plantuml-babel.org

Derek Feichtinger

August 18, 2021

Contents

1 Links

- Homepage: `http://plantuml.com/`
- Downloads: `http://plantuml.sourceforge.net/`
- Source code: `https://github.com/plantuml/plantuml`
- Language Reference: `http://plantuml.com/PlantUML_Language_Reference_Guide.pdf`

2 Information on the local installation

```
Emacs version: GNU Emacs 27.1.90 (build 1, x86_64-pc-linux-gnu, GTK+ Version 3.22.30)
 of 2021-01-26
org version: 9.4.5

Emacs variable org-plantuml-jar-path:/home/dfeich/.emacs.d/javalib/plantuml.jar


PlantUML version 1.2021.9 (Sun Jul 25 12:13:56 CEST 2021)
(GPL source distribution)
Java Runtime: OpenJDK Runtime Environment
JVM: OpenJDK 64-Bit Server VM
Default Encoding: UTF-8
Language: en
Country: US

PLANTUML_LIMIT_SIZE: 4096

Dot version: dot - graphviz version 2.40.1 (20161225.0304)
Installation seems OK. File generation OK
```

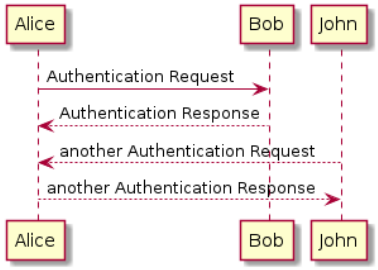
2.1 Help text

```
java -jar "$jpath" -help
```

3 simple test

```
@startuml
' this is a comment
```

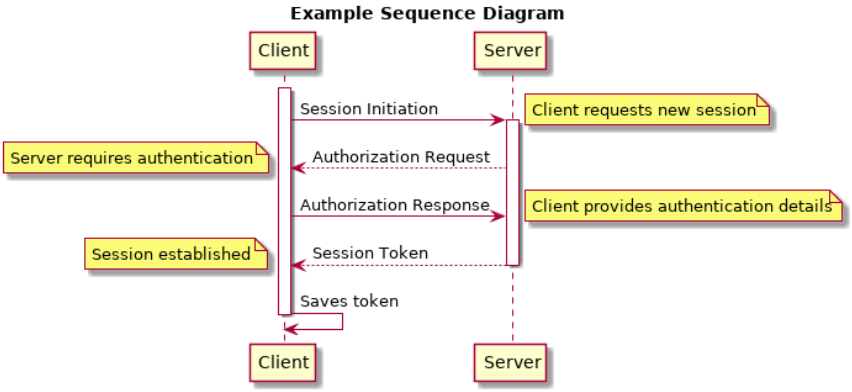
```
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
John --> Alice: another Authentication Request
Alice --> John: another Authentication Response
@enduml
```

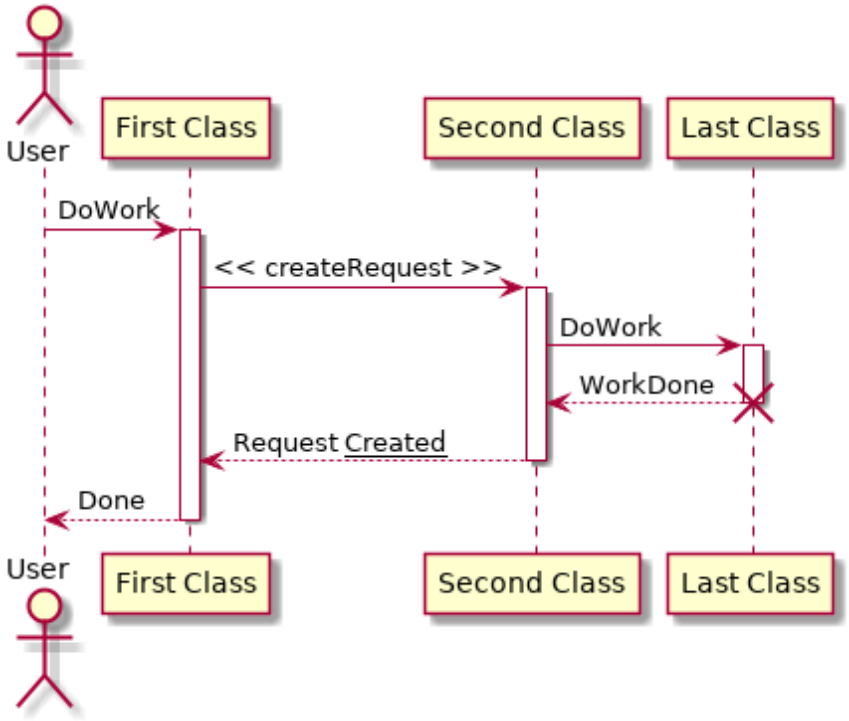


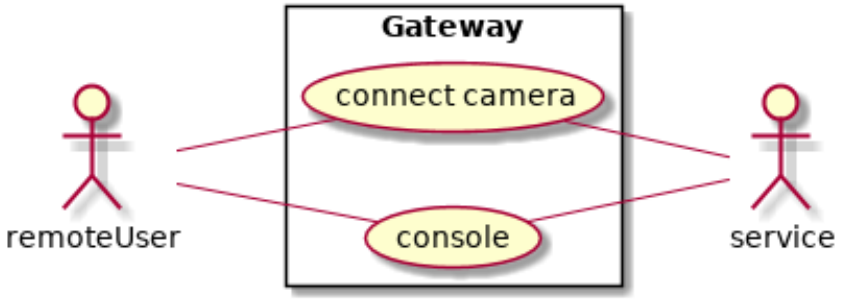
4 Diagram type examples

4.1 sequence diagrams

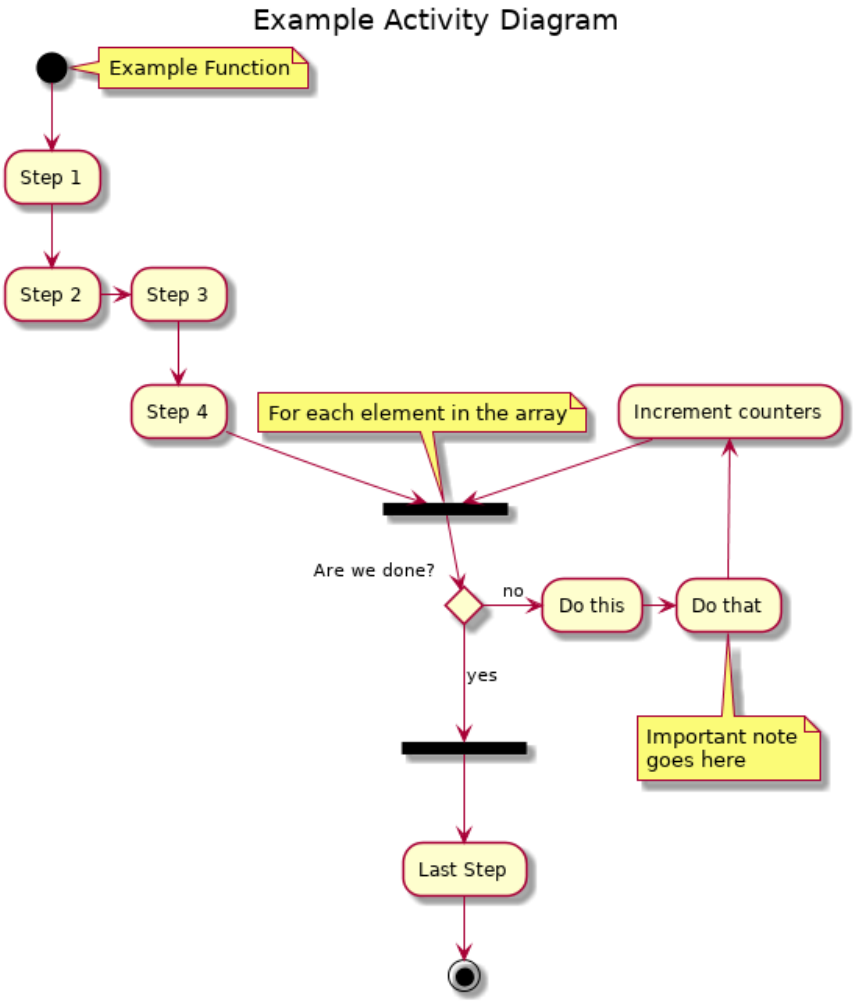
Note: The `skin` parameter I used in the earlier versions of this document is no longer supported by plantuml.

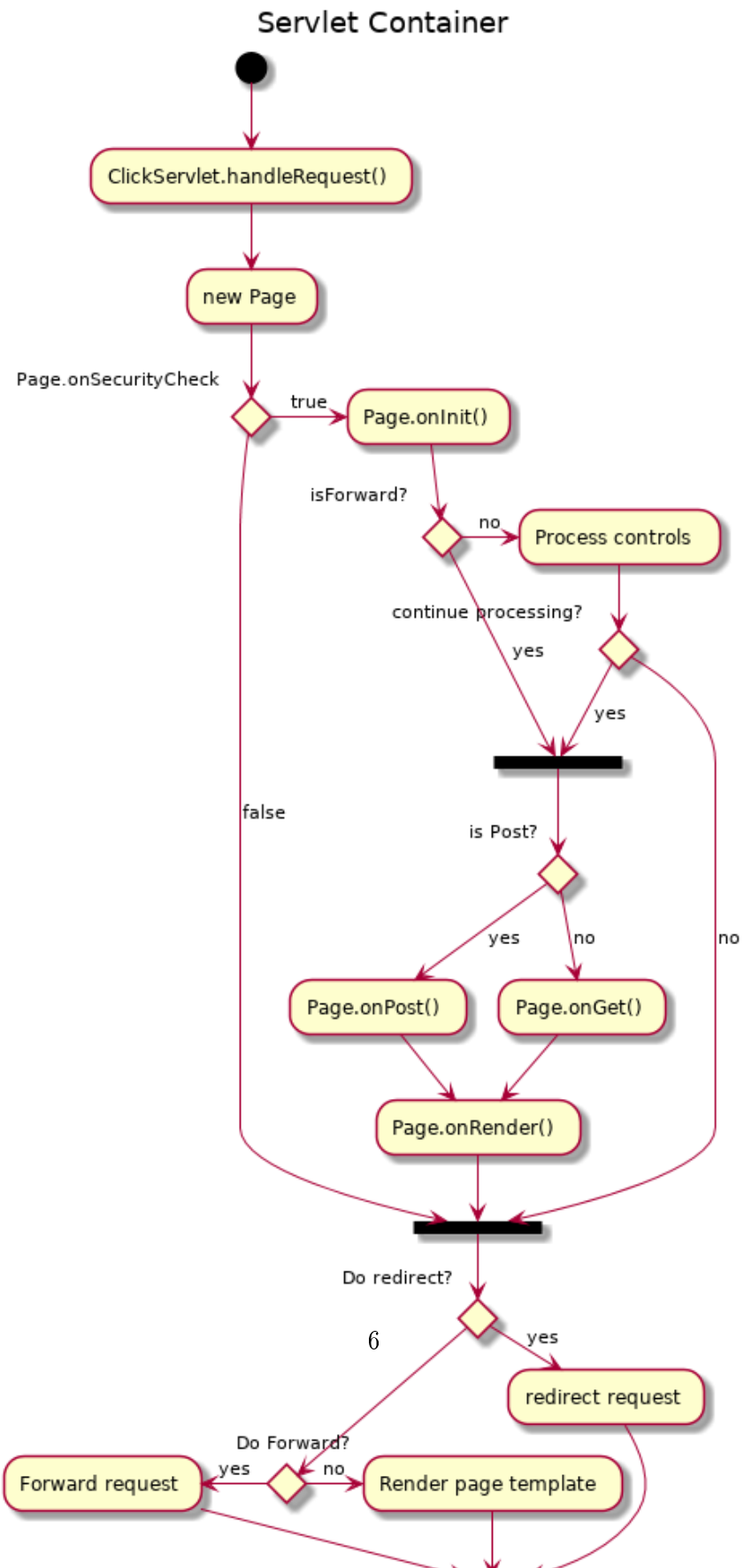



User
First Class
Second Class
Last Class
DoWork
<< createRequest >>
DoWork
WorkDone
Request Created
Done
User
First Class
Second Class
Last Class


Gateway
connect camera
console
remoteUser
service

4.2 old style activity diagrams



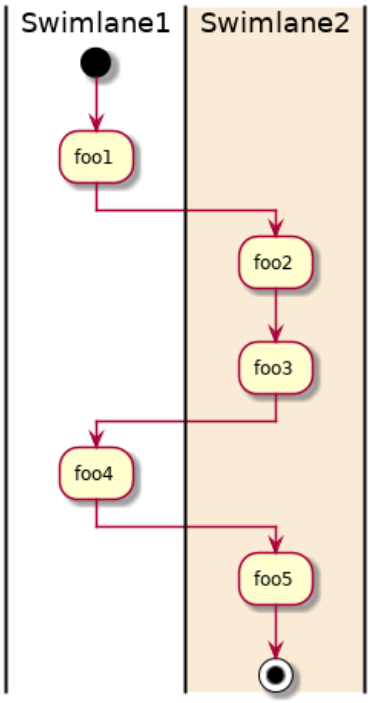

Servlet Container
ClickServlet.handleRequest()
new Page
Page.onSecurityCheck
true
Page.onInit()
isForward?
no
Process controls
continue processing?
yes
yes
false
is Post?
yes
no
no
Page.onPost()
Page.onGet()
Page.onRender()
Do redirect?
yes
redirect request
Do Forward?
yes
no
Forward request
Render page template

4.3 new style activity diagrams

- `http://plantuml.sourceforge.net/activity2.html`

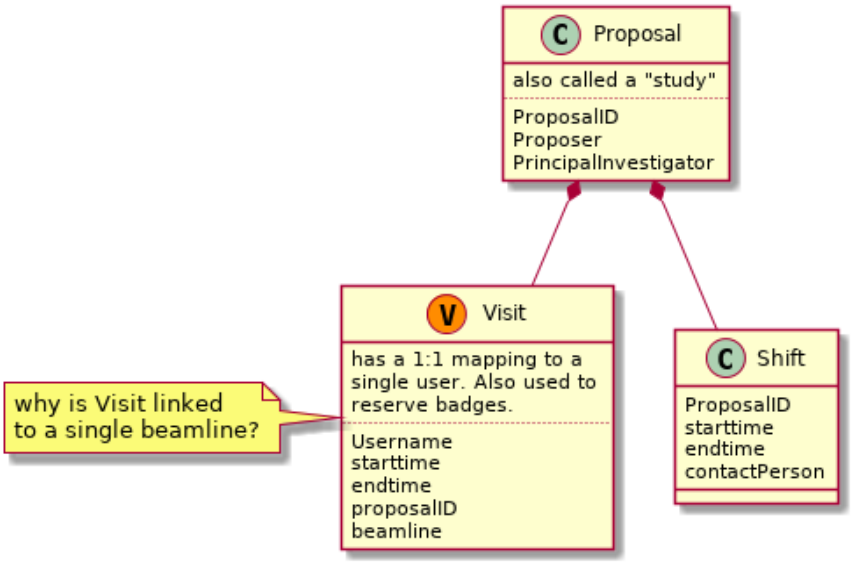
4.3.1 swimlanes

Swimlanes actually are activity diagrams using the new syntax.



4.4 Class diagrams

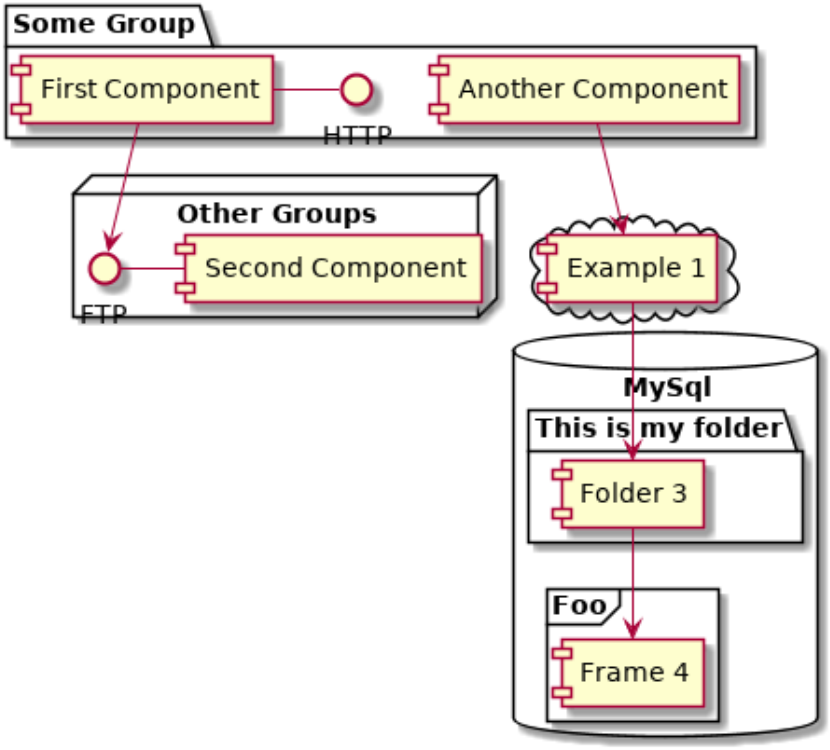
`http://plantuml.sourceforge.net/classes.html`



4.5 Component diagrams

```
@startuml

package "Some Group" {
    HTTP - [First Component]
    [Another Component]
}

node "Other Groups" {
    FTP - [Second Component]
    [First Component] --> FTP
    }

    cloud {
            [Example 1]
        }


        database "MySql" {
                    folder "This is my folder" {
                            [Folder 3]
                            }
```

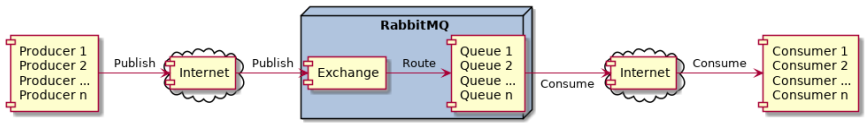
```
frame "Foo" {
        [Frame 4]
        }
        }


        [Another Component] --> [Example 1]
        [Example 1] --> [Folder 3]
        [Folder 3] --> [Frame 4]

        @enduml
```



The next example was posted by Cecil Westerhof on emacs-orgmode.gnu.org mailing list *2019-10-18 Fri*

```
@startuml

component [Producer 1\nProducer 2\nProducer ...\nProducer n] as Producers
```

```
cloud {
    [Internet] as Internet1
}

node RabbitMQ #LightSteelBlue {
    [Exchange]
    [Queue 1\nQueue 2\nQueue ...\nQueue n] as Queues
}

cloud {
    [Internet] as Internet2
}

[Consumer 1\nConsumer 2\nConsumer ...\nConsumer n] as Consumers


[Producers] -> [Internet1]   : Publish
[Internet1] -> [Exchange]    : Publish
[Exchange]  -> [Queues]      : Route
[Queues]    -> [Internet2]   : Consume
[Internet2] -> [Consumers]   : Consume

@enduml
```



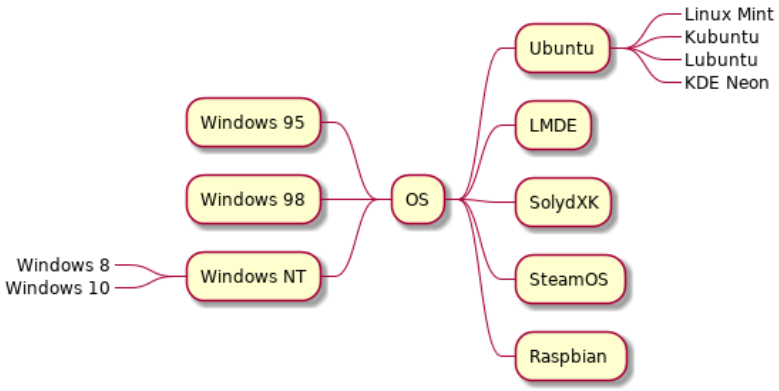
4.6 Mindmaps

- *2019-07-21 Sun* Needs plantuml-1.2019.08 or newer. Still in testing and features may change
- `http://plantuml.com/mindmap-diagram`
- Nice Link about mindmaps in PlantUML: `http://hangaroundtheweb.com/2019/07/mind-maps-in-spacemacs/`

The examples are taken from the official plantuml page.
This syntax looks like the most versatile and useful to me

- Leading "+/-" specify hierarchy level and whether the node is on the right or left of the central node.
- Undescores directly following the leading position characters prevent the creation of a box around an item.

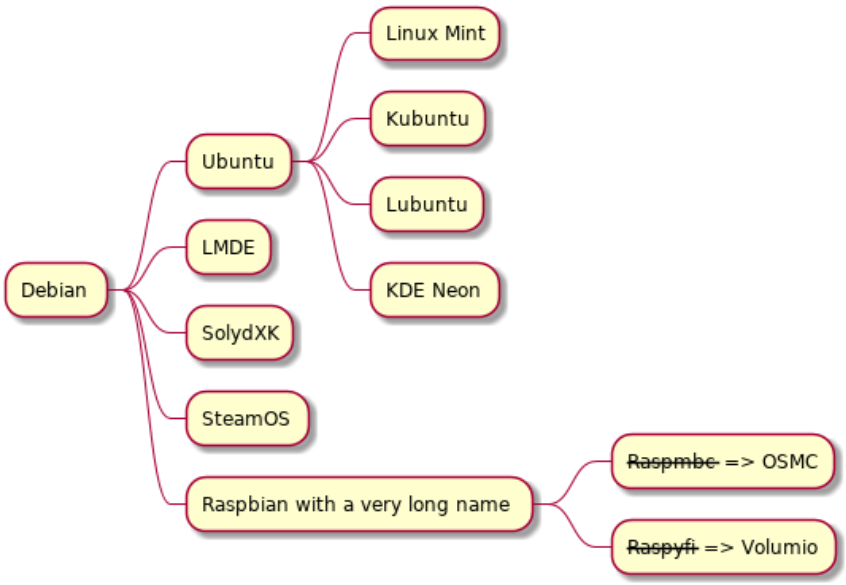
```
@startmindmap
+ OS
++ Ubuntu
+++_ Linux Mint
+++_ Kubuntu
+++_ Lubuntu
+++_ KDE Neon
++ LMDE
++ SolydXK
++ SteamOS
++ Raspbian
-- Windows 95
-- Windows 98
-- Windows NT
---_ Windows 8
---_ Windows 10
@endmindmap
```



A mindmap based on org mode syntax. Note that the org headline asterisks need to be escaped by "," inside of a source block. It's nice that they allow for an org mode syntax, but I think this is less convenient to

write and work with. The org headlines do not allow for text following them (syntax error).

```
@startmindmap
* Debian
** Ubuntu
*** Linux Mint
*** Kubuntu
*** Lubuntu
*** KDE Neon
** LMDE
** SolydXK
** SteamOS
** Raspbian with a very long name
*** <s>Raspmbc</s> => OSMC
*** <s>Raspyfi</s> => Volumio
@endmindmap
```




```
@startmindmap
caption figure 1
title My super title
```

```
* <&flag>Debian
** <&globe>Ubuntu
*** Linux Mint
*** Kubuntu
*** Lubuntu
*** KDE Neon
** <&graph>LMDE
** <&pulse>SolydXK
** <&people>SteamOS
** <&star>Raspbian with a very long name
*** <s>Raspmbc</s> => OSMC
*** <s>Raspyfi</s> => Volumio

header
My super header
endheader

center footer My super footer

legend right
  Short
  legend
endlegend
@endmindmap
```

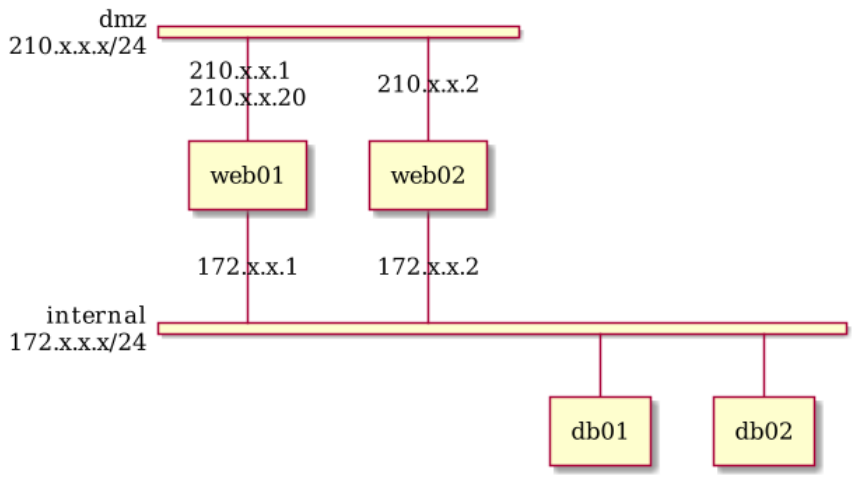
figure 1

My super footer

4.7 network

```
@startuml
scale 1.5
nwdiag {
   network dmz {
              address = "210.x.x.x/24"

              // set multiple addresses (using comma)
              web01 [address = "210.x.x.1, 210.x.x.20"];
              web02 [address = "210.x.x.2"];
              }
              network internal {
                    address = "172.x.x.x/24";

                    web01 [address = "172.x.x.1"];
                    web02 [address = "172.x.x.2"];
```

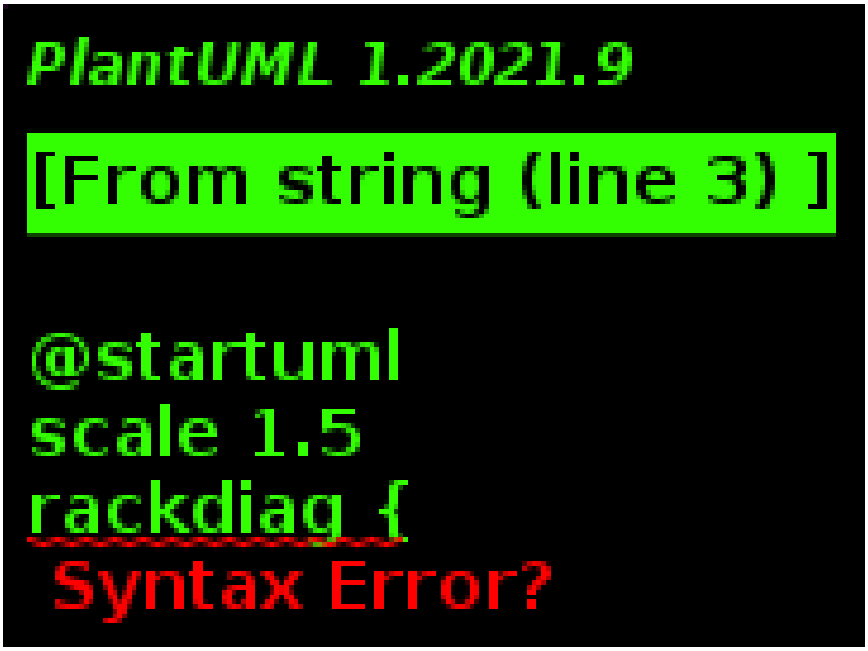
```
                db01;
                db02;
            }
            }
@enduml
```



Regrettably rackdiag is not yet (*2021-08-18 Wed*) part of the functionality that was integrated in plantuml from nwdiag. q.v. `http://blockdiag.com/en/nwdiag/rackdiag-examples.html`

```
@startuml
scale 1.5
rackdiag {
  // define height of rack
  16U;

  // define rack items
  1: UPS [2U];
  3: DB Server
  4: Web Server
  5: Web Server
  6: Web Server
  7: Load Balancer
  8: L3 Switch
```

```
}
@enduml
```

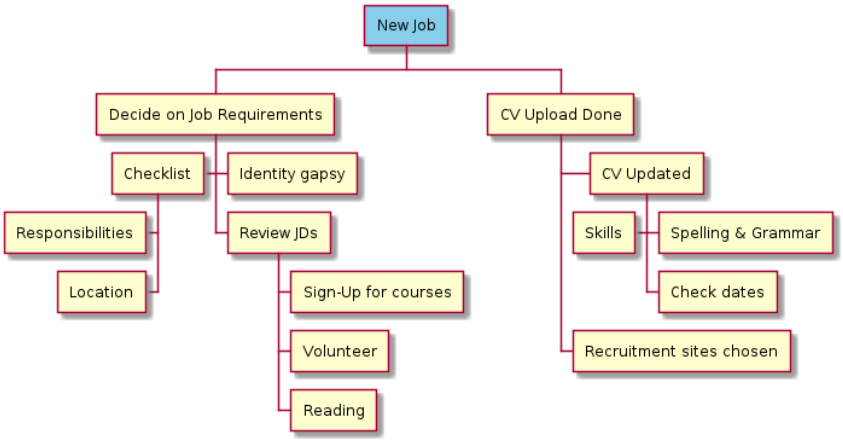


4.8 Work Breakdown Structure (WBS)

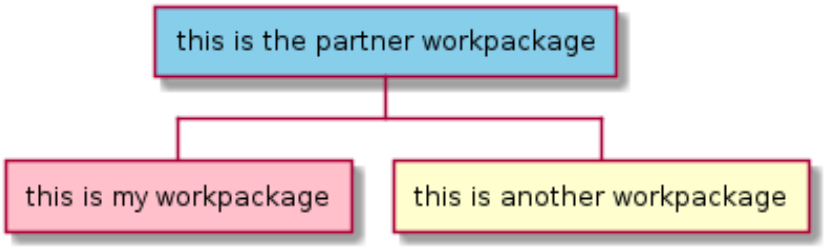
https://plantuml.com/wbs-diagram

```
@startwbs
+[#SkyBlue] New Job
++ Decide on Job Requirements
+++ Identity gapsy
+++ Review JDs
++++ Sign-Up for courses
++++ Volunteer
++++ Reading
++- Checklist
+++- Responsibilities
+++- Location
++ CV Upload Done
+++ CV Updated
++++ Spelling & Grammar
++++ Check dates
```

```
---- Skills
+++ Recruitment sites chosen
@endwbs
```

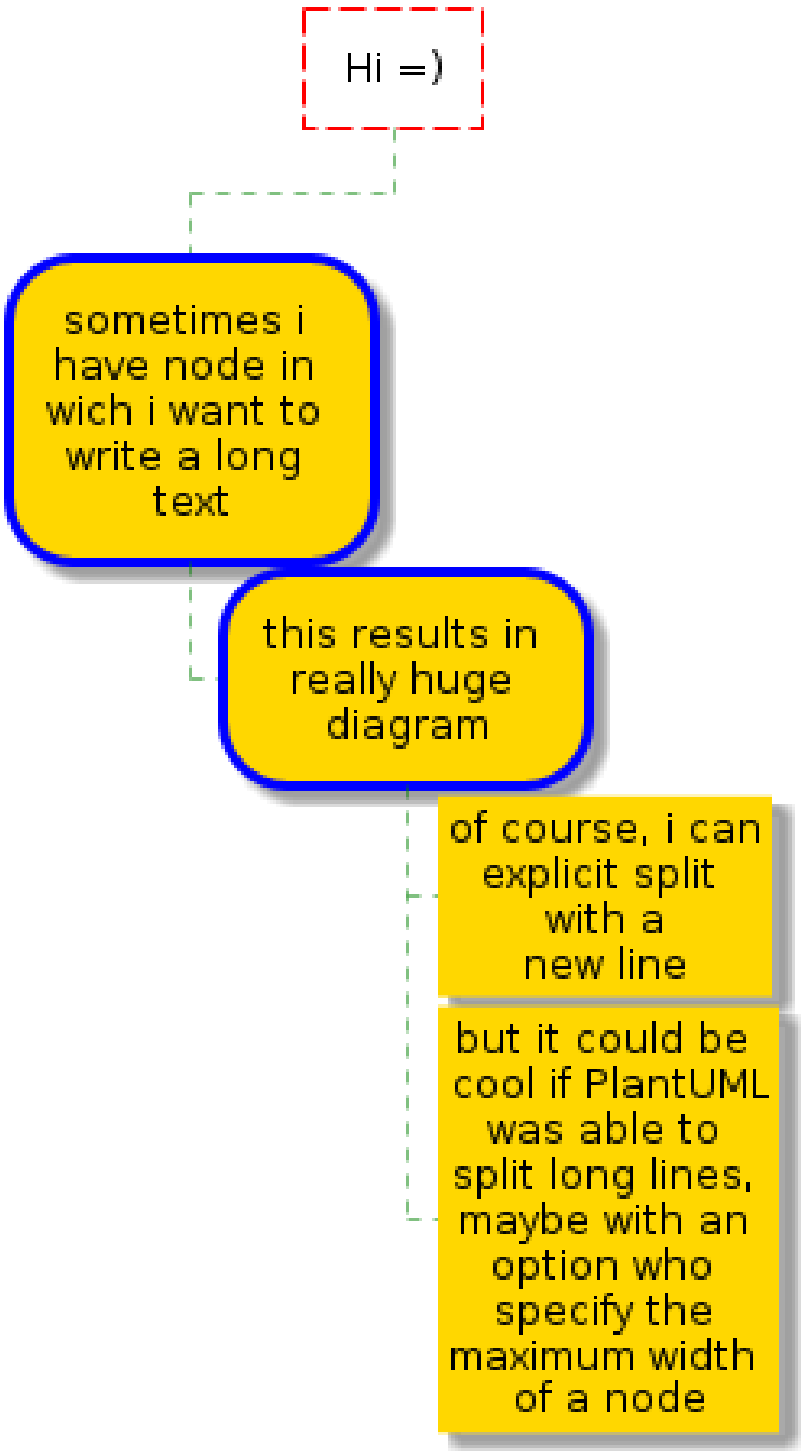



```
@startwbs
<style>
wbsDiagram {
        .pink {
                BackgroundColor pink
        }
        .your_style_name {
                BackgroundColor SkyBlue
        }
}
</style>
+ this is the partner workpackage <<your_style_name>>
++ this is my workpackage <<pink>>
++ this is another workpackage
@endwbs
```



```
@startwbs


<style>
node {
        Padding 12
        Margin 3
        HorizontalAlignment center
        LineColor blue
        LineThickness 3.0
        BackgroundColor gold
        RoundCorner 40
        MaximumWidth 100
}

rootNode {
        LineStyle 8.0;3.0
        LineColor red
        BackgroundColor white
        LineThickness 1.0
        RoundCorner 0
        Shadowing 0.0
}

leafNode {
        LineColor gold
        RoundCorner 0
        Padding 3
}

arrow {
        LineStyle 4
        LineThickness 0.5
        LineColor green
}
</style>

+ Hi =)
++ sometimes i have node in wich i want to write a long text
+++ this results in really huge diagram
```

```
++++ of course, i can explicit split with a\nnew line
++++ but it could be cool if PlantUML was able to split long lines, maybe with an opti

@endwbs
```

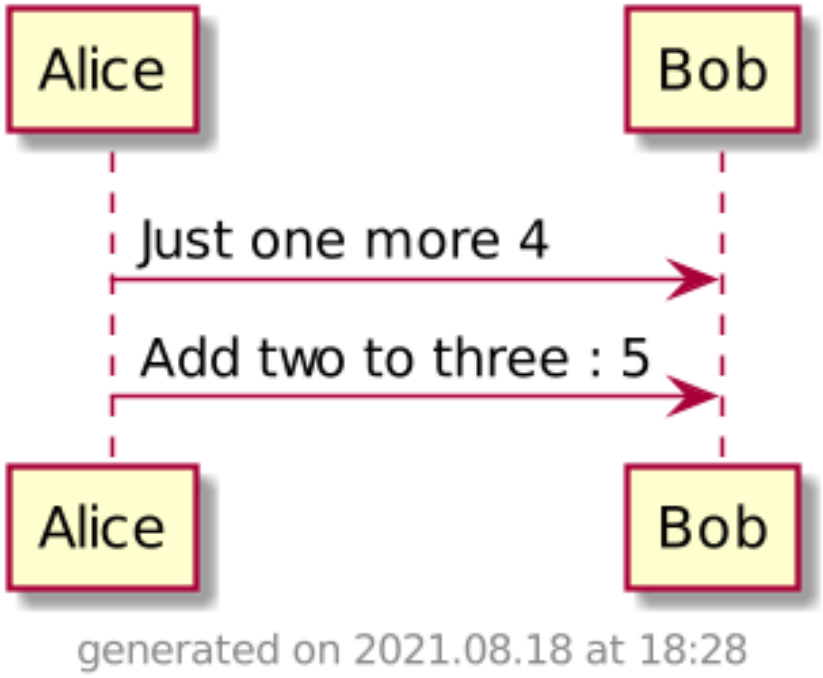

Hi =)
sometimes i have node in wich i want to write a long text
this results in really huge diagram
of course, i can explicit split with a new line
but it could be cool if PlantUML was able to split long lines, maybe with an option who specify the maximum width of a node

5 Preprocessing

https://plantuml.com/preprocessing

The simple preprocessor allows the definition of variables and functions. Some standard functions like %date are already provided.

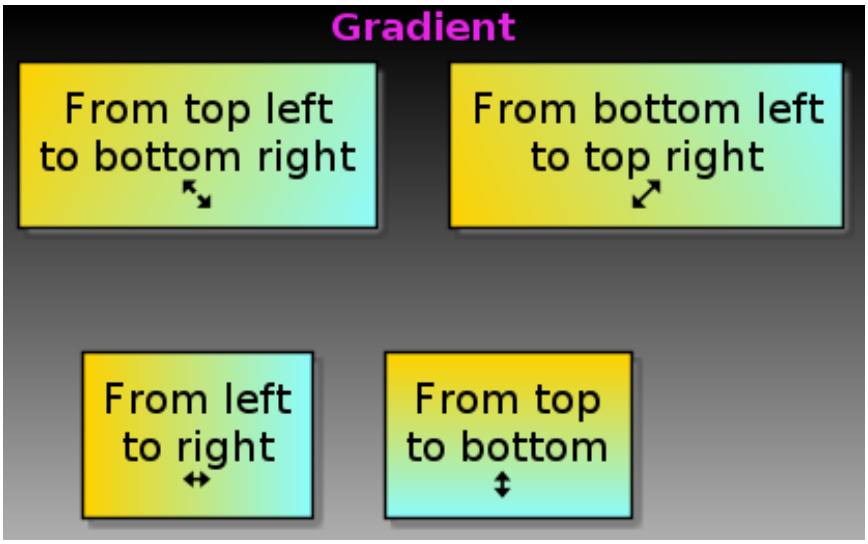
```
@startuml
scale 1.5
!function $inc($value, $step=1)
!return $value + $step
!endfunction

Alice -> Bob : Just one more $inc(3)
Alice -> Bob : Add two to three : $inc(3, 2)

center footer generated on %date("yyyy.MM.dd 'at' HH:mm")
@enduml
```



6 skinparam

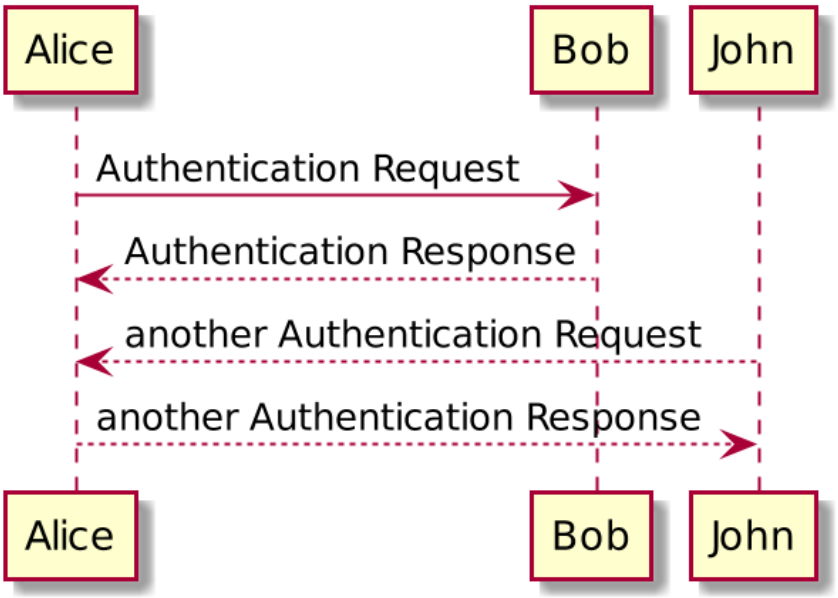
6.1 Gradients

Minimally adapted from `https://blog.jdriven.com/2017/10/plantuml-pleasantness-use-gradier`



7 Scaling

```
@startuml
scale 2

Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
John --> Alice: another Authentication Request
Alice --> John: another Authentication Response
@enduml
```



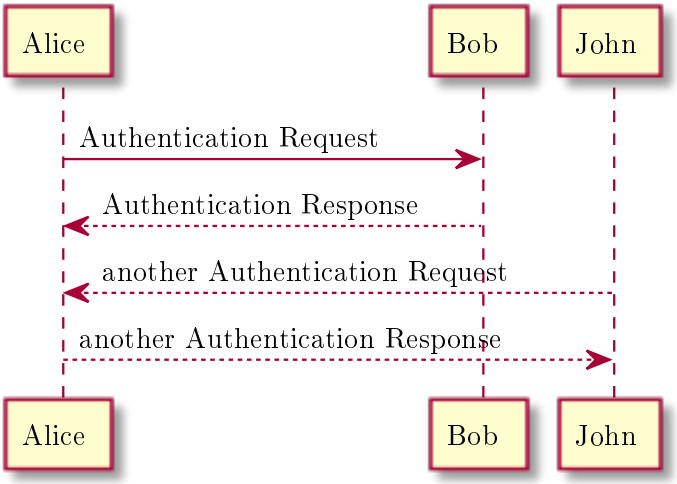
8 TODO using SVG graphics

The *svg* package uses inkscape to separate the text and graphical elements of the SVG into a Tex file (*.pdf$_{tex}$) and a PDF file containig the graph elements. E.g. svg-sequence1.svg into svg-sequence1.pdf$_{tex}$ and svg-sequence1.pdf.

Currently, SVG pictures can only be rendered correctly, **if the picture is in the same directory** as the tex source file (and therefore also the org source file).

Note: with the current org version 9.1.14 and Emacs 26.1 the SVG is not correctly displayed in the org buffer, but the SVG renders fine in the exported Latex PDF.

```
@startuml
' this is a comment
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
John --> Alice: another Authentication Request
Alice --> John: another Authentication Response
@enduml
```


Alice
Bob
John
Authentication Request
Authentication Response
another Authentication Request
another Authentication Response
Alice
Bob
John