```
                        \\\///
                       / _  _ \
                     (| (.)(.) |)
.------------------.OOOo--()--oOOO.-----------------.
|                                                   |
|           _             _                   _     |
|     _ __ | | __ _ _ __ | |_ _   _ _ __ ___ | |    |
|    | '_ \| |/ _` | '_ \| __| | | | '_ ` _ \| |    |
|    | |_) | | (_| | | | | |_| |_| | | | | | | |    |
|    | .__/|_|\__,_|_| |_|\__|\__,_|_| |_| |_|_|    |
|    |_|                                            |
|                                                   |
'-----------------.oooO-----------------------------'
                   (   )   Oooo.
                    \ (    (   )
                     \_)    ) /
                           (_/
```

```
Codeblock class: plantuml

    sudo apt-get install plantuml
    http://plantuml.com

    runs:
    > plantuml -t{im_fmt} <fname>.plantuml {im_opt}

    class->cmd
      plantuml -> plantuml

Metadata options
    imagine.im_out: img,fcb
    imagine.im_log: 4
    imagine.plantuml.im_fmt: svg
    imagine.plantuml.im_opt: width=60%
```

Notes:

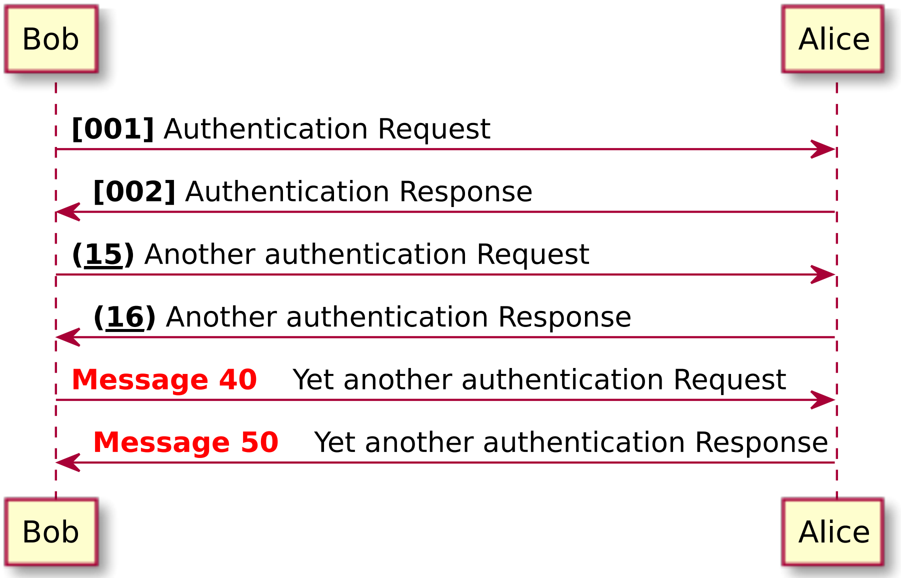
- metadata sets plantuml's default output format to `svg`
- local install doesn't seem to support mindmap
  see `plantuml -language | grep -i @`

Local installation

```
GNU/Linux
4.15.0-70-generic #79-Ubuntu SMP Tue Nov 12 10:36:11 UTC 2019

Package: plantuml
Version: 1:1.2017.15-1
Priority: optional
Section: universe/graphics
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Andrew Shadura <andrewsh@debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 6321 kB
Depends: default-jre, libfop-java, libbatik-java, libjlatexmath-java
Recommends: graphviz
Homepage: http://plantuml.com/
Download-Size: 5683 kB
APT-Manual-Installed: yes
APT-Sources: http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages
Description: text-to-UML converter
 PlantUML is a program allowing to draw UML diagrams, using a simple
 human readable text description.
 .
 PlantUML supports the following diagram types:
   - sequence diagram
   - use case diagram
   - class diagram
   - activity diagram
   - component diagram
   - state diagram
 .
 Output images can be generated in PNG, in SVG or LaTeX format.
 PlantUML also supports generation of ASCII art diagrams (only for
 sequence diagrams).
```

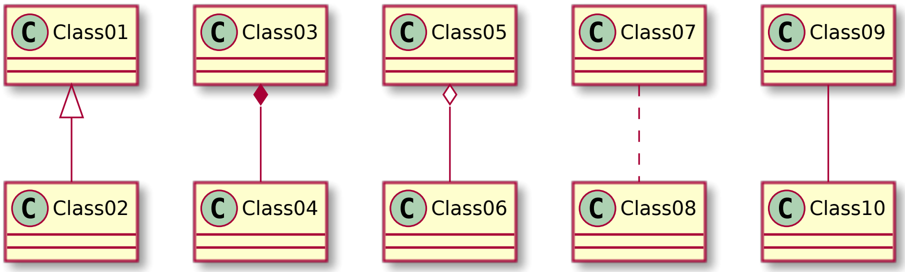
Examples

Sequence diagram (svg)


Bob
Alice
[001] Authentication Request
[002] Authentication Response
(15) Another authentication Request
(16) Another authentication Response
Message 40 Yet another authentication Request
Message 50 Yet another authentication Response
Bob
Alice

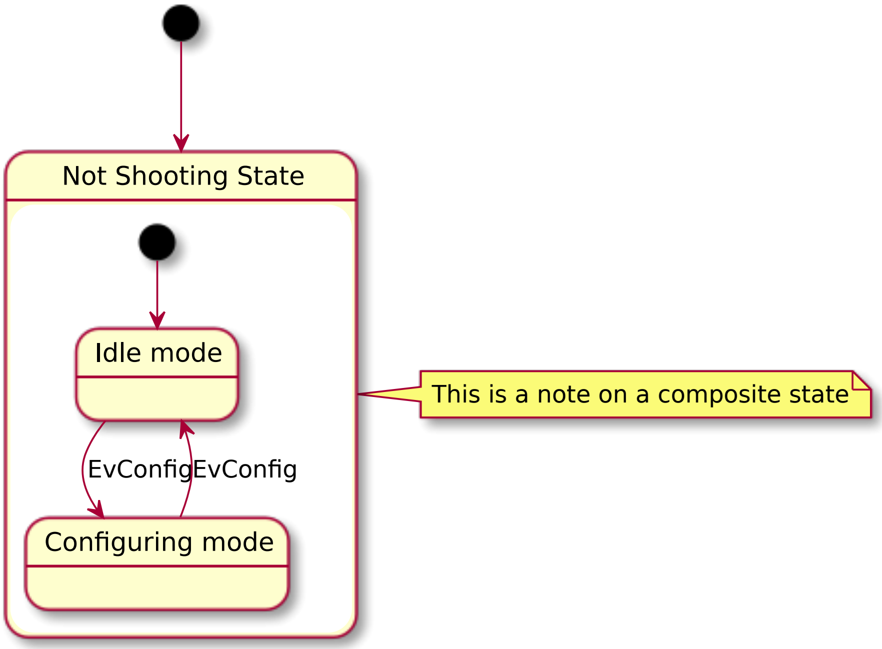
~~~
```plantuml
@startuml
autonumber "<b>[000]"
Bob -> Alice : Authentication Request
Bob <- Alice : Authentication Response

autonumber 15 "<b>(<u>##</u>)"
Bob -> Alice : Another authentication Request
Bob <- Alice : Another authentication Response

autonumber 40 10 "<font color=red><b>Message 0  "
Bob -> Alice : Yet another authentication Request
Bob <- Alice : Yet another authentication Response

@enduml
```
~~~

Class diagrams (svg)



~~~
```plantuml
@startuml
Class01 <|-- Class02
Class03 *-- Class04
Class05 o-- Class06
Class07 .. Class08
Class09 -- Class10
@enduml
```
~~~

Notes (svg)



```plantuml
@startuml
[*] --> NotShooting
state "Not Shooting State" as NotShooting {
state "Idle mode" as Idle
state "Configuring mode" as Configuring
[*] --> Idle
Idle --> Configuring : EvConfig
Configuring --> Idle : EvConfig
}
note right of NotShooting : This is a note on a composite state
@enduml
```

Documentation

plantuml -h

```
Usage: java -jar plantuml.jar [options] -gui
(to execute the GUI)
    or java -jar plantuml.jar [options] [file/dir] [file/dir] [file/dir]
(to process files or directories)

You can use the following wildcards in files/dirs:
* means any characters but '/'
? one and only one character but '/'
** means any characters (used to recurse through directories)

where options include:
    -gui To run the graphical user interface
    -tpng To generate images using PNG format (default)
    -tsvg To generate images using SVG format
    -teps To generate images using EPS format
    -tpdf To generate images using PDF format
    -tvdx To generate images using VDX format
    -txmi To generate XMI file for class diagram
    -tscxml To generate SCXML file for state diagram
    -thtml To generate HTML file for class diagram
    -ttxt To generate images with ASCII art
    -tutxt To generate images with ASCII art using Unicode characters
    -tlatex To generate images using LaTeX/Tikz format
    -tlatex:nopreamble To generate images using LaTeX/Tikz format without preamble
    -o[utput] "dir" To generate images in the specified directory
    -DVAR1=value To set a preprocessing variable as if '!define VAR1 value' were used
    -Sparam1=value To set a skin parameter as if 'skinparam param1 value' were used
    -r[ecurse] recurse through directories
    -I/path/to/file To include file as if '!include file' were used
    -I/path/to/*.puml To include files with pattern
    -charset xxx To use a specific charset (default is UTF-8)
    -e[x]clude pattern To exclude files that match the provided pattern
    -metadata To retrieve PlantUML sources from PNG images
    -nometadata To NOT export metadata in PNG/SVG generated files
    -version To display information about PlantUML and Java versions
    -checkversion To check if a newer version is available for download
    -v[erbose] To have log information
    -quiet To NOT print error message into the console
    -debugsvek To generate intermediate svek files
    -h[elp] To display this help message
    -testdot To test the installation of graphviz
    -graphvizdot "exe" To specify dot executable
    -p[ipe] To use stdin for PlantUML source and stdout for PNG/SVG/EPS generation
    -encodesprite 4|8|16[z] "file" To encode a sprite at gray level (z for compression) from
    -computeurl|-encodeurl To compute the encoded URL of a PlantUML source file
    -decodeurl To retrieve the PlantUML source from an encoded URL
    -syntax To report any syntax error from standard input without generating images
    -language To print the list of PlantUML keywords
    -nosuggestengine To disable the suggest engine when errors in diagrams
    -checkonly To check the syntax of files without generating images
    -failfast To stop processing as soon as a syntax error in diagram occurs
    -failfast2 To do a first syntax check before processing files, to fail even faster
    -pattern To print the list of Regular Expression used by PlantUML
    -duration To print the duration of complete diagrams processing
    -nbthread N To use (N) threads for processing
    -nbthread auto To use 4 threads for processing
    -timeout N Processing timeout in (N) seconds. Defaults to 15 minutes (900 seconds).
    -author[s] To print information about PlantUML authors
    -overwrite To allow to overwrite read only files
    -printfonts To print fonts available on your system
    -enablestats To enable statistics computation
    -disablestats To disable statistics computation (default)
    -htmlstats To output general statistics in file plantuml-stats.html
    -xmlstats To output general statistics in file plantuml-stats.xml
    -realtimestats To generate statistics on the fly rather than at the end
    -loopstats To continuously print statistics about usage
    -splash To display a splash screen with some progress bar
    -progress To display a textual progress bar in console
    -pipeimageindex N To generate the Nth image with pipe option

If needed, you can setup the environment variable GRAPHVIZ_DOT.
```

man page

```
PLANTUML(1)              General Commands Manual              PLANTUML(1)

NAME
       plantuml - program to generate UML diagram from a text description

SYNOPSIS
       plantuml [options] files/dirs...

DESCRIPTION
       PlantUML  is  used  to  draw UML diagram, using a simple and human
       readable text description.

       Notice that graphviz is required to be installed for generation of
       all diagrams but sequence diagrams.

       When  files/directories  are  given,  they  are processed in batch
       mode. Otherwise GUI window  appears.  It  allows  user  to  browse
       directories  and view diagrams generated from suitable files (must
       have one of the following extensions:  .txt,  .tex,  .java,  .htm,
       .html, .c, .h, .cpp).

       You can use the following wildcards in files/dirs:

       *      means any characters but '/'

       ?      one and only one character but '/'

       **     means any characters (used to recurse through directories)

OPTIONS
       -tsvg  Generate images using SVG format.

       -output dir, -o dir
              Generate images in the specified directory.

       -config file
              Read the provided config file before each diagram.

       -charset xxx
              Use a specific charset (default is UTF-8).

       -exclude pattern, -x pattern
              Exclude files that match the provided pattern.
```

-metadata
Retrieve PlantUML sources from PNG images.

-version
Display information about PlantUML and Java versions.

-verbose, -v
Have log information.

-keepfiles
NOT delete temporary files after process.

-help, -h
Display help message.

-testdot
Test the installation of graphviz.

-graphvizdot path
Specify dot path.

-pipe, -p
Use stdin for PlantUML source and stdout for PNG/SVG generation.

-computeurl
Compute the encoded URL of a PlantUML source file.

-decodeurl
Retrieve the PlantUML source from an encoded URL.

SEE ALSO

graphviz(1).

AUTHOR

PlantUML was written by Arnaud Roques <plantuml@gmail.com>.

This manual page was written by Ilya Paramonov <ivparamonov@gmail.com>, for the Debian project (and may be used by others).