

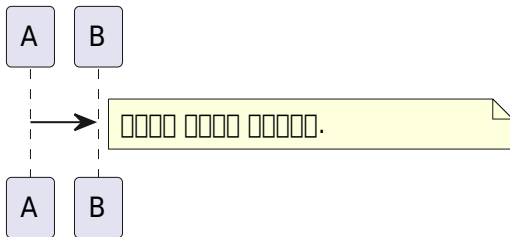
PlantUML

Editor

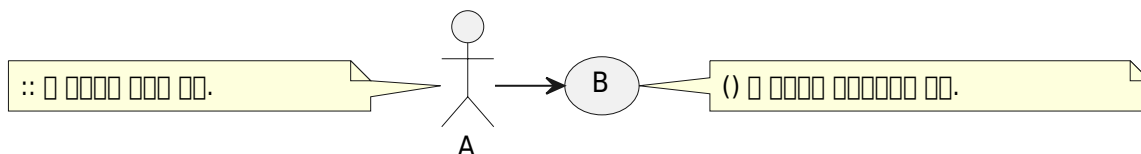
- <https://plantuml-editor.kkeisuke.com/>

plantuml

```
<uml>
A -> B
note right of A : 가 .
</uml>
```

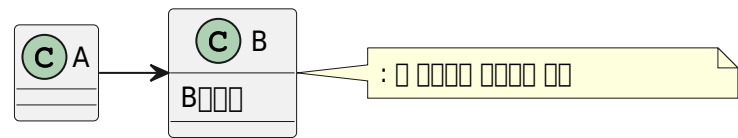


```
<uml>
:A: -> (B)
note left of A : ::
note right of B : ()
</uml>
```

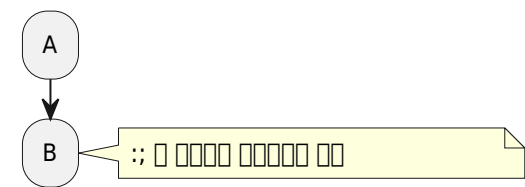


```
<uml>
A -> B
B : B
note right of B : :
```

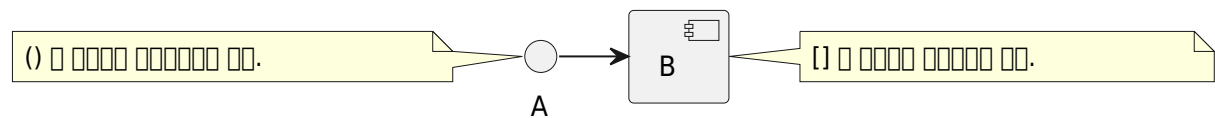
</uml>



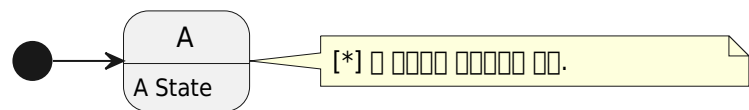
```
<uml>
:A;
:B;
note right : ;;
</uml>
```



```
<uml>
()A -> [B]
note left of A : ()
note right of B : []
</uml>
```



```
<uml>
[*] -> A
A : A State
note right of A : [*]
</uml>
```



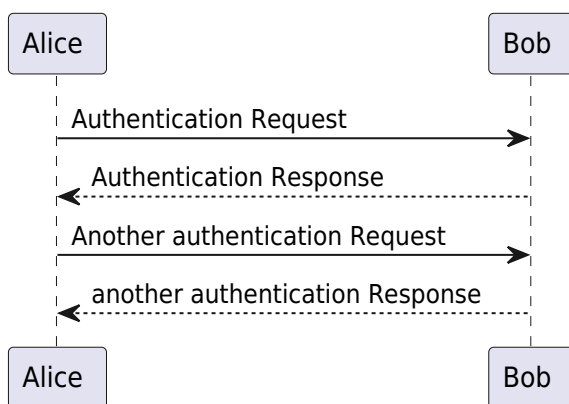
Sequence

Basic examples

- ->
- -->

```
<uml>
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

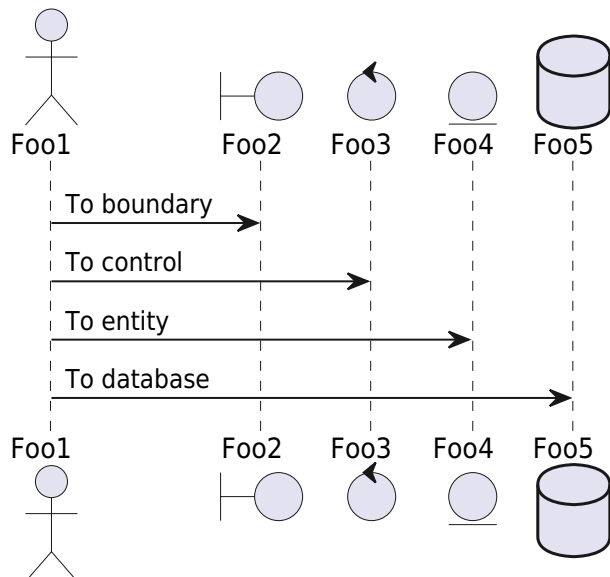
Alice -> Bob: Another authentication Request
Alice <-- Bob: another authentication Response
</uml>
```



- : '
- : '/' '/'

Declaring participant

```
actor Foo1
boundary Foo2
control Foo3
entity Foo4
database Foo5
Foo1 -> Foo2 : To boundary
Foo1 -> Foo3 : To control
Foo1 -> Foo4 : To entity
Foo1 -> Foo5 : To database
```

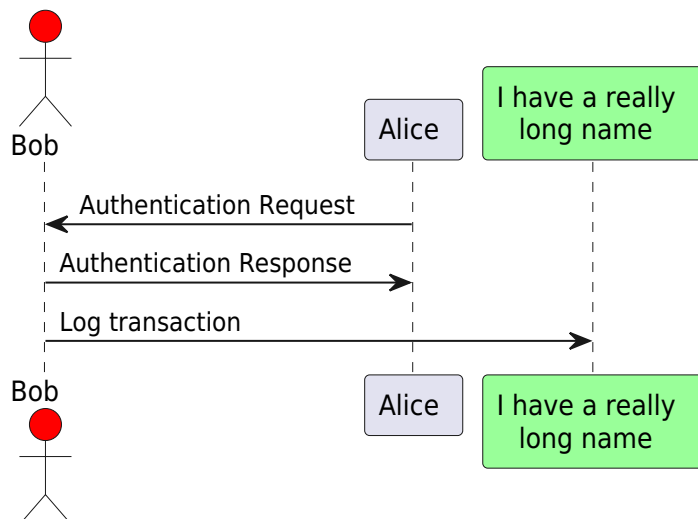


```

<uml>
actor Bob #red
' The only difference between actor
'and participant is the drawing
participant Alice
participant "I have a really\nlong name" as L #99FF99
/' You can also declare:
    participant L as "I have a really\nlong name"  #99FF99
  '/
  
```

```

Alice->>Bob: Authentication Request
Bob->>Alice: Authentication Response
Bob->>L: Log transaction
</uml>
  
```



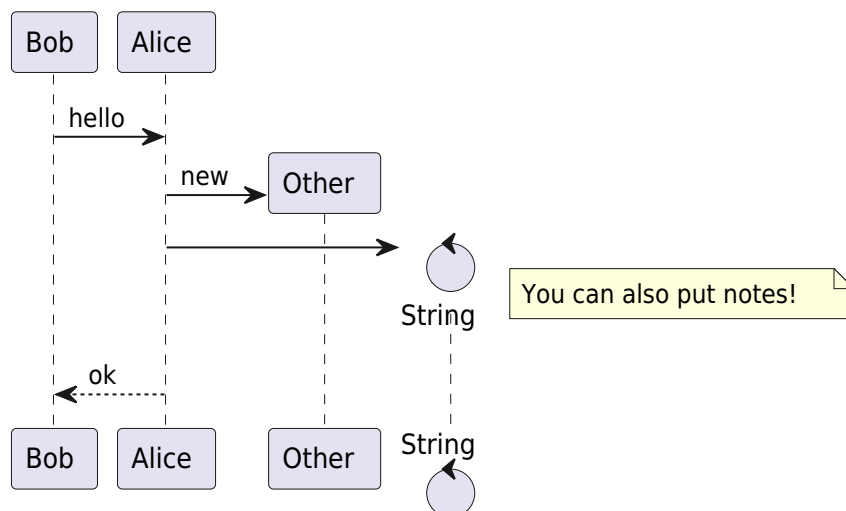
Participant creation

```
<uml>
Bob -> Alice : hello

create Other
Alice -> Other : new

create control String
Alice -> String
note right : You can also put notes!

Alice --> Bob : ok
</uml>
```

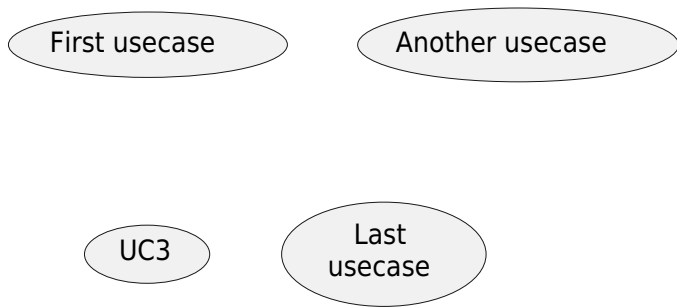


Use Case Diagram

Usecases

- ()
- usecase

```
<uml>
(First usecase)
(Another usecase) as (UC2)
usecase UC3
usecase (Last\nusecase) as UC4
</uml>
```

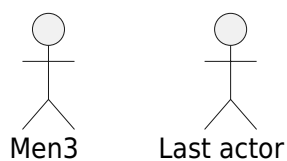
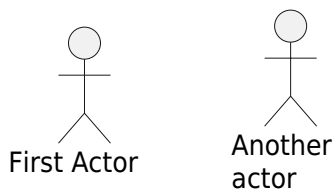


Actors

- actor

```

<uml>
:First Actor:
:Another\nactor: as Men2
actor Men3
actor :Last actor: as Men4
</uml>
  
```

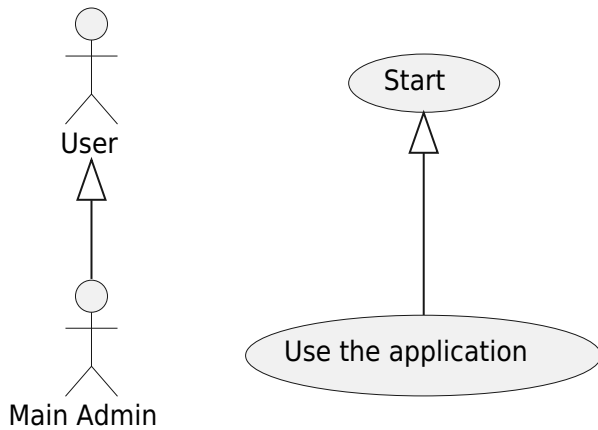


- <|--

```

<uml>
:Main Admin: as Admin
(Use the application) as (Use)

User <|-- Admin
(Start) <|-- (Use)
</uml>
  
```



- note left of
 - note right of
 - note top of
 - note bottom
 - .. :

```

<uml>
:Main Admin: as Admin
(Use the application) as (Use)

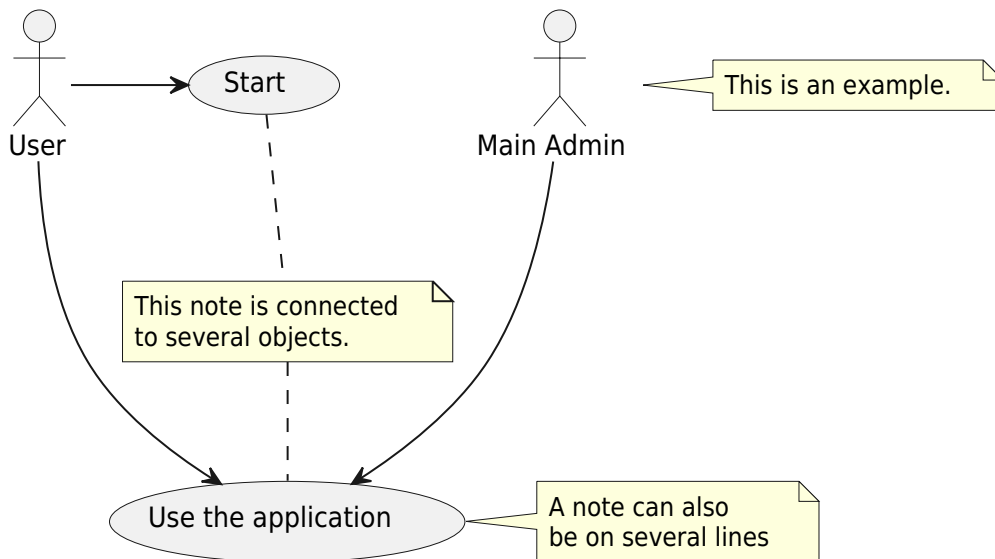
User -> (Start)
User --> (Use)

Admin ---> (Use)

note right of Admin : This is an example.

note right of (Use)
  A note can also
  be on several lines
end note

note "This note is connected\nto several objects." as N2
(Start) .. N2
N2 .. (Use)
</uml>
  
```

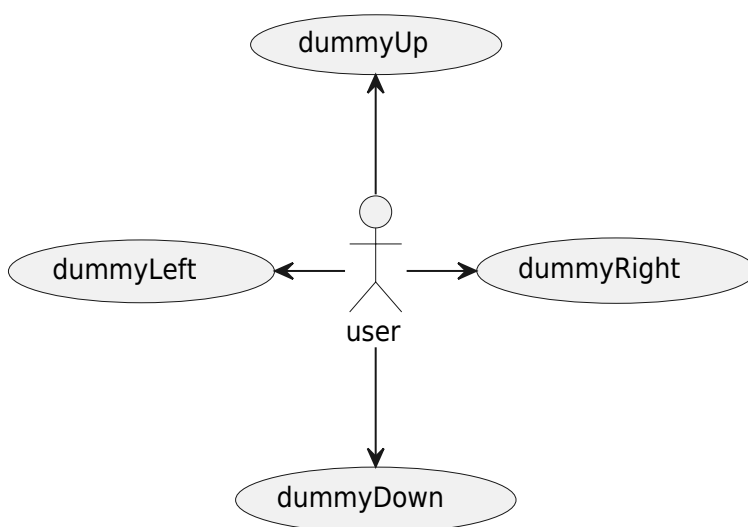


- -- :
- - :
- .. :
- left, right, up or down

```

<uml>
:user: -left-> (dummyLeft)
:user: -right-> (dummyRight)
:user: -up-> (dummyUp)
:user: -down-> (dummyDown)
</uml>

```

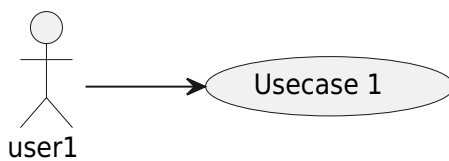
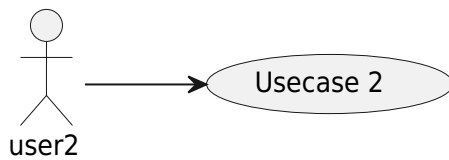


- : top to bottom
- : top to bottom


```

<uml>
left to right direction
user1 --> (Usecase 1)
user2 --> (Usecase 2)
</uml>

```

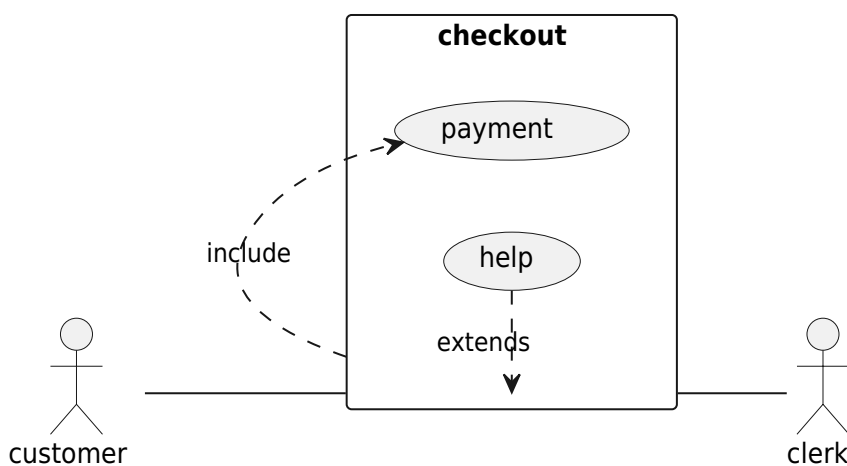


Complete example

```

<uml>
left to right direction
skinparam packageStyle rect
actor customer
actor clerk
rectangle checkout {
  customer -- (checkout)
  (checkout) .> (payment) : include
  (help) .> (checkout) : extends
  (checkout) -- clerk
}
</uml>

```



Class Diagram

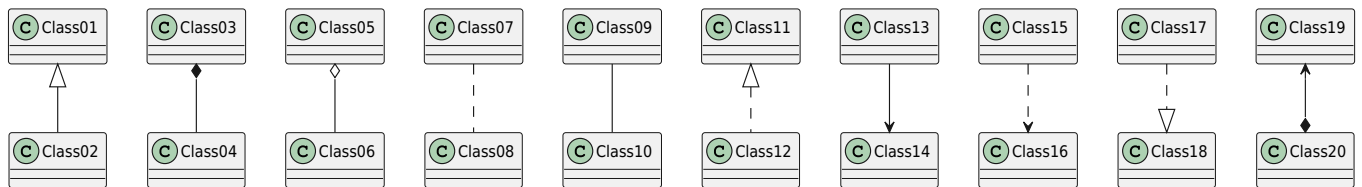
Relations between classes

Extension	< --
Composition	*--
Agregation	o--

```

<uml>
scale 900 width
Class01 <|-- Class02
Class03 *-- Class04
Class05 o-- Class06
Class07 .. Class08
Class09 -- Class10
Class11 <|.. Class12
Class13 --> Class14
Class15 ..> Class16
Class17 ..|> Class18
Class19 <--* Class20
</uml>

```



Label on relations

- cardinality : ""
- label : :

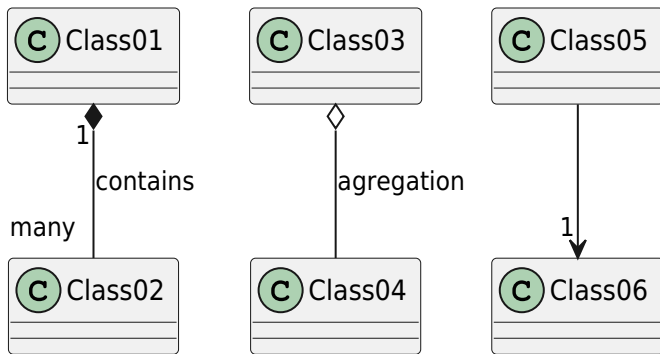
```

<uml>
Class01 "1" *-- "many" Class02 : contains

Class03 o-- Class04 : agregation

Class05 --> "1" Class06
</uml>

```



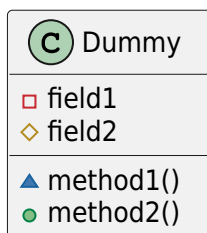
Defining visibility

Character	Visibility
-	private
#	protected
~	package private
+	public

```

<uml>
class Dummy {
  -field1
  #field2
  ~method1()
  +method2()
}
</uml>

```




Abstract and Static

- {abstract}
- {static} or {classifier}

```

<uml>
class Dummy {
  {static} String id
  {abstract} void methods()
}
</uml>

```

 Dummy
String id
<i>void methods()</i>

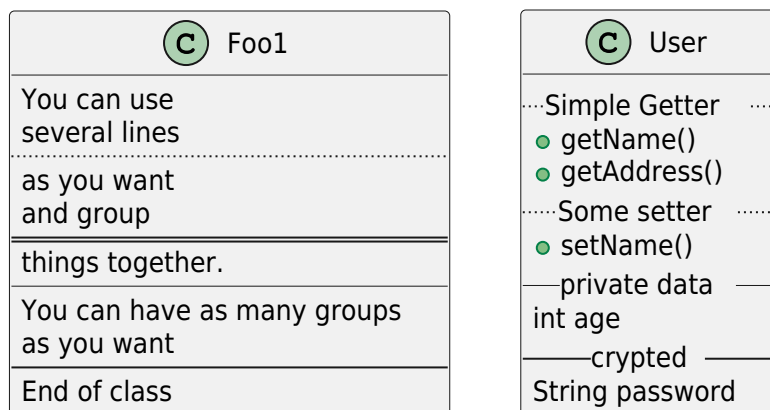
Advanced class body

- --
- ..
- ==
- __

```
<uml>
class Foo1 {
    You can use
    several lines
    ..
    as you want
    and group
    ==
    things together.

    __
    You can have as many groups
    as you want
    --
    End of class
}

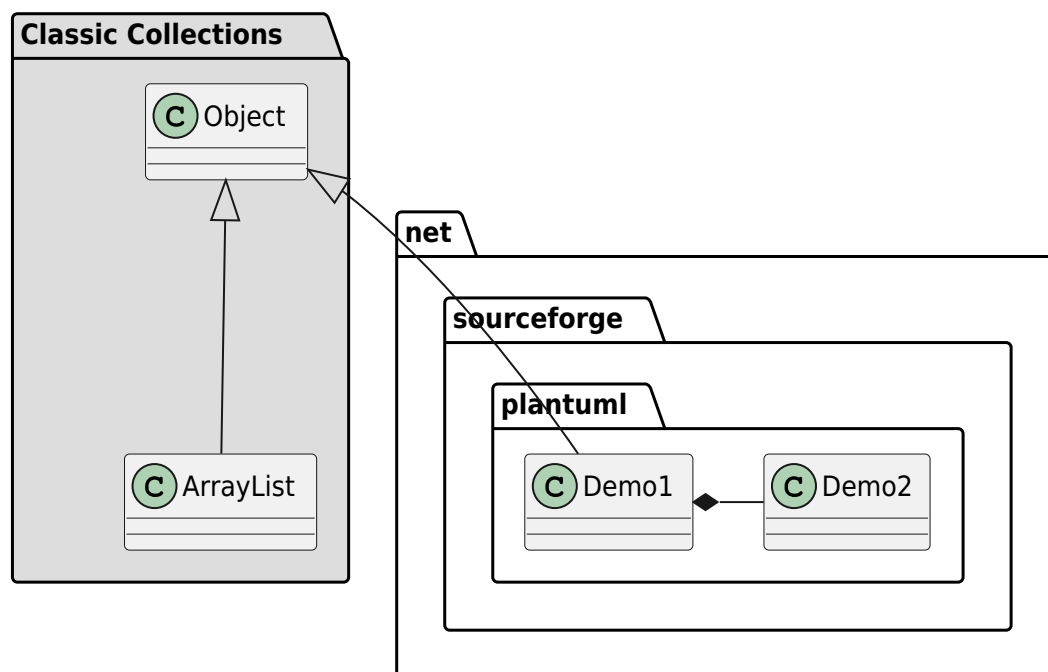
class User {
    .. Simple Getter ..
    + getName()
    + getAddress()
    .. Some setter ..
    + setName()
    __ private data __
    int age
    -- crypted --
    String password
}
</uml>
```



Packages

```
<uml>
package "Classic Collections" #DDDDDD {
    Object <|-- ArrayList
}

package net.sourceforge.plantuml {
    Object <|-- Demo1
    Demo1 *- Demo2
}
</uml>
```



Splitting large files

- page (hpages)x(vpages)

```

<uml>
' Split into 4 pages
page 2x2

class BaseClass

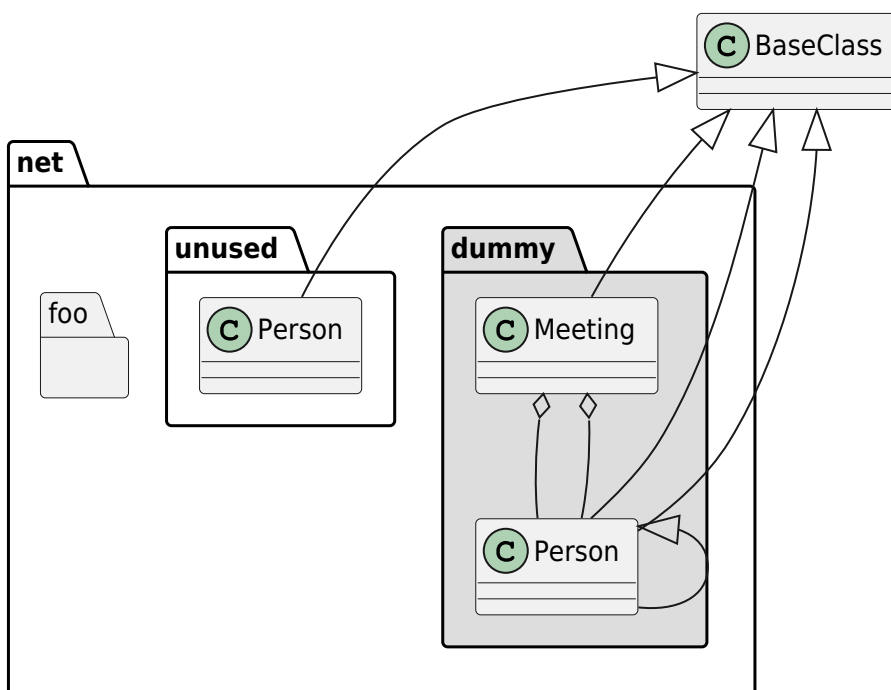
namespace net.dummy #DDDDDD {
    .BaseClass <|-- Person
    Meeting o-- Person
    .BaseClass <|-- Meeting
}

namespace net.foo {
    net.dummy.Person <|-- Person
    .BaseClass <|-- Person

    net.dummy.Meeting o-- Person
}

BaseClass <|-- net.unused.Person
</uml>

```



Activity

```

<uml>
title Servlet Container

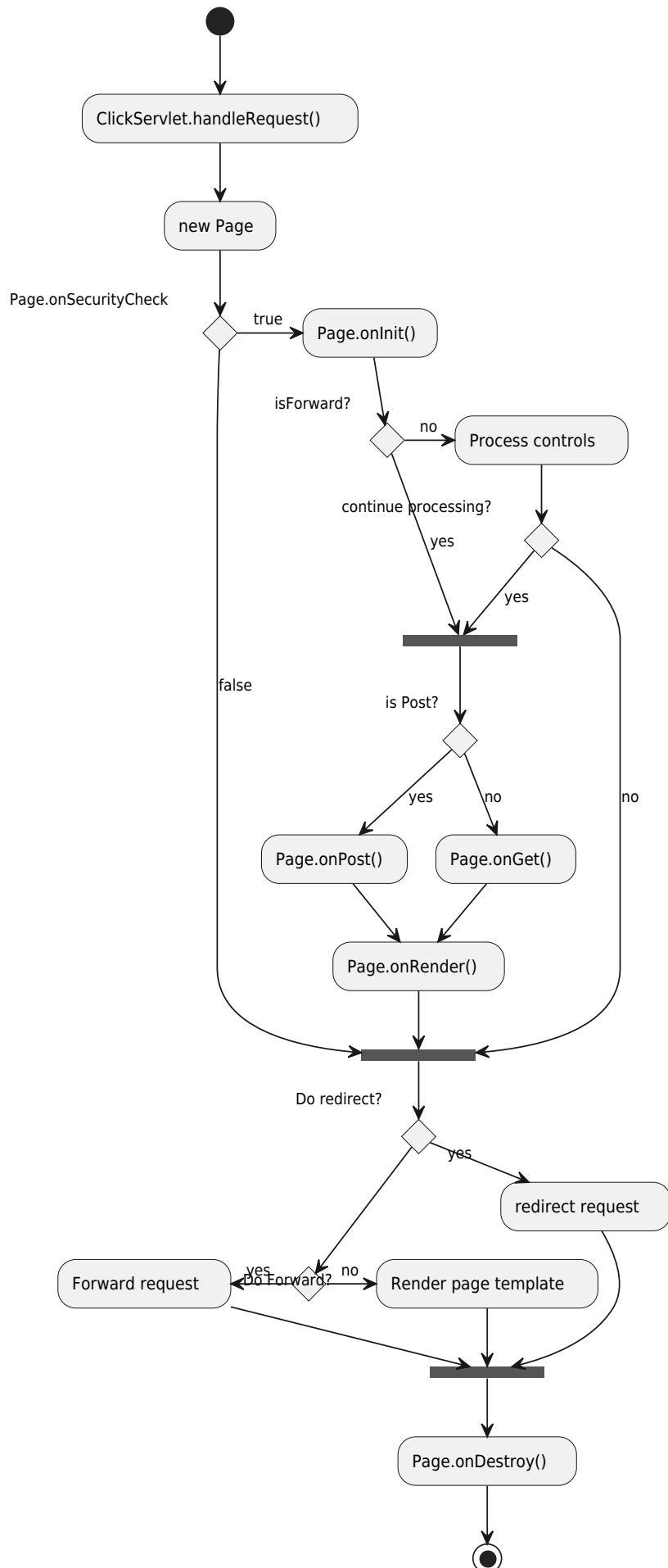
```

```
(*) --> "ClickServlet.handleRequest()"
--> "new Page"

if "Page.onSecurityCheck" then
  ->[true] "Page.onInit()"
  if "isForward?" then
    ->[no] "Process controls"
    if "continue processing?" then
      -->[yes] ===RENDERING===
    else
      -->[no] ===REDIRECT_CHECK===
    endif
  else
    -->[yes] ===RENDERING===
  endif
  if "is Post?" then
    -->[yes] "Page.onPost()"
    --> "Page.onRender()" as render
    --> ===REDIRECT_CHECK===
  else
    -->[no] "Page.onGet()"
    --> render
  endif
endif
else
  -->[false] ===REDIRECT_CHECK===
endif

if "Do redirect?" then
  ->[yes] "redirect request"
  --> ==BEFORE_DESTROY==
else
  if "Do Forward?" then
    -left->[yes] "Forward request"
    --> ==BEFORE_DESTROY==
  else
    -right->[no] "Render page template"
    --> ==BEFORE_DESTROY==
  endif
endif

--> "Page.onDestroy()"
-->(*)
</uml>
```

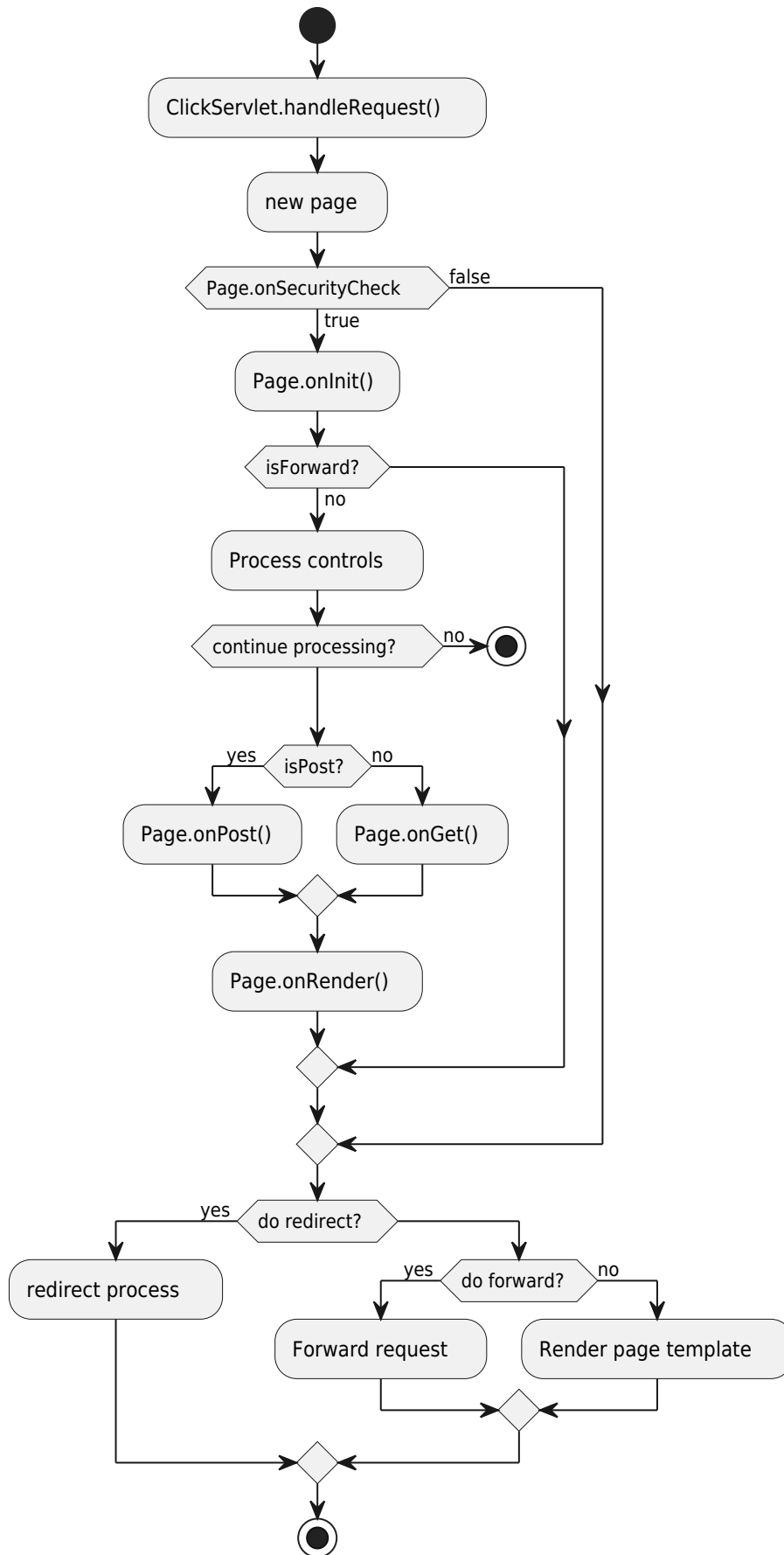
Servlet Container

Activity Beta

```
<uml>
start
:ClickServlet.handleRequest();
:new page;
if (Page.onSecurityCheck) then (true)
  :Page.onInit();
  if (isForward?) then (no)
    :Process controls;
    if (continue processing?) then (no)
      stop
    endif
  if (isPost?) then (yes)
    :Page.onPost();
  else (no)
    :Page.onGet();
  endif
  :Page.onRender();
endif
else (false)
endif

if (do redirect?) then (yes)
  :redirect process;
else
  if (do forward?) then (yes)
    :Forward request;
  else (no)
    :Render page template;
  endif
endif

stop
</uml>
```



Component Diagram

State

Obejct

Gantt Diagram

Project starts the 20th of september 2018

saturday are closed

sunday are closed

[Prototype design] lasts 13 days and is colored in Lavender/LightBlue

[Test prototype] lasts 9 days and is colored in Coral/Green and starts 3 days after [Prototype design]'s end

[Write tests] lasts 5 days and ends at [Prototype design]'s end

[Hire tests writers] lasts 6 days and ends at [Write tests]'s start

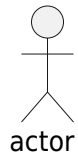
[Init and write tests report] is colored in Coral/Green

[Init and write tests report] starts 1 day before [Test prototype]'s start and ends at [Test prototype]'s end

deployment diagram

- <http://plantuml.com/deployment-diagram>

```
actor actor
agent agent
artifact artifact
boundary boundary
card card
cloud cloud
component component
control control
database database
entity entity
file file
folder folder
frame frame
interface interface
node node
package package
queue queue
stack stack
rectangle rectangle
storage storage
usecase usecase
```



actor

agent

artifact

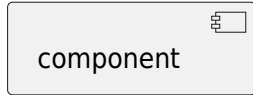


boundary

card



cloud



component



control



database



entity



file



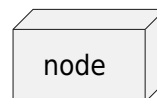
folder



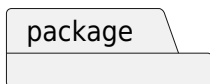
frame



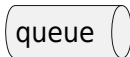
interface



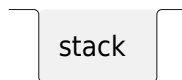
node



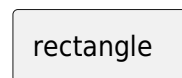
package



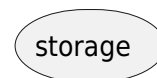
queue



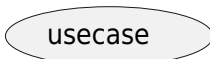
stack



rectangle



storage



usecase

- [OPEN](#)
- [OPEN](#)

From:

<http://jace.link/> - **Various Ways**

Permanent link:

<http://jace.link/open/plantuml>Last update: **2021/11/25 06:52**