package-lock.json	v1.6.3	3 days ago
package.json	v1.6.3	3 days ago
vercel.json	server rendered svg - verbal routing	2 years ago

README MIT license

nomnoml

Hello, this is nomnoml, a tool for drawing UML diagrams based on a simple syntax. It tries to keep its syntax visually as close as possible to the generated UML diagram without resorting to ASCII drawings.

Created by Daniel Kallin with help from a group of contributors.

Library

The nomnoml javascript library can render diagrams on your web page. The only dependency is graphre. Install nomnoml using either *npm* or good old script inclusion.

SVG output in NodeJS

```
npm install nomnoml
```

```
var nomnoml = require('nomnoml')
var src = '[nomnoml] is -> [awesome]'
console.log(nomnoml.renderSvg(src))
```

In the SVG output the node name is attached to SVG shapes and `<g>` containers with `data-name` attribute. You can use this to implement interactive diagrams.

```
document.querySelector('svg').onclick = function (e) {
  console.log(e.target.closest('g[data-name]')?.attributes['data-name'])
}
```

HTML Canvas rendering target

```
<script src="//unpkg.com/graphre/dist/graphre.js"></script>
<script src="//unpkg.com/nomnoml/dist/nomnoml.js"></script>

<canvas id="target-canvas"></canvas>
<script>
  var canvas = document.getElementById('target-canvas')
  var source = '[nomnoml] is -> [awesome]'
  nomnoml.draw(canvas, source)
</script>
```

Command Line Interface

`npx nomnoml` exposes the SVG renderer with a command-line interface. This mode also supports the `#import: <filename>` directive for dividing complex diagrams into multiple files.

```
npx nomnoml input-file.noml
```

Web application

The nomnoml web application is a simple editor with a live preview. It is purely client-side and uses your browser's *localStorage*, so your diagram should be here the next time you visit (but no guarantees).

Example

This is how the Decorator pattern can look in nomnoml syntax:

```
[<frame>Decorator pattern|
  [<abstract>Component||+ operation()]
  [Client] depends --> [Component]
  [Decorator|- next: Component]
  [Decorator] decorates -- [ConcreteComponent]
  [Component] <:- [Decorator]
  [Component] <:- [ConcreteComponent]
]
```

Association types

```
-    association
->   association
<->  association
-->  dependency
<--> dependency
-:>  generalization
<:-  generalization
--:> implementation
<:-- implementation
+-   composition
+->  composition
o-   aggregation
o->  aggregation
-o)  ball and socket
o<-) ball and socket
->o  ball and socket
--   note
-/-  hidden
```

Classifier types

```
[name]
[<abstract> name]
[<instance> name]
[<reference> name]
[<note> name]
[<package> name]
[<frame> name]
[<database> name]
[<pipe> name]
```

```
[<start> name]
[<end> name]
[<state> name]
[<choice> name]
[<sync> name]
[<input> name]
[<lollipop> lollipop]
[<sender> name]
[<socket> socket]
[<receiver> name]
[<transceiver> name]
[<actor> name]
[<usecase> name]
[<label> name]
[<hidden> name]
[<table> name| a | 5 || b | 7]
```

Comments

Comments are supported at the start of a line.

```
//[commented]
[not //commented]
```

Id attribute

Two distinct nodes can have the same display name with the id attribute.

```
[<actor id=a>User]
[<actor id=b>User]
[a] -- [b]
```

Directives

```
#import: my-common-styles.nomnoml
#arrowSize: 1
#bendSize: 0.3
#direction: down | right
#gutter: 5
#edgeMargin: 0
#gravity: 1
#edges: hard | rounded
#background: transparent
#fill: #eee8d5; #fdf6e3
#fillArrows: false
#font: Calibri
#fontSize: 12
#leading: 1.35
#lineWidth: 3
#padding: 8
#spacing: 40
#stroke: #33322E
#title: filename
#zoom: 1
#acyclicer: greedy
#ranker: network-simplex | tight-tree | longest-path
```

Custom classifier styles

A directive that starts with "." define a classifier style. The style is written as a space separated list of modifiers and key/value pairs.

```
#.box: fill=#8f8 dashed
#.blob: visual=ellipse title=bold
[<box> GreenBox]
[<blob> HideousBlob]
```

Modifiers

```
dashed
empty
```

Key/value pairs

```
fill=(any css color)

stroke=(any css color)

align=center
align=left

direction=right
direction=down

visual=actor
visual=class
visual=database
visual=ellipse
visual=end
visual=frame
visual=hidden
visual=input
visual=none
visual=note
visual=package
visual=pipe
visual=receiver
visual=rhomb
visual=roundrect
visual=sender
visual=start
visual=table
visual=transceiver
```

Style title and text body with a comma separated list of text modifiers

```
title=left,italic,bold
body=center,italic,bold
```

Text modifiers

```
bold
center
italic
left
underline
```