

Convenience Functions

Janko Thyson

Thursday, October 29, 2014

Reactive objects (focus *visible* part)

Example objects

```
setReactiveS3(id = "x_1", value = 10)
setReactiveS3(id = "x_2", value = function() "object-ref: {id: x_1}")
```

Compute object UID

```
computeObjectUid(id = "x_1")
```

```
## [1] "2fc2e352f72008b90a112f096cd2d029"
```

```
computeObjectUid(id = "x_2")
```

```
## [1] "ab22808532ff42c87198461640612405"
```

Copy reactive objects

```
copyReactive(id_from = "x_1", id_to = "x_1_copy")
x_1_copy
```

```
## [1] 10
```

Get reactive

```
getReactive(id = "x_1")
```

```
## [1] 10
```

```
getReactive(id = "x_1", hidden = TRUE)
```

```
## <environment: 0x0000000009933168>
## attr(,"class")
## [1] "ReactiveObject.S3" "environment"
```

```
getReactive(id = "x_2")
```

```
## [1] 10
```

Is reactive

```
isReactive(id = "x_1")
```

```
## [1] TRUE
```

```
isReactive(id = "x_2")
```

```
## [1] TRUE
```

```
x_3 <- "hello world!"  
isReactive(id = "x_3")
```

```
## [1] FALSE
```

Unset reactive

```
unsetReactive(id = "x_1")
```

```
## [1] TRUE
```

```
isReactive(id = "x_1")
```

```
## [1] FALSE
```

Removing reactive

```
rmReactive(id = "x_1")
```

```
## [1] TRUE
```

```
exists("x_1")
```

```
## [1] FALSE
```

```
## --> already unset  
rmReactive(id = "x_2")
```

```
## [1] TRUE
```

```
exists("x_2")
```

```
## [1] FALSE
```

Reactive objects (focus *invisible* part)

Example objects

```
setReactiveS3(id = "x_1", value = 10)
setReactiveS3(id = "x_2", value = function() "object-ref: {id: x_1}")
```

Get binding function

```
getBinding(id = "x_1")
```

```
## NULL
```

```
getBinding(id = "x_2")
```

```
## function ()
## {
##   x_1 <- get(x = "x_1", envir = <environment>, inherits = FALSE)
## }
```

Get reactive

```
getReactive(id = "x_1", hidden = TRUE)
```

```
## <environment: 0x0000000009da9af0>
## attr("class")
## [1] "ReactiveObject.S3" "environment"
```

```
getReactive(id = "x_2", hidden = TRUE)
```

```
## <environment: 0x0000000009bd0978>
## attr("class")
## [1] "ReactiveObject.S3" "environment"
```

Clean up

```
rmReactive("x_1")
```

```
## [1] TRUE
```

```
rmReactive("x_2")
```

```
## [1] TRUE
```

Registry

Example objects

```
setReactiveS3(id = "x_1", value = 10)
setReactiveS3(id = "x_2", value = function() "object-ref: {id: x_1}")
```

Get registry

```
reg <- getRegistry()
ls(reg)
```

```
## [1] "2fc2e352f72008b90a112f096cd2d029" "ab22808532ff42c87198461640612405"
## [3] "ea8c305f523234921d3ff8b7c8bab390"
```

Show registry

```
showRegistry()
```

```
## [1] "2fc2e352f72008b90a112f096cd2d029" "ab22808532ff42c87198461640612405"
## [3] "ea8c305f523234921d3ff8b7c8bab390"
```

Get from registry

```
getFromRegistry(id = "x_1")
```

```
## <environment: 0x0000000009e6f7a0>
## attr(,"class")
## [1] "ReactiveObject.S3" "environment"
```

```
getFromRegistry(id = "x_2")
```

```
## <environment: 0x0000000009ab64a8>
## attr(,"class")
## [1] "ReactiveObject.S3" "environment"
```

```
getFromRegistryById(computeObjectId("x_1"))
```

```
## <environment: 0x0000000009e6f7a0>
## attr(,"class")
## [1] "ReactiveObject.S3" "environment"
```

```
getFromRegistryById(computeObjectId("x_2"))
```

```
## <environment: 0x000000009ab64a8>  
## attr(,"class")  
## [1] "ReactiveObject.S3" "environment"
```

Remove from registry

```
computeObjectId("x_1") %in% showRegistry()
```

```
## [1] TRUE
```

```
rmFromRegistry("x_1")
```

```
## [1] TRUE
```

```
computeObjectId("x_1") %in% showRegistry()
```

```
## [1] FALSE
```

```
computeObjectId("x_2") %in% showRegistry()
```

```
## [1] TRUE
```

```
rmFromRegistryById(computeObjectId("x_2"))
```

```
## [1] TRUE
```

```
computeObjectId("x_2") %in% showRegistry()
```

```
## [1] FALSE
```

Reset registry

```
## Ensure content that might previously have been removed //  
setReactiveS3(id = "x_1", value = 10)  
setReactiveS3(id = "x_2", value = function() "object-ref: {id: x_1}")  
  
## Reset //  
showRegistry()
```

```
## [1] "2fc2e352f72008b90a112f096cd2d029" "ab22808532ff42c87198461640612405"  
## [3] "ea8c305f523234921d3ff8b7c8bab390"
```

```
resetRegistry()
```

```
## [1] TRUE
```

```
showRegistry()
```

```
## character(0)
```