
- trackr_search (trackr_lookup), 42
- trackr_search, character, DocCollection-method
(trackr_lookup), 42
- trackr_search, character, DocCollectionRef-method
(trackr_lookup), 42
- trackr_search, character, SolrList-method
(trackr_lookup), 42
- trackr_search, character, TrackrDB-method
(trackr_lookup), 42
- trackr_write (trackr_lookup), 42
- trackr_write, ANY-method
(trackr_lookup), 42
- trackr_write, JSONBackend-method
(trackr_lookup), 42
- trackr_write, TrackrDB-method
(trackr_lookup), 42
- trackrAddin, 39
- TrackrDB (TrackrDB-class), 40
- TrackrDB-class, 40
- TrackrOptions (TrackrOptions-class), 40
- TrackrOptions-class, 40
- trackropts, 41
- trellis (ggplot-class), 14
- trellis-class (ggplot-class), 14

TrellisFeatureSet (ObjFeatureSet), 26

TrellisFeatureSet-class

(FeatureSet-class), 10

uniqueID (uniqueID<-), 44

uniqueID, ANY-method (uniqueID<-), 44

uniqueID, FeatureSet-method

(uniqueID<-), 44

uniqueID<-, 44

uniqueID<-, FeatureSet-method

(uniqueID<-), 44

user (user<-), 44

user, FeatureSet-method (user<-), 44

user<-, 44

user<-, FeatureSet-method (user<-), 44