

[Back to websequencediagrams.com](http://www.websequencediagrams.com)

Simple signals

Draw a signal from one participant to another like this:

Alice->Bob: Authentication Request
 Bob-->Alice: Authentication Response

The participants are automatically created when they are used. Use the "-->" syntax to draw a dotted line.

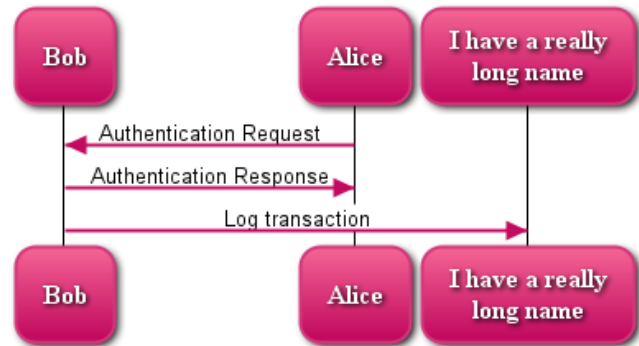


Changing the order of participants

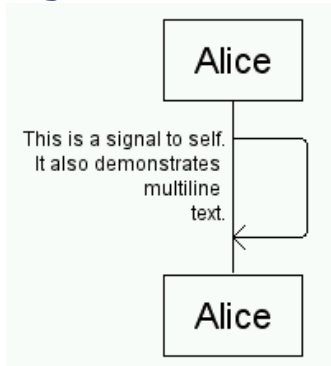
If you want to participants to be shown in a different order than they are used, declare them first using the participant keyword. You can also rename them this way to save typing.

participant Bob
 participant Alice
 participant "I have a really\nlong name" as L

Alice->Bob: Authentication Request
 Bob->Alice: Authentication Response
 Bob->L: Log transaction



Signal to Self



A participant can send a signal to itself. This will result in an arrow that turns back on itself.

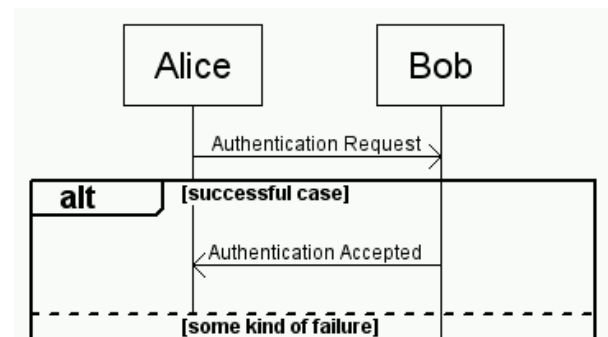
You may break the text into multiple lines by using "\n".

Alice->Alice: This is a signal to self.\nIt also demonstrates \nmultiline \ntext.

Grouping signals together

You can group signals together using the alt/else, opt, and loop keywords. All of them can take a text description that will be displayed in the group header. Use the end keyword to signal the end of a group. The groups may be nested to any depth.

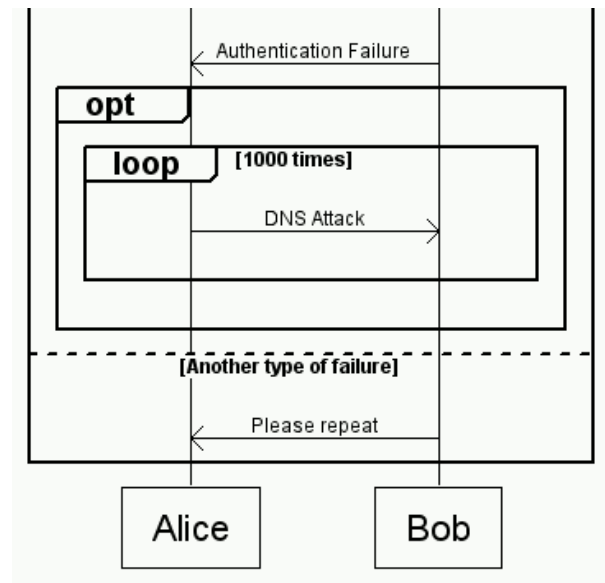
Alice->Bob: Authentication Request
 alt successful case
 Bob->Alice: Authentication Accepted
 else some kind of failure
 Bob->Alice: Authentication Failure
 opt



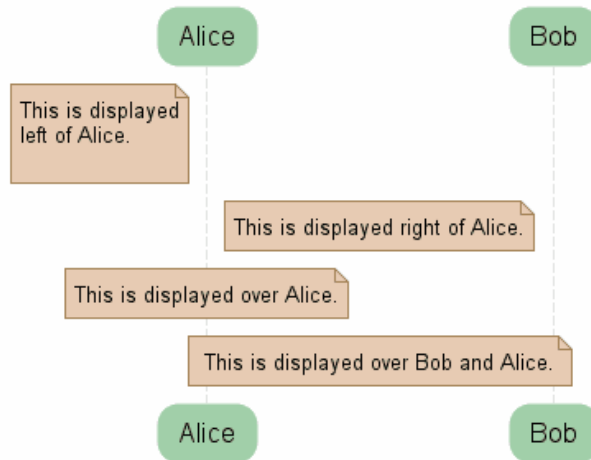
```

loop 1000 times
  Alice->Bob: DNS Attack
end
else Another type of failure
  Bob->Alice: Please repeat
end

```



Notes in the diagram



note over Alice: This is displayed over Alice.
 note over Alice, Bob: This is displayed over Bob and Alice.

You can add notes to your diagram. Notes can be placed to the left of a participant or to the right of a participant. In addition, you can centre a note over one or more participants.

If a note contains more than one line, it will be not be word-wrapped. Instead, it will be formatted exactly as written.

participant Alice
 participant Bob

note left of Alice
 This is displayed
 left of Alice.
 end note

note right of Alice: This is displayed right of Alice.

Lifeline Activation and Destruction

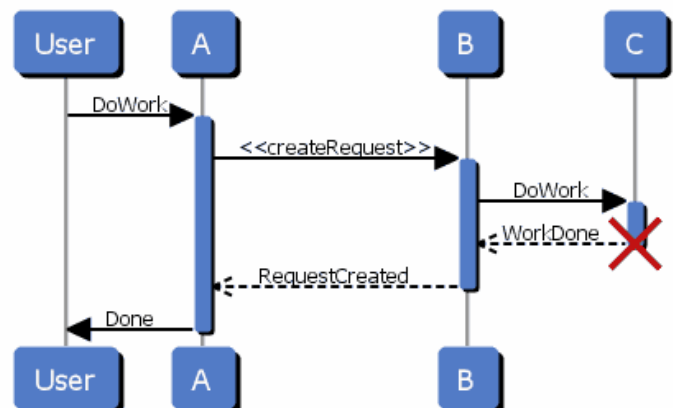
Use the activate and deactivate keywords to denote object activation. While activated, the participant's lifeline will be highlighted. The activate/deactivate keywords will apply to the previous signal.

You can use the destroy keyword to destroy a participant. The participant's lifeline will end at the previous signal.

```

User->>A: DoWork
activate A
A->>B: <<createRequest>>
activate B
B->>C: DoWork

```



```
activate C
C-->B: WorkDone
destroy C
B-->A: RequestCreated
deactivate B
A->User: Done
```