List of S3 Utils

Yang Wu

10/4/2021

Contents

Testing helpers

Base type

- `base::typeof(x)` determines the (R internal) type or storage mode of any object, including non-base objects.

Object type

- `base::is.object(x)` tests if the object `x` has the R internal `OBJECT` bit set. The OBJECT bit is set when a "class" attribute is added and removed when that attribute is removed, so this is a very efficient way to check if an object has a class attribute.
- `sloop::otpye(x)` tests if x is a base, S3, S4, RC, or R6 object.

Class

Class vector

- `base::class(x)` prints the vector of names of classes an object inherits from and is safe only for `S3` and `S4` objects.
- `class(x) <- "my_class"` sets the class attributes.
- `sloop::s3_class(x)` returns the class vector that is **used** for dispatch.
- `unclass()` returns (a copy of) its argument with its class attribute removed. It is not allowed for objects which cannot be copied, namely environments and external pointers.

- `inherits(x, what, which = FALSE)` indicates whether its first argument `x` inherits from any of the classes specified in the `what` argument. If `which` is set to TRUE, then an integer vector of the same length as the `what` argument is returned. Each element of this returned integer vector indicates the positions of elements in `class(x)` that are matched by the elements of `what`; zero indicates no match. For example:

```
# Class
df <- data.table::data.table()
class(df)
```

```
## [1] "data.table" "data.frame"
```

```
# Inheritance
inherits(df, what = c("type", "data.table", "python", "data.frame"), which = TRUE)
```

```
## [1] 0 1 0 2
```

- The returned integer vector has the same length as `what`. The *first* element of `class(df)` is "data.table" and it is not matched by the first and third elements of `what`; therefore, in the returned integer vector, the first and third elements are zeros. This element is matched by the second element of `what`, and so the returned integer has the index `1` as its second element.
- The *second* element of `class(df)` is "data.frame" and it is matched by the fourth element of `what`; therefore, the returned integer vector has the index `2` as its fourth element. The index `2` indicates that this element is the second element of the `class(df)` vector.

If `which` is FALSE then TRUE is returned by inherits if any of the names in `what` matches with any of the values in `class(x)`.

Attributes

- `attributes(x)` returns the object's attribute list.
- `attributes(x) <- value` uses the list on the right-hand side of the assignment as the object's attributes.
- `attr(x, which, exact = FALSE)` provide access to a single attribute of an object. This extraction function first looks for an exact match among the attributes of `x`, then (unless exact = TRUE) a unique partial match.
- `attr(x, which) <- value` causes the named attribute to take the value specified (or create a new attribute with the value given). This replacement function only uses exact matches.

Sloop helpers

Determine if an object is a generic function

- `ftype(f)` determines whether the input function `f` (unquoted function name) is a regular/primitive/internal function, a internal/S3/S4 generic, or a S3/S4/RC method.
- `is_s3_generic(fname, env = parent.frame())` compares name checks for both internal and regular generics.
- `is_s3_method(fname, env = parent.frame())` builds names of all possible generics for that function `fname` and then checks if any of them actually is a generic.

Method dispatch

- `s3_dispatch(call, env = parent.frame())` prints a list of all possible function names that will be considered for method dispatch. There are four possible states:
  - => method exists and is found by `UseMethod(generic, object)`.
  - -> method exists and is used by `NextMethod(generic = NULL, object = NULL, ...)`.
  - * method exists but is not used.
  - nothing (and grayed out in console) method does not exist.
- `s3_get_method(name)` find S3 method from its name.

List methods for S3 generic or class

- `s3_methods_class(x)` returns the methods associated with a given class.
- `s3_methods_generic(x)` returns all methods that belong to a given generic.

Inheritance

- `UseMethod(generic, object)`. When a function calling `UseMethod("fun")` is applied to an object with class attributec`("first", "second")`, the system searches for a function called `fun.first` and, if it finds it, applies it to the object. If no such function is found a function called `fun.second` is tried. If no class name produces a suitable function, the function fun.default is used, if it exists, or an error results.
- `NextMethod(generic = NULL, object = NULL, ...)` invokes the next method (determined by the class vector, either of the object supplied to the generic, or of the first argument to the function containing `NextMethod` if a method was invoked directly.