

# Package ‘annotator’

January 5, 2023

**Title** Image Annotation and Polygon Outlining using Free Drawing

**Version** 0.0.3

**Description** Provides functions to create image annotations through polygon outlining. Annotator has the same function as 'graphics::locator()' but achieves its purpose through drawing, rather than multiple mouse clicks. It is based on the 'htmlwidgets' package and 'fabric.js' JavaScript library <<http://fabricjs.com/>>.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**URL** <https://github.com/cran/annotator>

**BugReports** <https://github.com/cran/annotator/issues>

**Depends** R (>= 4.0.0),

**Imports** methods, utils, shiny, htmlwidgets, xfun

**Suggests** testthat (>= 3.0.0), stringr, glue, bslib, sf, terra, keys, colourpicker, spelling

**Config/testthat/edition** 3

**Language** en-US

**NeedsCompilation** no

**Author** Mihai Valcu [aut, cre] (<<https://orcid.org/0000-0002-6907-7802>>)

**Maintainer** Mihai Valcu <mvalcu@gwdg.de>

**Repository** CRAN

**Date/Publication** 2023-01-05 15:30:02 UTC

## R topics documented:

annotate . . . . .	2
annotator . . . . .	3
annotatorOutput . . . . .	3
empty_png . . . . .	4
renderAnnotator . . . . .	4
runExample . . . . .	5

---

annotate	<i>Create an annotation widget</i>
----------	------------------------------------

---

### Description

This function creates an annotation using [htmlwidgets](#). The widget can be rendered on HTML pages generated from Shiny or other applications.

### Usage

```
annotate(  
  im,  
  resultId = "annot_id",  
  brushWidth = 4,  
  brushColor = "red",  
  opacity = 0.5,  
  fill = "grey"  
)
```

### Arguments

<code>im</code>	the input image. If missing, a transparent 800x600 png is used.
<code>resultId</code>	the id of the div in the UI where the annotation (the drawn polygon) is stored. Only relevant when the widget is used in shiny. Defaults to "annot_id".
<code>brushWidth</code>	default to 3 pixels.
<code>brushColor</code>	Any valid CSS color. For example "red", "rgba(255,93,0,1)", or "#1c4564".
<code>opacity</code>	default to 0.5.
<code>fill</code>	default to 'gray'

### Value

An object generated by [htmlwidgets::createWidget\(\)](#).

### Examples

```
if (interactive()) {  
  require(annotator)  
  im = system.file("sample_images", "aves", "5.png", package = "annotator")  
  annotate(im)  
}
```

---

annotator	<i>Image annotation</i>
-----------	-------------------------

---

**Description**

Image annotation using free drawing.

**Author(s)**

Mihai Valcu <mvalcu@gwdg.de>

---

annotatorOutput	<i>Widget output function for use in Shiny</i>
-----------------	------------------------------------------------

---

**Description**

Widget output function for use in Shiny

**Usage**

```
annotatorOutput(outputId, width = "auto", height = "auto", ...)
```

**Arguments**

outputId	The name of the input.
width	in CSS units, default to "auto".
height	in CSS units, default to "auto".
...	further arguments to pass to <a href="#">htmlwidgets::shinyWidgetOutput()</a> e.g. inline.

**Value**

An object generated by [htmlwidgets::shinyWidgetOutput\(\)](#).

**See Also**

[runExample\(\)](#)

**Examples**

```
# Print the directory containing the code for all examples to see this function in use.  
system.file("examples", package = "annotator")
```

---

empty_png	<i>this is an 800x600 transparent png</i>
-----------	-------------------------------------------

---

**Description**

this is an 800x600 transparent png

**Usage**

```
empty_png()
```

---

renderAnnotator	<i>Widget render function for use in Shiny</i>
-----------------	------------------------------------------------

---

**Description**

Widget render function for use in Shiny

**Usage**

```
renderAnnotator(expr, env = parent.frame(), quoted = FALSE)
```

**Arguments**

expr	An annotator expression.
env	A environment. Default to <code>parent.frame()</code> .
quoted	A boolean value.

**Value**

An object generated by `htmlwidgets::shinyRenderWidget()`.

**See Also**

[runExample\(\)](#)

**Examples**

```
# Print the directory containing the code for all examples to see this function in use.  
system.file("examples", package = "annotator")
```

---

`runExample`*Run Shiny Examples*

---

**Description**

Launch Shiny example applications bundled in annotator.

**Usage**

```
runExample(example, ...)
```

**Arguments**

<code>example</code>	The name of the example to run. Available are: 01_hello, 02_pipeline
<code>...</code>	further arguments to pass to <code>shiny::runApp()</code>

**Value**

No return value, called for its side effects.

**Examples**

```
## Only run this example in interactive R sessions
if (interactive()) {
  annotator::runExample("01_hello")
  annotator::runExample("02_pipeline")

  # Print the directory containing the code for all examples
  system.file("examples", package = "annotator")
}
```

# Index

annotate, [2](#)  
annotator, [3](#)  
annotatorOutput, [3](#)

empty\_png, [4](#)

htmlwidgets, [2](#)  
htmlwidgets::createWidget(), [2](#)  
htmlwidgets::shinyRenderWidget(), [4](#)  
htmlwidgets::shinyWidgetOutput(), [3](#)

renderAnnotator, [4](#)  
runExample, [5](#)  
runExample(), [3](#), [4](#)

shiny::runApp(), [5](#)