



PlantUML с нуля до гуру

Учимся «кодить» sequence-диаграммы



Содержание

1

PlantUML

Как инструмент

2

Практика

Готовим вместе

3

Лайфхаки

С живыми примерами

4

Агрегация

Для командной работы

Обо мне

- Лид системного анализа в SberDevices
- Веду канал и авторский курс [@Analyst_Boost](#)
- Преподаю в онлайн-школе
- Иногда побеждаю в хакатонах
- Люблю конференции :)





1

PlantUML

Как инструмент

“UML is the worst modeling language
...”

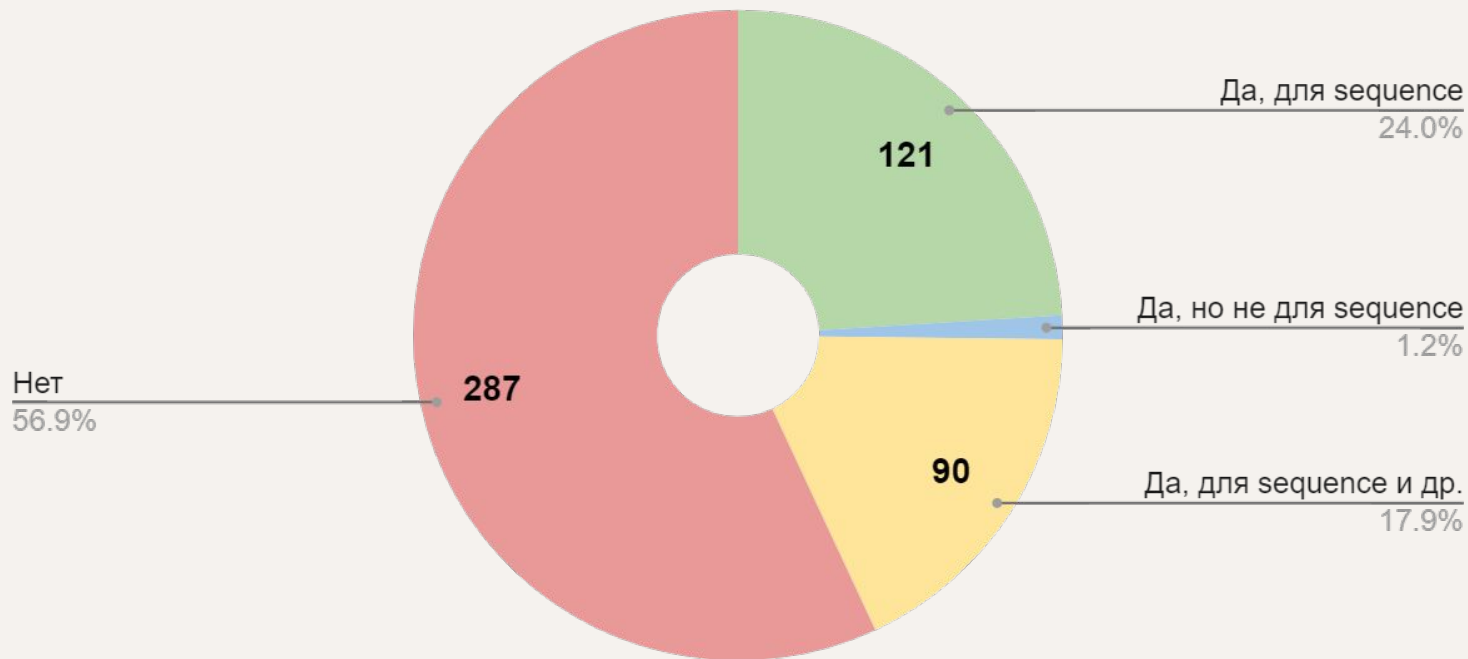
—Bran Selic, coauthor UML

“UML is the worst modeling language
except for all the others”

—Bran Selic, coauthor UML

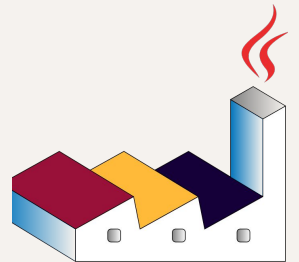
Используете ли вы PlantUML в своей работе? Если да, то для каких диаграмм?

Опрошено аналитиков: 504. Актуально на 25.11.22



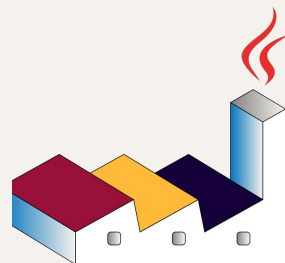
Что такое PlantUML

- Diagrams-as-a-Code инструмент
- Доступен и онлайн, и локально



Что такое PlantUML

- Diagrams-as-a-Code инструмент
- Доступен и онлайн, и локально
- Генерирует картинки в LaTeX, PNG и SVG
- Встраивается в другие системы (например, confluence)



Что поддерживает PlantUML

UML диаграммы:

- **Последовательности**
- Исползования
- Классов
- Объектов
- Деятельности
- Компонентов
- Развёртывания
- Состояний
- Синхронизации

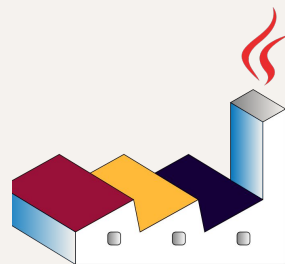
не-UML диаграммы:

- Entity Relationship Диаграммы
- JSON и YAML
- Wireframe GUI or UI mockups (salt)
- Диаграммы ArchiMate
- Диаграммы сети (nwdiag)
- SDL и Ditaа
- Work Breakdown Structure (WBS)
- Gantt и MindMap
- AsciiMath или JLaTeXMath



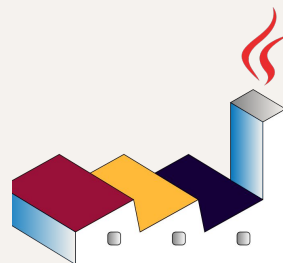
Преимущества PlantUML

- Интуитивно понятный “язык”
- Свобода в кастомизации отображения



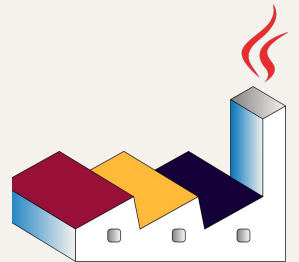
Преимущества PlantUML

- Интуитивно понятный “язык”
- Свобода в кастомизации отображения
- Отдельный файл единого стиля
- Автонумерация шагов и фиксированные расстояния
- Экспорт в svg с поддержкой гиперссылок



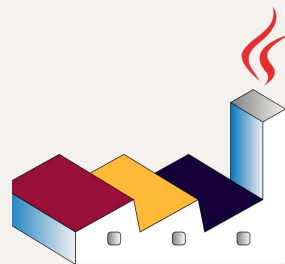
Преимущества PlantUML

- Интуитивно понятный “язык”
- Свобода в кастомизации отображения
- Отдельный файл единого стиля
- Автонумерация шагов и фиксированные расстояния
- Экспорт в svg с поддержкой гиперссылок
- Стандартные и самописные функции
- Поддержка git для отслеживания изменений
- Совместная работа над общей диаграммой без коллизий



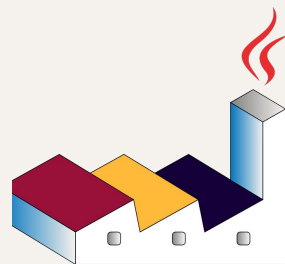
Недостатки PlantUML

- Сложно повлиять на расстановку элементов (см. [dot](#))
- Не все диаграммы целесообразно переносить в PlantUML
- При агрегации файлов alias'ы должны совпадать
- Функция return бывает непредсказуема



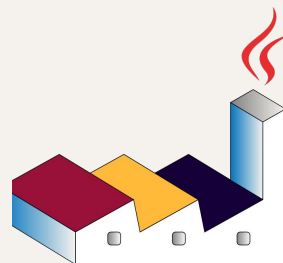
Локально

- Git [скачать](#) || Visual Studio Code [скачать](#). Далее:
 - Ctl + Shift + X (Вкладка View → Extensions): [PlantUML](#) (by jebbs)
 - Скачать: [java](#) || [graphviz](#)
 - github.vscode-pull-request-github + см. [видео](#) и [инструкцию](#)
- IntelliJ IDEA [скачать](#)
- Редактор [Atom](#): file → settings → install: [plantuml-viewer](#) или [plantuml-preview](#)
 - Не все фишки поддерживаются. Например, !function



Онлайн

- <https://www.planttext.com/>
- <http://www.plantuml.com/plantuml/uml/>
- <https://plantuml-editor.kkeisuke.dev/>
- <https://plantuml-editor.kkeisuke.com/>
- прочие возможности: <https://plantuml.com/ru/running>

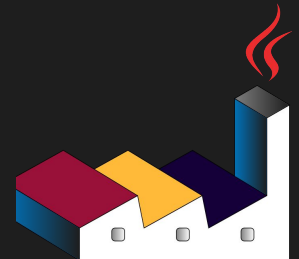
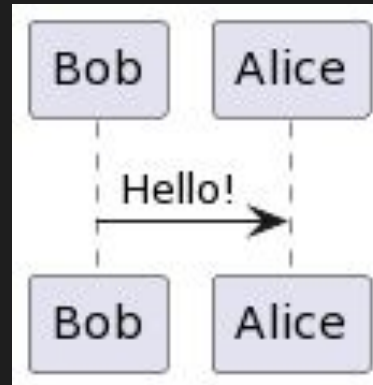


2

Практика

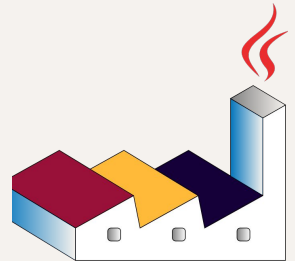
Готовим вместе

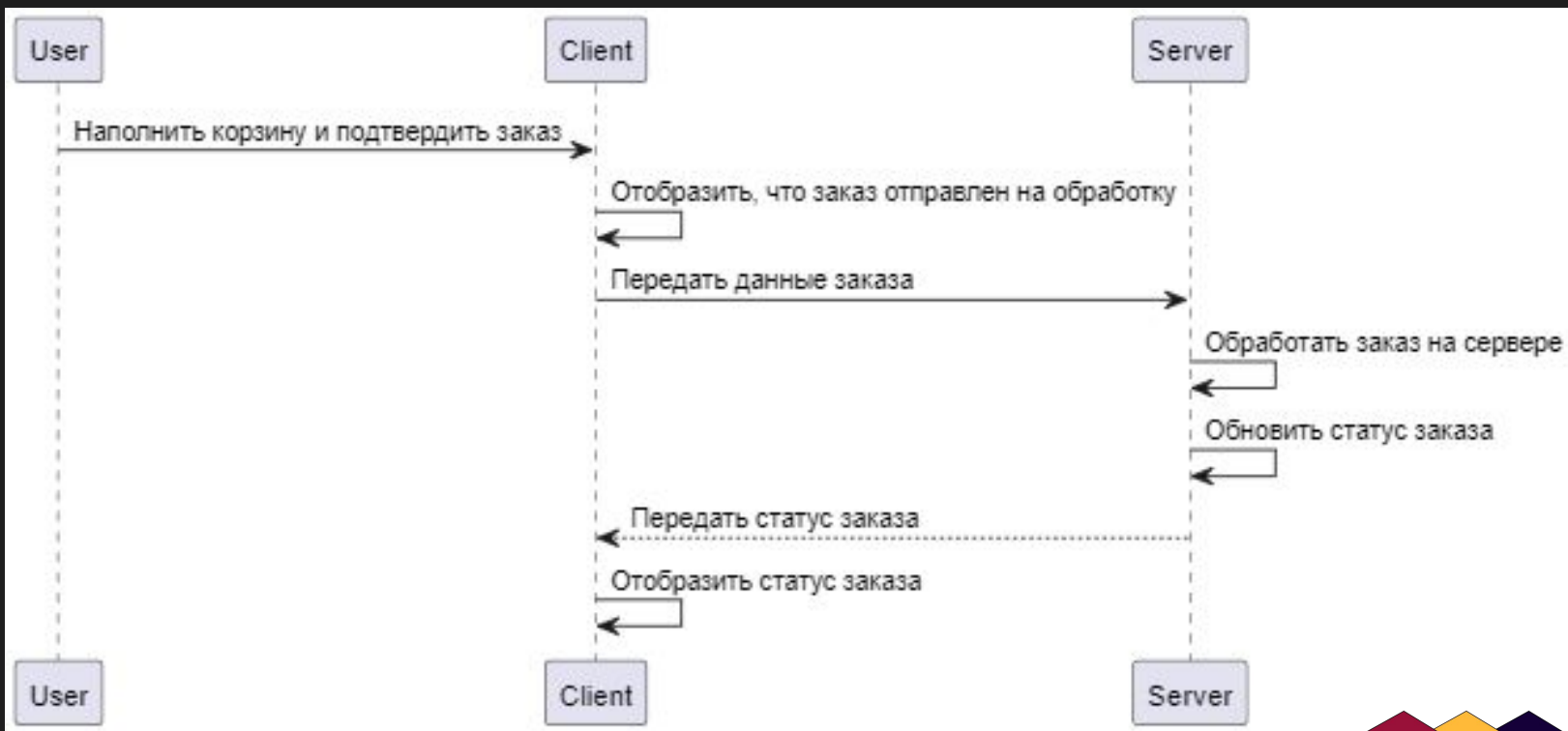
```
@startuml simple_example
Bob -> Alice: Hello!
@enduml
```



Заказ букета. Исходник. 3 минуты

- Компоненты: User Client Server
- Наполнить корзину и подтвердить заказ
- Отобразить, что заказ отправлен на обработку
- Передать данные заказа
- Обработать заказ на сервере
- Обновить статус заказа
- Передать статус заказа
- Отобразить статус заказа

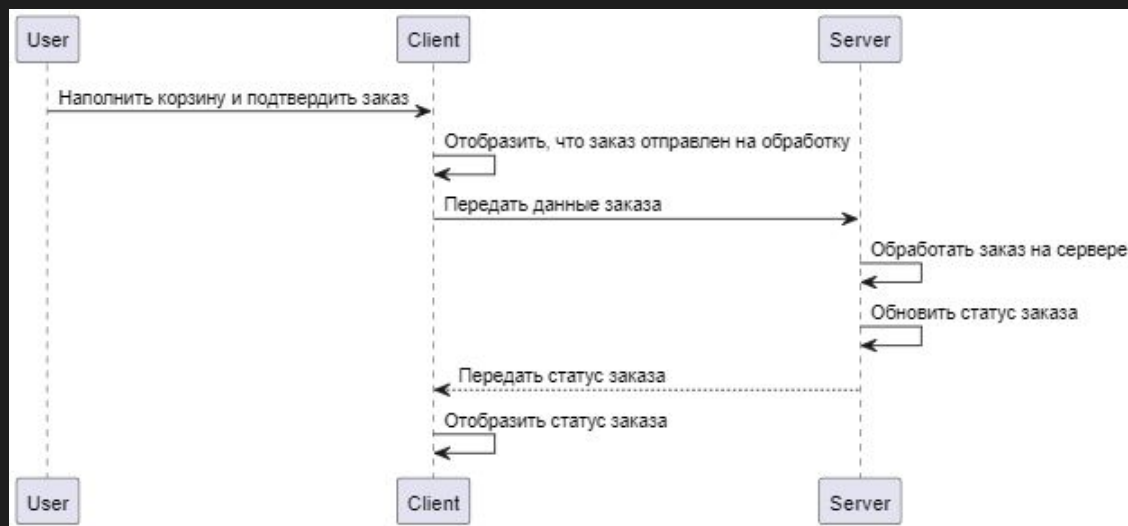




```

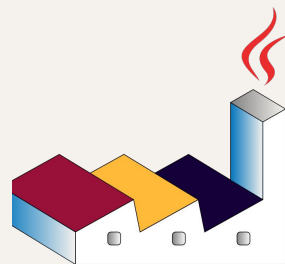
@startuml order
User -> Client: Наполнить корзину и подтвердить заказ
Client -> Client: Отобразить, что заказ отправлен на обработку
Client -> Server: Передать данные заказа
Server -> Server: Обработать заказ на сервере
Server -> Server: Обновить статус заказа
Server --> Client: Передать статус заказа
Client -> Client: Отобразить статус заказа
@enduml

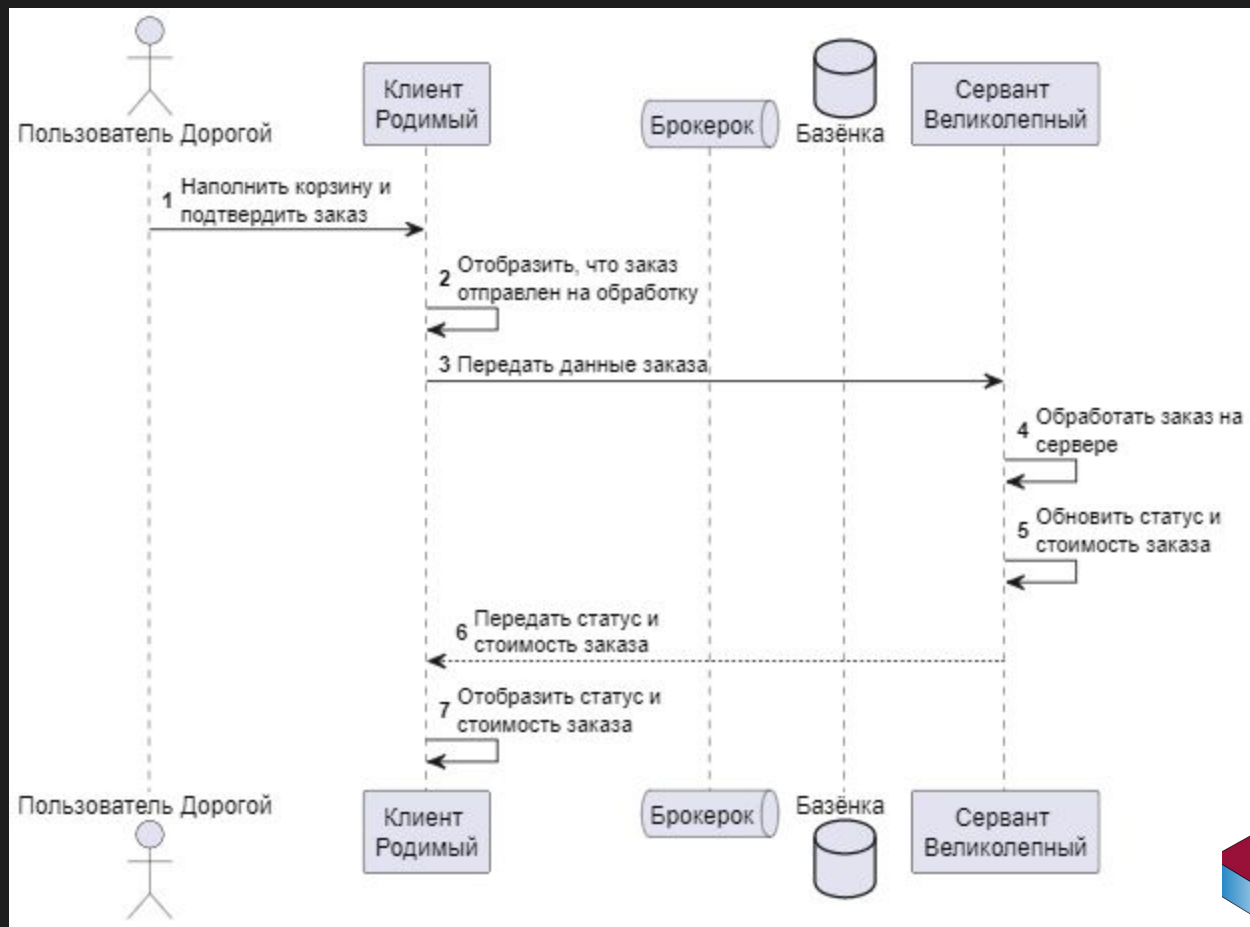
```



Заказ букета. Обновления. 1 минута

- Переименовать Actor'ов:
 - "Пользователь Дорогой"
 - "Клиент Родимый"
 - "Сервант Великолепный"
- Возвращать и **статус, и стоимость заказа**
- Завести и поместить **Брокерок** и **Базёнку** между Клиентом и Сервером
- Пронумеровать, чтоб покомпактнее, и иконки понятные





@startuml

autonumber

skinparam MaxMessageSize 150

Actor "Пользователь Дорогой" as u order 10

participant "Клиент \nРодимый" as c order 20

participant "Сервант \nВеликолепный" as s order 30

queue "Брокерок" as br order 25

database "Базёнка" as db order 27

u -> c: Наполнить корзину и подтвердить заказ

c -> c: Отобразить, что заказ отправлен на обработку

c -> s: Передать данные заказа

s -> s: Обработать заказ на сервере

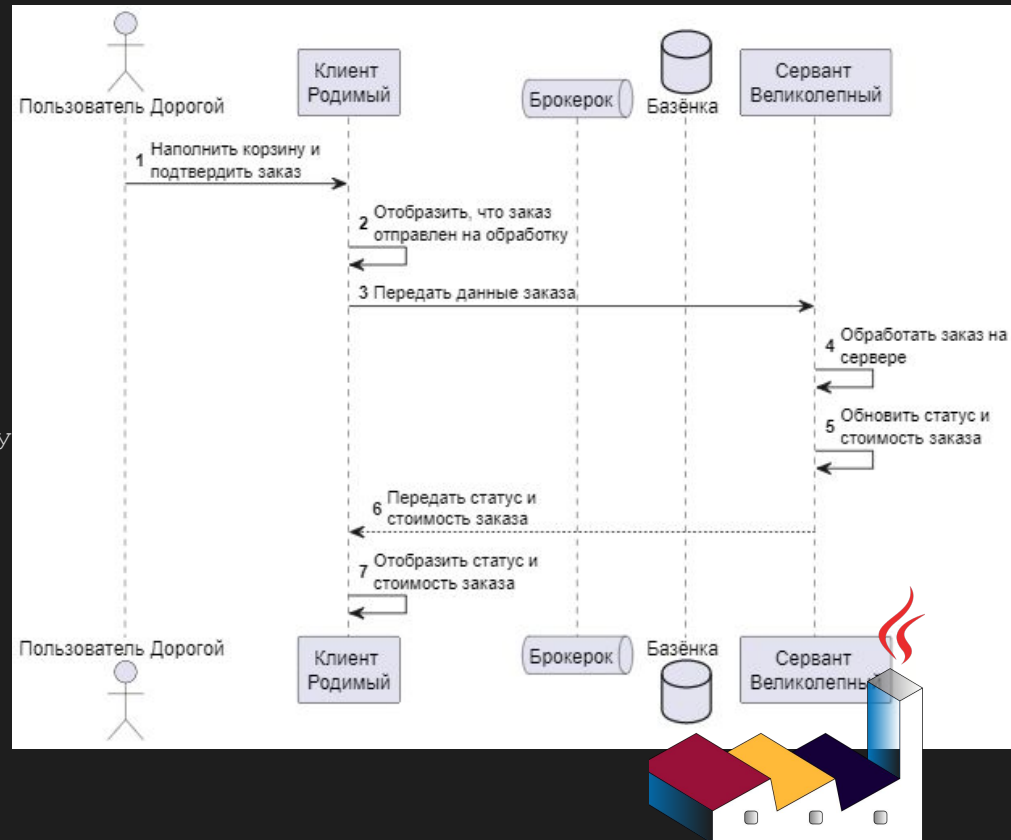
!\$st = "статус и стоимость заказа"

s -> s: Обновить \$st

s --> c: Передать \$st

c -> c: Отобразить \$st

@enduml



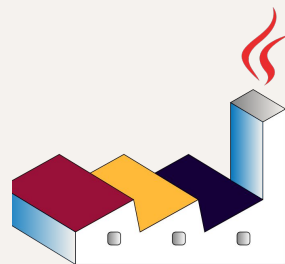
3

Лайфхаки

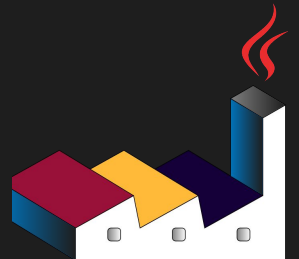
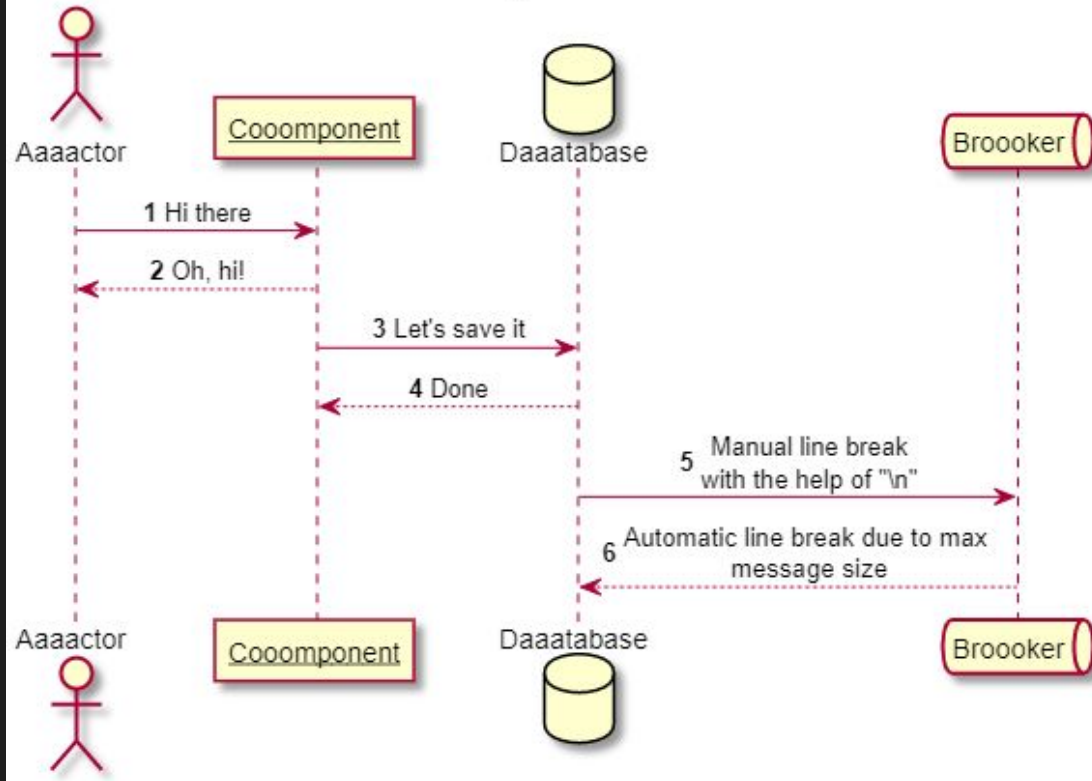
С живыми примерами

Лайфхаки VS Code

- Выбираем тип файла на экспорт:
 - Курсор должен стоять между @startuml и @enduml
 - Command Palette (F1 или Ctrl+Shift+P), команда: Export Current Diagram
 - Предлагаются, но не всегда работают, такие форматы:
 - PNG, SVG, EPS, PDF, VDX, XMI, SCXML, HTML, TXT, UTX, LaTeX
- Preview Diagram: комбинация **Alt + D** (MacOS: **option + D**)

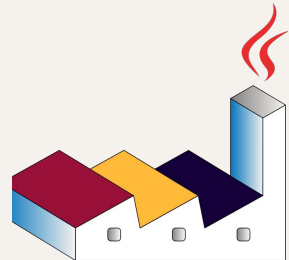


Your diagram title here



Skinparam и ключевые слова

- Ответное сообщение **a --> b** или **b <-- a** или **return**
- Название диаграммы:
 - на картинке: **title**
 - только в .puml файле: **@startuml название**
 - **header** Page Header
 - **footer** Page %page% of %lastpage%
- Alias **as** и **order** с шагом 10

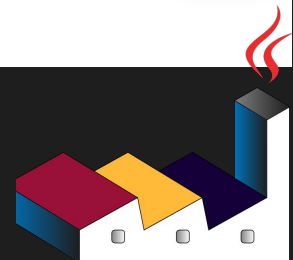
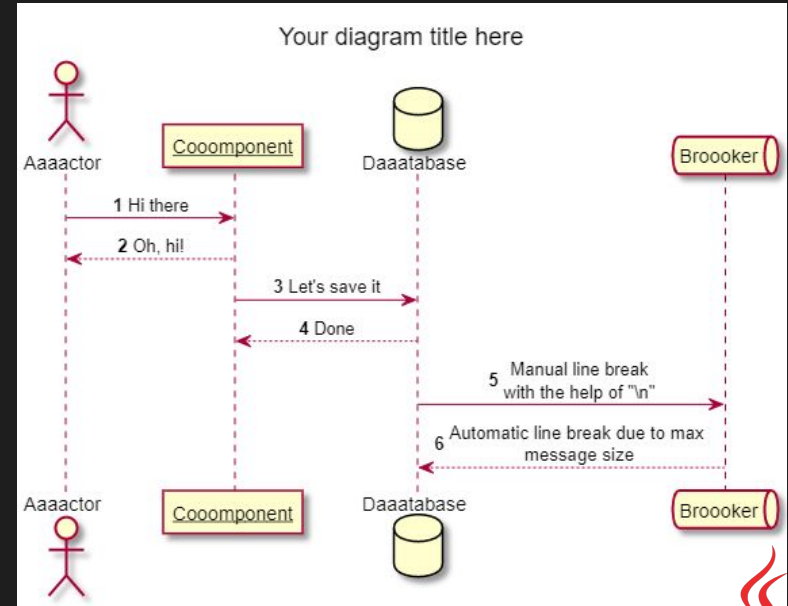


```
@startuml
    slide_skinparam
    skinparam {
        MaxmessageSize 200
    }
    skinparam sequence {
        ParticipantPadding 30
        MessageAlign center
    }
}
```

```
autonumber
title Your diagram title here
```

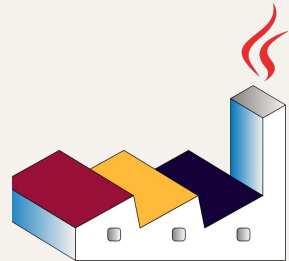
```
actor Aaaactor as ac order 10
participant "__Cooomponent__" as a order 20
database Daaatabase as db order 30
queue "Brooker" as b order 40
```

```
ac -> a: Hi there
return Oh, hi!
a -> db: Let's save it
db --> a: Done
db -> b: Manual line break\nwith the help of "\\n"
return Automatic line break due to max message size
@enduml
```



Skinparam и ключевые слова

- Ответное сообщение **a --> b** или **b <-- a** или **return**
- Название диаграммы (**title**, **header**, **footer**)
- Alias **as** и **order** с шагом 10
- Автонумерация **Autonumber**
- Равные отступы между компонентами **ParticipantPadding** 30

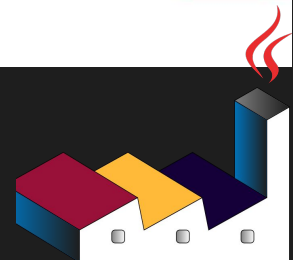
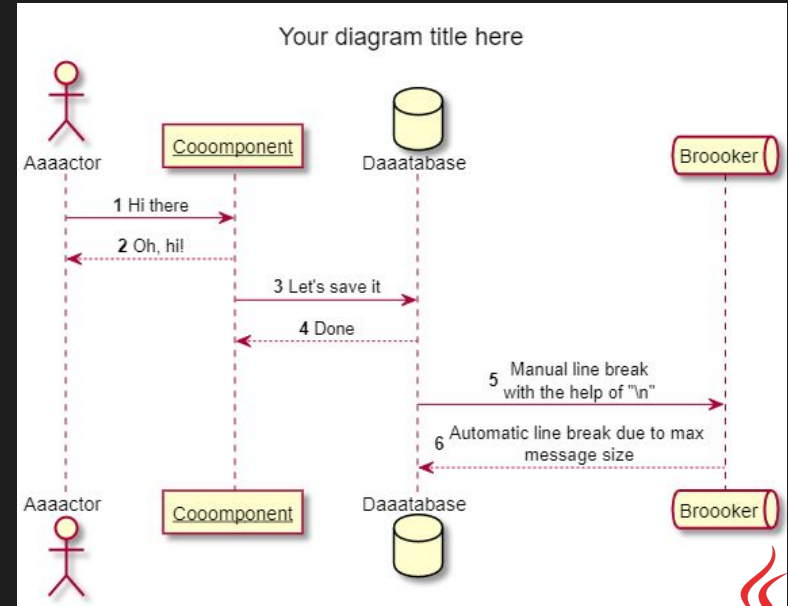


```
@startuml
    slide_skinparam
    skinparam {
        MaxmessageSize 200
    }
    skinparam sequence {
        ParticipantPadding 30
        MessageAlign center
    }
}
```

```
autonumber
title Your diagram title here
```

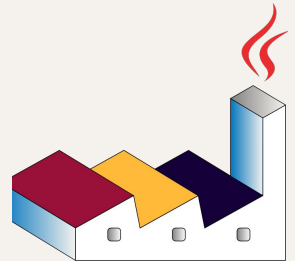
```
actor Aaaactor as ac order 10
participant "__Cooomponent__" as a order 20
database Daaatabase as db order 30
queue "Brooker" as b order 40
```

```
ac -> a: Hi there
return Oh, hi!
a -> db: Let's save it
db --> a: Done
db -> b: Manual line break\nwith the help of "\\n"
return Automatic line break due to max message size
@enduml
```



Skinparam и ключевые слова

- Ответное сообщение **a --> b** или **b <-- a** или **return**
- Название диаграммы (**title**, **header**, **footer**)
- Alias **as** и **order** с шагом 10, автонумерация **Autonumber**
- Равные отступы между компонентами **ParticipantPadding** 30
- Перенос строки **\n**
 - Автоматический: **Maxmessagesize** 300
 - **\n** Экранировать с помощью ещё ****, так: **"\\n"**

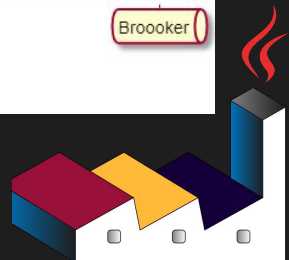
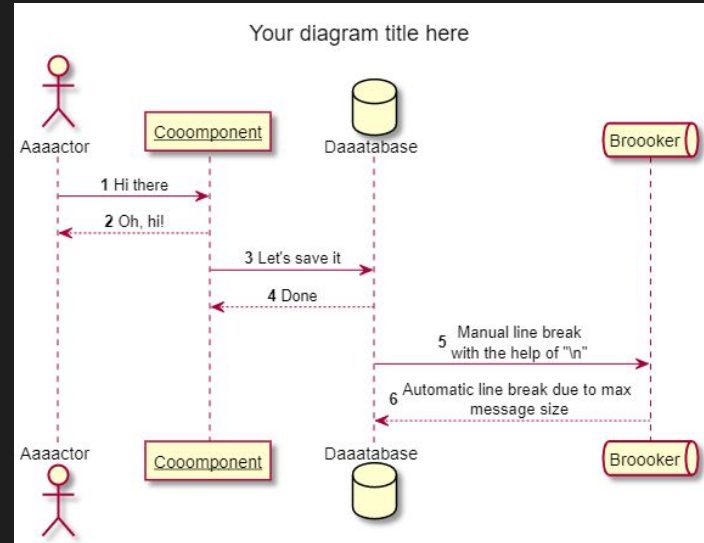


```
@startuml
    slide_skinparam
    skinparam {
        MaxmessageSize 200
    }
    skinparam sequence {
        ParticipantPadding 30
        MessageAlign center
    }
}
```

```
autonumber
title Your diagram title here
```

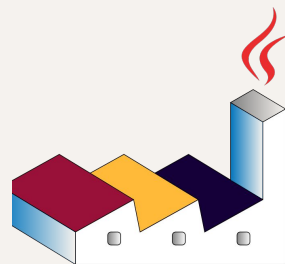
```
actor Aaaactor as ac order 10
participant "__Cooomponent__" as a order 20
database Daaatabase as db order 30
queue "Broooker" as b order 40
```

```
ac -> a: Hi there
return Oh, hi!
a -> db: Let's save it
db --> a: Done
db -> b: Manual line break\nwith the help of "\\n"
return Automatic line break due to max message size
@enduml
```



Цветовая схема

- Сегменты сети
- True / false или main / alternative / exception flow
- Прочие сущности



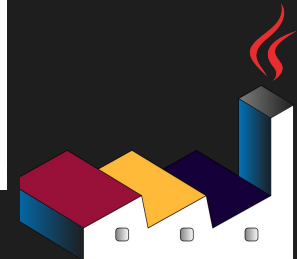
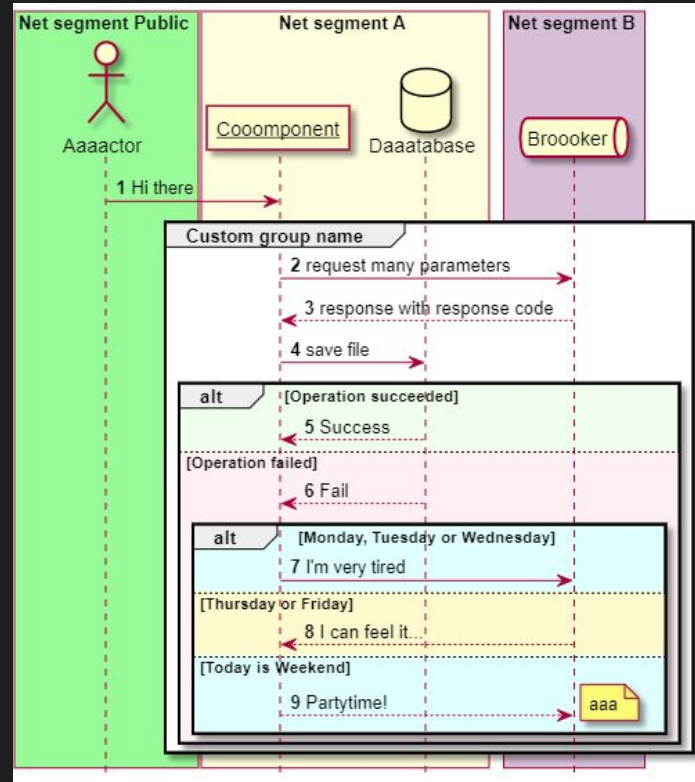
```
@startuml color_scheme
```

```
'this is single line comment  
/' this is  
multi line comment  
'/
```

```
autonumber  
hide footbox
```

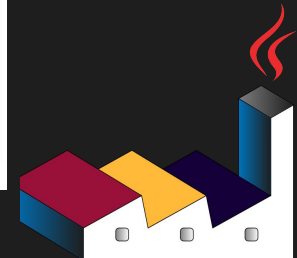
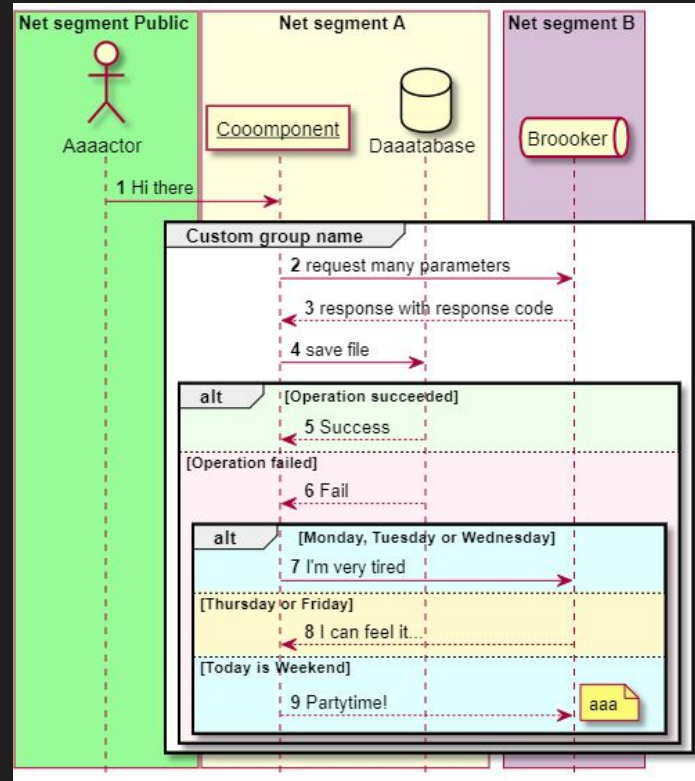
```
box Net segment Public #PaleGreen  
    actor Aaaaactor as act  
end box  
box Net segment A #LightYellow  
    participant " Cooomponent__" as a  
    database Daaatabase as db  
end box  
box Net segment B #Thistle  
    queue "Brooker" as b  
end box
```

```
... NEXT SLIDE ...
```



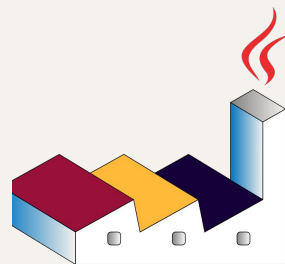
... PREVIOUS SLIDE ...

```
act -> a: Hi there
Group Custom group name
  a -> b: request many parameters
  return response with response code
  a -> db: save file
  alt #f0feed Operation succeeded
    return Success
  else #fff0f5 Operation failed
    a <-- db: Fail
    alt #LightCyan Monday, Tuesday
    or Wednesday
      a -> b: I'm very tired
    else #LemonChiffon Thursday or
    Friday
      return I can feel it...
    else Today is Weekend
      return Partytime!
      note right: aaa
    end
  end
end
@enduml
```



Цветовая схема

- Сегменты сети
 - True / false или main / alternative / exception flow
 - Прочие сущности
-
- Табы, комментарии и группы
 - Единый стиль и префикс автонумерации
 - Начальный номер и шаг нумерации

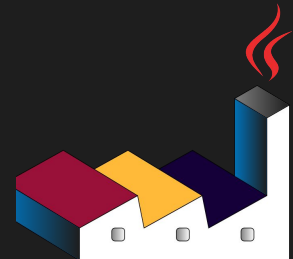
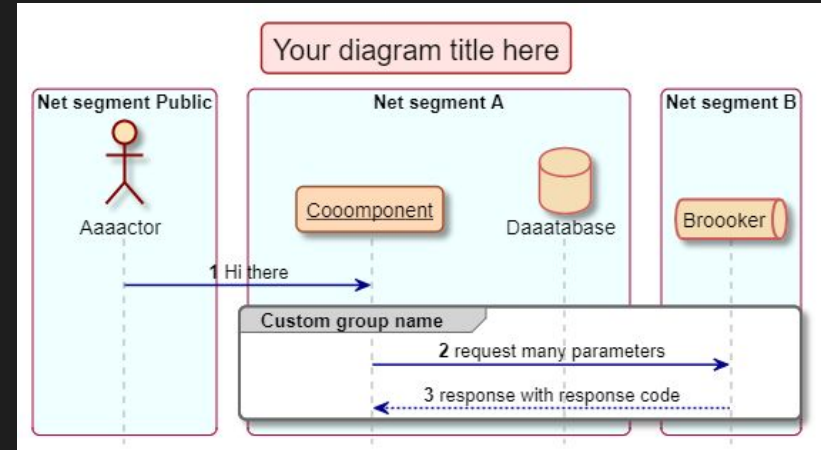


```

@startuml
    styles
    skinparam {
        'monochrome true
        'actorStyle awesome
        BackgroundColor White
        BoxPadding 10
        Roundcorner 10
        MaxmessageSize 300
        DatabaseBackgroundColor Wheat
        DatabaseBorderColor IndianRed
        DatabaseBorderThickness 2
        DatabaseFontColor Black
        DatabaseFontSize 14
        QueueBackgroundColor Wheat
        QueueBorderColor IndianRed
        QueueBorderThickness 2
        QueueFontColor Black
        QueueFontSize 14
    }
    skinparam note {
        BackgroundColor BlanchedAlmond
        BorderColor SandyBrown
    }

```

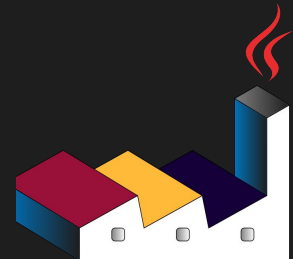
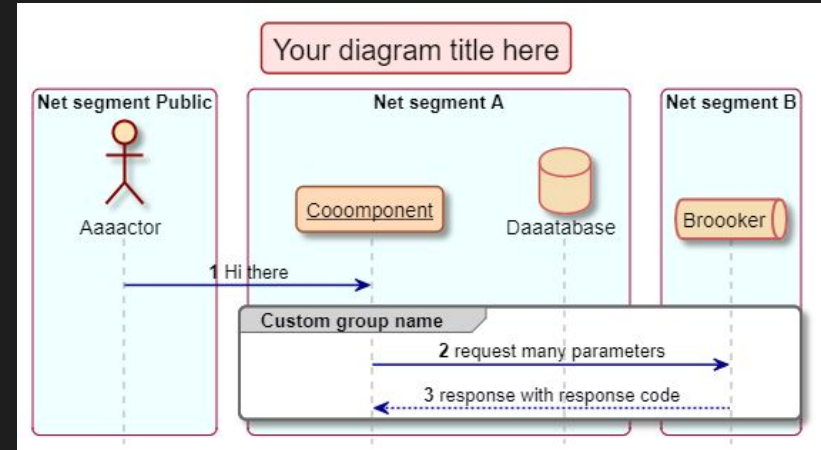
... NEXT SLIDE ...



```

... PREVIOUS SLIDE ...
skinparam title {
  BorderRadiusCorner 10
  BorderThickness 1.5
  BorderColor Firebrick
  BackgroundColor MistyRose
  FontSize 20
}
skinparam sequence {
  ArrowThickness 1.5
  ParticipantFontSize 14
  ParticipantPadding 30
  BoxBackgroundColor Azure
  MessageAlign center
  ArrowColor DarkBlue
  ArrowFontColor Black
  LifeLineBorderColor DarkGray
  LifeLineBackgroundColor SlateGray
  GroupBackgroundColor LightGray
  GroupBorderColor DimGray
  ActorBackgroundColor Moccasin
  ActorBorderColor Maroon
  ActorFontColor Black
  ParticipantBackgroundColor PeachPuff
  ParticipantBorderColor Sienna
  ParticipantFontColor Black
} ... NEXT SLIDE ...

```

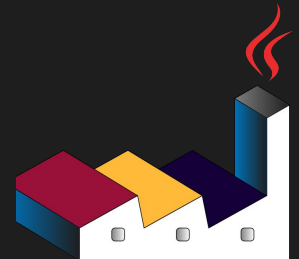
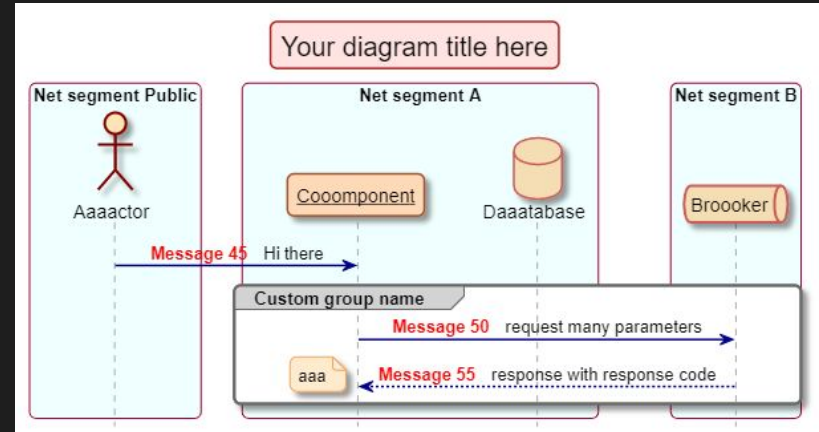


... PREVIOUS SLIDE ...

```
hide footbox
autonumber 45 5 "<font color=red><b>Message 0  "
title Your diagram title here

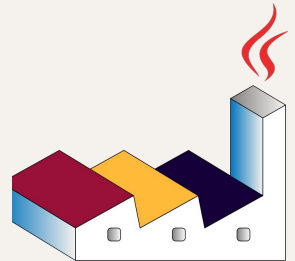
box Net segment Public
  actor Aaaactor as act
end box
box Net segment A
  participant "__Cooomponent__" as a
  database Daaatabase as db
end box
box Net segment B
  queue "Brooker" as b
end box

act -> a: Hi there
Group Custom group name
  a -> b: request many parameters
  return response with response code
  note left: aaa
end
@enduml
```



Таблицы, разделы, ссылки

- Задать легенду
 - Название, расположение, заглавная строка
- Задать разделы и отступы
 - Нумерованные и маркированные списки
- Вставлять гиперссылки
 - При экспорте в svg ссылки сохраняются на картинке



```

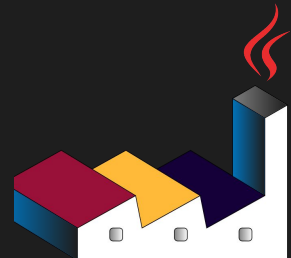
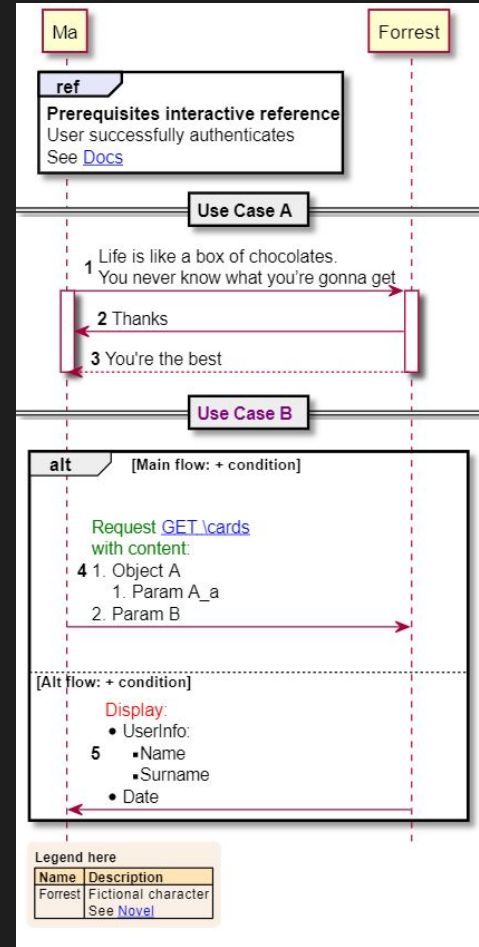
@startuml slide_table
skinparam Legend {
    BackgroundColor Linen
    FontColor black
    BorderThickness 0
    FontSize 10
}
autonumber
hide footbox

ref#Lavender over Ma
    **Prerequisites interactive reference**
    User successfully authenticates
    See [[https://plantuml.com/news Docs]]
end ref

== Use Case A ==
Ma -> Forrest: Life is like a box of choco...
activate Ma
activate Forrest
    Forrest -> Ma: Thanks
    return You're the best
    'F. auto-deactivates due to "return".
    'Return is for operation before F's act-n
deactivate Ma

... NEXT SLIDE ...

```

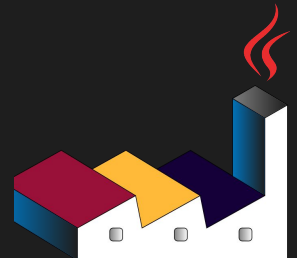
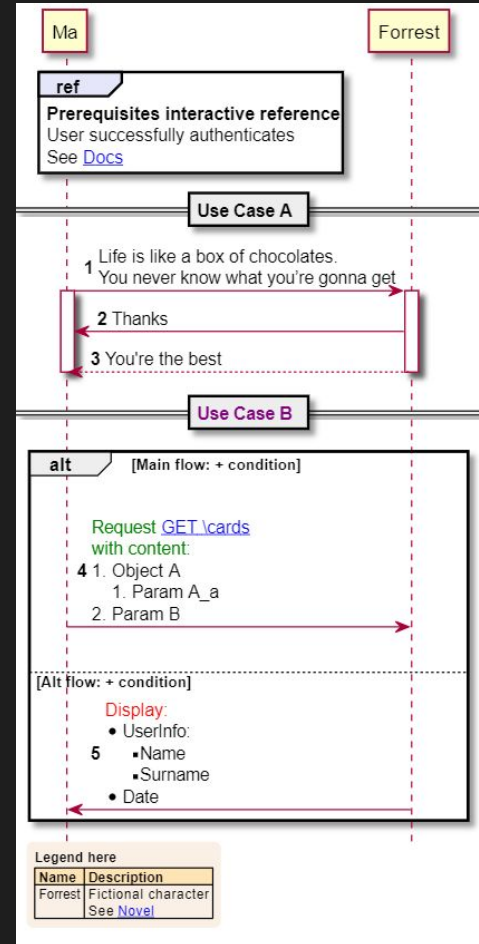


... PREVIOUS SLIDE ...

```
== <font color=Purple>Use Case B ==
alt Main flow: + condition
  |||
  Ma -> Forrest: <font color=green>Request
  [[https://plantuml.com/news GET \cards]] \n<font
  color=green>with content: \n#Object A \n##Param
  A_a \n#Param B
  |||
else Alt flow: + condition
  Forrest -> Ma: <font color=red>Display:
  \n*UserInfo: \n**Name \n**Surname \n*Date
end

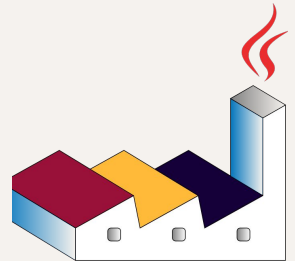
legend left
  ' possible parameters: left | right | top |
  bottom | center
  **Legend here**
  <#Moccasin>|= Name |= Description |
  | Forrest | Fictional character \n See
  [[https://plantuml.com/news Novel]]|
endlegend
```

@enduml



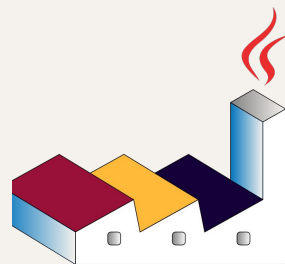
Возвратные функции

- Препроцессинг как в Си
 - **!function** *\$func_name(\$argument_name_a, \$arg_2=default_value, ...)*
!return *"Some operation with args" + \$arg_N*
!endfunction



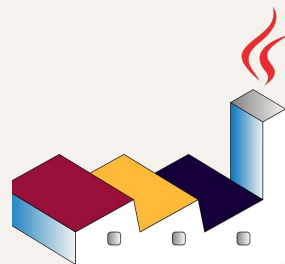
Возвратные функции

- Препроцессинг как в Си
 - **!function** *\$func_name(\$argument_name_a, \$arg_2=default_value, ...)*
!return *"Some operation with args" + \$arg_N*
!endfunction
- Можно вызвать:
 - Вернув значение в переменную или сразу на "стрелку" sequence'a
 - Внутри другой возвратной функции или процедуры



Возвратные функции

- Препроцессинг как в Си
 - **!function** *\$func_name(\$argument_name_a, \$arg_2=default_value, ...)*
!return *"Some operation with args" + \$arg_N*
!endfunction
- Можно вызвать:
 - Вернув значение в переменную или сразу на "стрелку" sequence'a
 - Внутри другой возвратной функции или процедуры
- Переменные
 - **!\$variable_name** = *value* (сразу глобальная)
 - **!local \$variable_name** = *value* (локальная)



```

@startuml slide_function
!function $func_double($a)
!return $a + $a
!endfunction

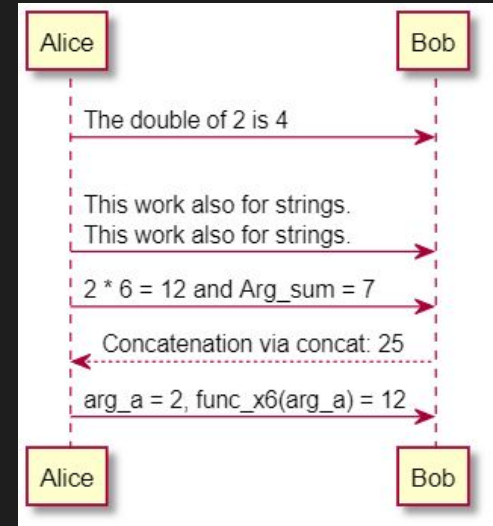
!function $func_x6($a)
!return 6*$a
!endfunction

!function $concat($a, $b)
!return "Bob --> Alice : " + "Concatenation via concat: " + $a + $b
!endfunction

!$arg_a = 2
!$arg_b = 5
!$arg_sum = $arg_a + $arg_b

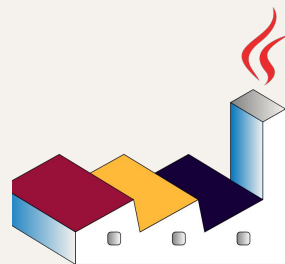
Alice -> Bob : The double of $arg a is $func_double($arg a)
Alice -> Bob : $func_double("\nThis work also for strings.")
Alice -> Bob : $arg a * 6 = $func_x6($arg_a) and Arg_sum = $arg_sum
$concat($arg_a, $arg_b)
!if ($arg_a == 2) && ($func_x6($arg_a) >= 12)
    Alice -> Bob : arg_a = $arg_a, func_x6(arg_a) = $func_x6($arg_a)
!else
    Alice -> Bob : ERROR!
!endif
@enduml

```

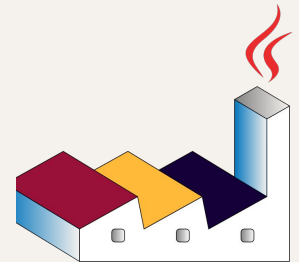
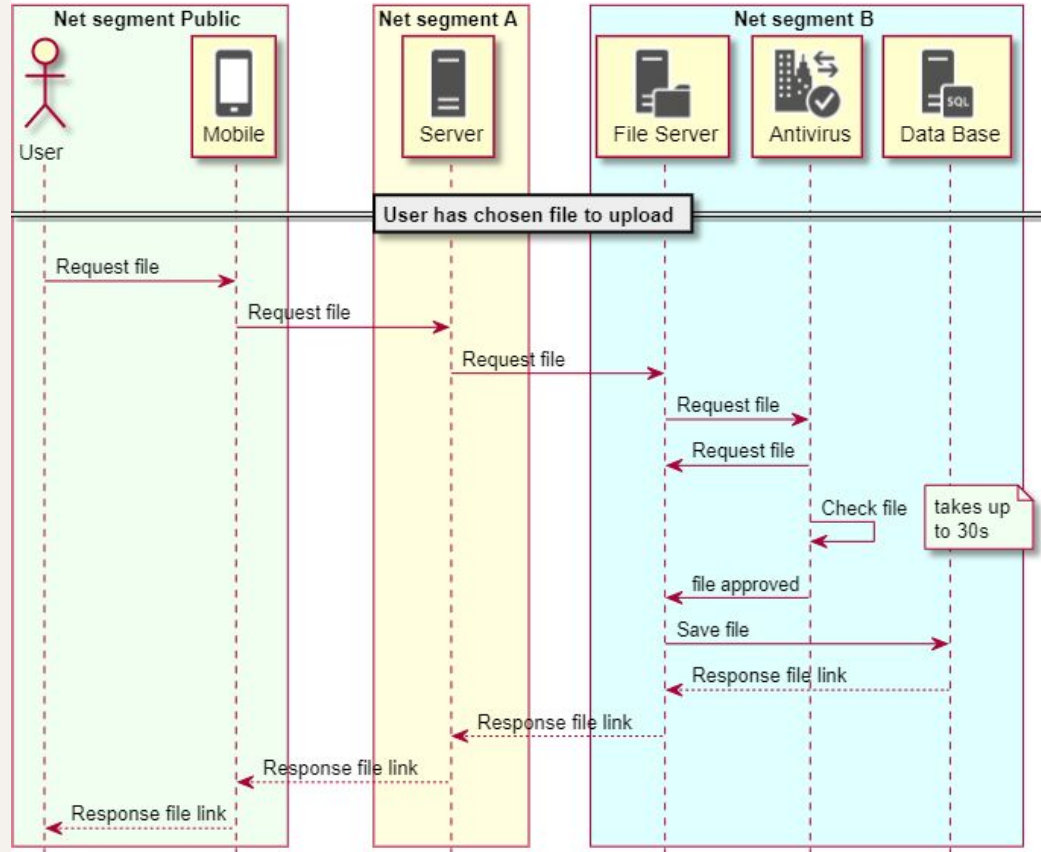


Процедуры

- Процедура может вызывать другую процедуру
 - **!procedure** \$proc_name(\$arg_a, \$arg_b=default_value, ...)
!return "Some operation with args" + \$arg_N
!endprocedure
- **%invoke_procedure**("\$proc_name", \$arg_a, \$arg_N)



Upload file and check via AntiVirus



```

@startuml upload_file
!include https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/procedures_library.puml

' Function acts as a global parameter of root URL address with icons
!function $ICONURL()
!return "<img:https://raw.githubusercontent.com/Roemer/plantuml-office/master/office2014/  "
!endfunction

' Another way to define exact URL to
!function $participant icon($alias)
!return $ICONURL() + "/Devices/cell_phone_generic.png>\n" + $alias
!endfunction

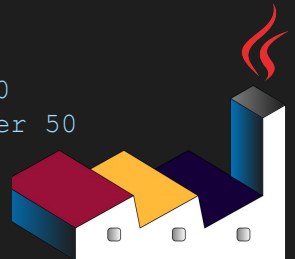
box Net segment Public #Honeydew
    actor User as user order 10
    participant "$participant_icon(Mobile)" as client order 20
end box

box Net segment A #LightYellow
    participant "$ICONURL()/Servers/server_generic.png>\n Server" as server order 30
end box

box Net segment B #LightCyan
    participant "$ICONURL()/Servers/file_server.png>\n File Server" as fs order 40
    participant "$ICONURL()/Security/federation_trust.png>\n Antivirus" as av order 50
    participant "$ICONURL()/Servers/sql_server.png>\n Data Base" as db order 60
end box

```

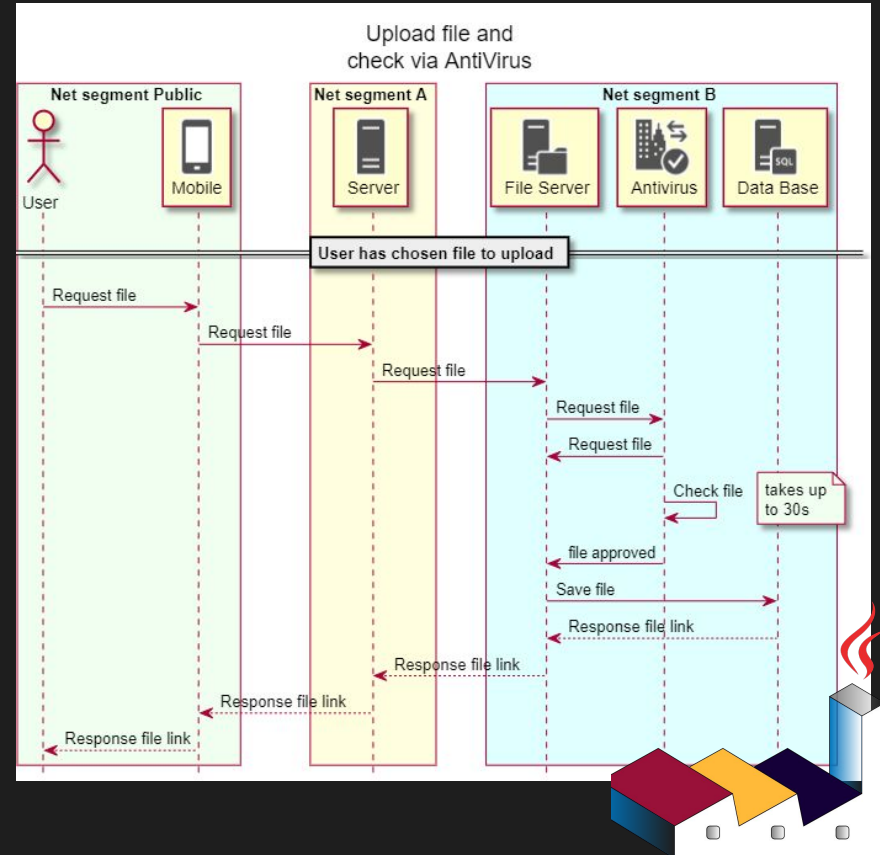
... NEXT SLIDE ...



... PREVIOUS SLIDE ...

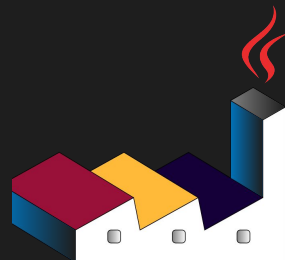
```
title
  Upload file and
  check via AntiVirus
end title
hide footbox
== User has chosen file to upload ==

!$msg = "file"
%invoke_procedure("$proxy", req($msg), "user,
client, server, fs, av, fs")
av -> av: Check $msg
note right #Honeydew: takes up\nto 30s
av -> fs: $msg approved
fs -> db: Save $msg
!$msg = $msg + " link"
%invoke_procedure("$proxy response",
resp($msg), "db, fs, server, client, user")
@enduml
```



Файл с процедурами

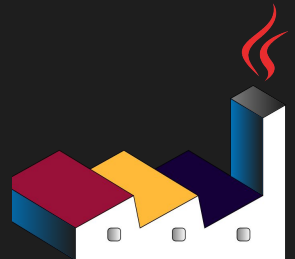
```
'  
' Library with useful procedures.  
' Include this procedures_library.puml file in your main .puml file to use procedures.  
' Use following construction in your main .puml file:  
' %invoke_procedure("$procedure_name", argument1, ... )  
'  
  
@startuml procedures_lib  
  
!function req($text)  
!return "Request " + $text  
!endfunction  
  
!function resp($text)  
!return "Response " + $text  
!endfunction  
  
... NEXT SLIDE ...
```



```

!procedure $proxy($text, $actors)
'Procedure takes as input:
  'text -- will be displayed upon each arrow
  'actors -- string in "", where strictly COMMA delimited actors must follow in desired order
!$string = $actors
'input string of actors and first input string length constant
!$len = %strlen($string)
!$pos = %strpos($string, ",")
'returns position of symbol in argument. Starts from 0.
!$start = 0
'defines parsing start position. iteration starts with input string length
!$i=$len
!$j = 0
!while $pos >= 0
  !$j = $j + 1
  !$source = $res
  !if $j == 1
    !$source = $res
  !endif
  !$res = %substr($string, $start, $pos)
  !$dest = $res
  !if $j > 1
    $source -> $dest: $text
  !endif
  !$string = %substr($string, $pos+1, $len - $pos)
  !$i=%strlen($string)
  !$pos = %strpos($string, ",")
!endwhile
!$j = $j + 1
!$source = $res
!$res = $string
!$dest = $res
$source -> $dest: $text
!endprocedure

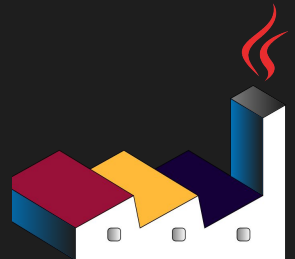
```



```

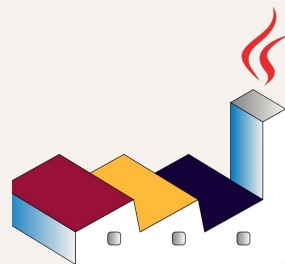
!procedure $proxy response($text, $actors)
'Procedure takes as input:
  'text -- will be displayed upon each arrow
  'actors -- string in "", where strictly COMMA delimited actors must follow in desired order
!$string = $actors
'input string of actors and first input string length constant
!$len = %strlen($string)
!$pos = %strpos($string, ",")
'returns position of symbol in argument. Starts from 0.
!$start = 0
'defines parsing start position. iteration starts with input string length
!$i=$len
!$j = 0
!while $pos >= 0
  !$j = $j + 1
  !$source = $res
  !if $j == 1
    !$source = $res
  !endif
  !$res = %substr($string, $start, $pos)
  !$dest = $res
  !if $j > 1
    $source --> $dest: $text
  !endif
  !$string = %substr($string, $pos+1, $len - $pos)
  !$i=%strlen($string)
  !$pos = %strpos($string, ",")
!endwhile
!$j = $j + 1
!$source = $res
!$res = $string
!$dest = $res
$source --> $dest: $text
!endprocedure
@enduml

```



Бонус: макеты

- Salt
 - Можно строить макеты
 - Можно включать в sequence и activity diagram



```

@startuml slide_salt_1
salt
{
    Just plain text
    [This is my button]
    () Unchecked radio
    (X) Checked radio
    [] Unchecked box
    [X] Checked box
    "Enter text here "
    ^This is a droplist^
}

{+
    Text1
    ..
    "Some field"
    ==
    Note on usage
    ~~
    Another text
    --
    [Ok]
}

... NEXT SLIDE ...

```

Just plain text

This is my button

☐ Unchecked radio

☒ Checked radio

☐ Unchecked box

☒ Checked box

Enter text here

This is a droplist

Text1

Some field

Note on usage

Another text

Ok

My group box:

File	Edit	Source	Refactor
General	Fullscre	New	avior
Open image in:	Sm	Open File	Saving
<input checked="" type="checkbox"/> Smooth images		Close	med
<input checked="" type="checkbox"/> Confirm image deletion		Close All	
<input type="checkbox"/> Show hidden images			
Close			

Login

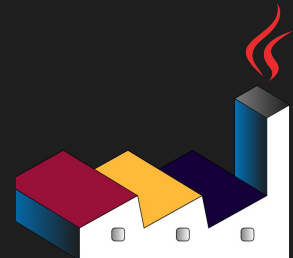
MyName

Password

Cancel

Ok

Region	Population	Age
World	7.13 billion	30
America	964 million	30
Canada	35 million	30
USA	319 million	30
NYC	8 million	30
Boston	617 thousand	30
Mexico	117 million	30
Europe	601 million	30
Italy	61 million	30
Germany	82 million	30
Berlin	3 million	30
Africa	1 billion	30



... PREVIOUS SLIDE ...

```
{^"My group box"
  {+
    { * File | Edit | Source | Refactor
      Refactor | New | Open File | - |
    Close | Close All }
    { / <b>General | Fullscreen | Behavior |
    Saving }
    {
      { Open image in: | ^Smart Mode^ }
      [X] Smooth images when zoomed
      [X] Confirm image deletion
      [ ] Show hidden images
    }
    [Close]
  }
  Login<&person> | "MyName "
  Password<&key> | "**** "
  [Cancel <&circle-x>] | [OK <&account-login>]
}
```

... NEXT SLIDE ...

Just plain text

This is my button

☐ Unchecked radio
☒ Checked radio
☐ Unchecked box
☒ Checked box

Enter text here

This is a droplist

Text1

Some field

Note on usage

Another text

Ok

My group box:

File	Edit	Source	Refactor
General	Fullscreen	New	avior
Open image in: Sm	Open File		
<input checked="" type="checkbox"/> Smooth images	Close		med
<input checked="" type="checkbox"/> Confirm image deletion	Close All		
<input type="checkbox"/> Show hidden images			

Close

Login

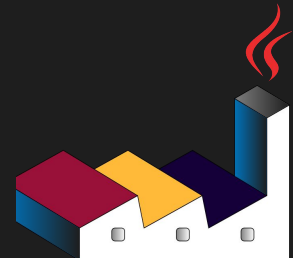
MyName

Password

Cancel

OK

Region	Population	Age
World	7.13 billion	30
America	964 million	30
Canada	35 million	30
USA	319 million	30
NYC	8 million	30
Boston	617 thousand	30
Mexico	117 million	30
Europe	601 million	30
Italy	61 million	30
Germany	82 million	30
Berlin	3 million	30
Africa	1 billion	30



... PREVIOUS SLIDE ...

```
{
  {T
  +Region | Population | Age
  + World | 7.13 billion | 30
  ++ America | 964 million | 30
  +++ Canada | 35 million | 30
  +++ USA | 319 million | 30
  ++++ NYC | 8 million | 30
  ++++ Boston | 617 thousand | 30
  +++ Mexico | 117 million | 30
  ++ Europe | 601 million | 30
  +++ Italy | 61 million | 30
  +++ Germany | 82 million | 30
  ++++ Berlin | 3 million | 30
  ++ Africa | 1 billion | 30
}

}

@enduml
```

Just plain text

This is my button

☐ Unchecked radio
☒ Checked radio
☐ Unchecked box
☒ Checked box

Enter text here

This is a droplist

Text1

Some field

Note on usage

Another text

Ok

My group box:

File Edit Source Refactor

General Fullscreen New vior Saving

Open image in: Sm Open File

☒ Smooth images Close med

☒ Confirm image deletion Close All

☐ Show hidden images

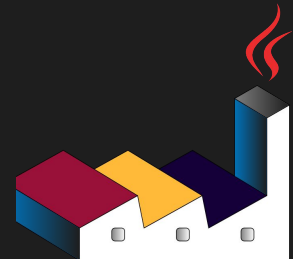
Close

Login MyName

Password ****

Cancel OK

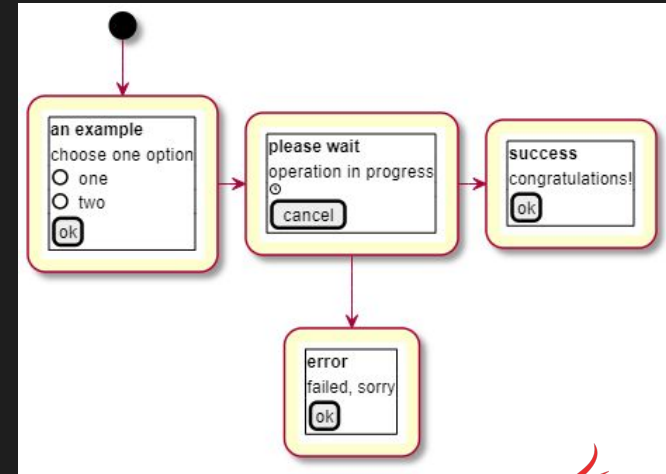
Region	Population	Age
World	7.13 billion	30
America	964 million	30
Canada	35 million	30
USA	319 million	30
NYC	8 million	30
Boston	617 thousand	30
Mexico	117 million	30
Europe	601 million	30
Italy	61 million	30
Germany	82 million	30
Berlin	3 million	30
Africa	1 billion	30



```

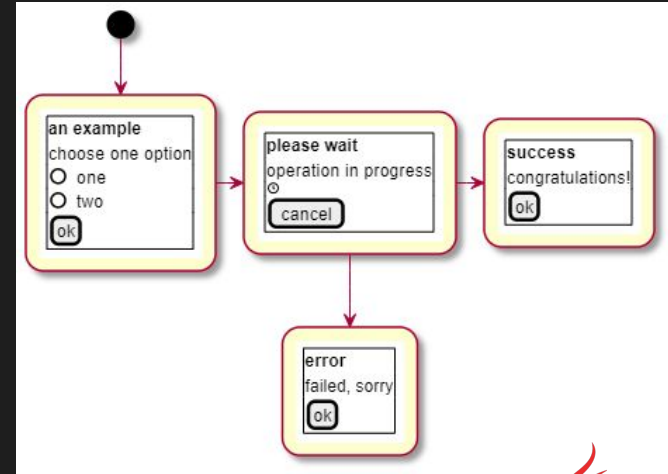
@startuml slide_salt_2
(*) --> "
{{
salt
    {+
        <b>an example
        choose one option
        () one
        () two
        [ok]
    }
}}
" as choose
choose -right-> "
{{
salt
    {+
        <b>please wait
        operation in progress
        <&clock>
        [cancel]
    }
}}
" as wait
wait -right-> "
... NEXT SLIDE ...

```



... PREVIOUS SLIDE ...

```
{{
salt
  {+
    <b>success
    congratulations!
    [ok]
  }
}}
" as success
wait -down-> "
{{
salt
  {+
    <b>error
    failed, sorry
    [ok]
  }
}}
"
@enduml
```

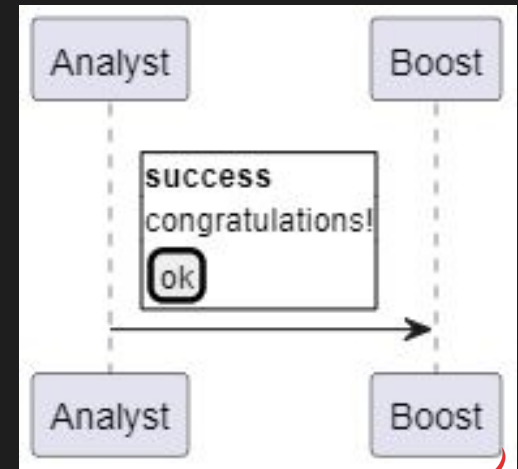


```
@startuml salt_in_sequence
```

```
!$val = "{  
    "salt  
    "{+  
    "<b>success  
    "congratulations!  
    "[ok]  
    }  
}"
```

```
Analyst -> Boost: $val
```

```
@enduml
```



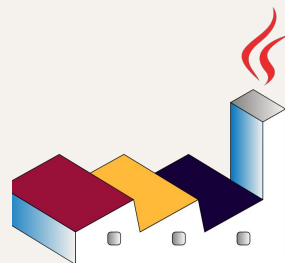
4

Агрегация

Для командной работы

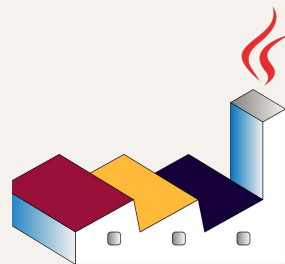
Распределённая работа

- Подключаем Git и работаем в общем репозитории
- Импортируем *.puml* внутрь диаграммы
 - `!includeurl` https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/style_plantuml_file.puml



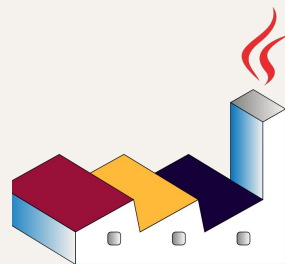
Распределённая работа

- Подключаем Git и работаем в общем репозитории
- Импортируем *.puml* внутрь диаграммы
 - `!includeurl` https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/style_plantuml_file.puml
- Каждая команда работает над своей частью, а в агрегате end-to-end
- Независимые изменения частей не требуют обновления агрегата



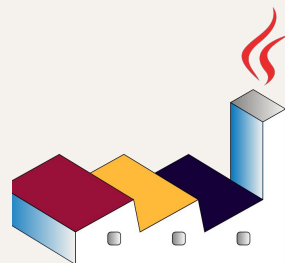
Распределённая работа

- Подключаем Git и работаем в общем репозитории
- Импортируем *.puml* внутрь диаграммы
 - `!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/style_plantuml_file.puml`
- Каждая команда работает над своей частью, а в агрегате end-to-end
- Независимые изменения частей не требуют обновления агрегата
- Командный файл стилей \ функций \ процедур
- Нейминги \ alias участников совпадают по всем файлам



Включение

- **O** BaseURL
- Описание локальной функции по возврату базового URL вашего пространства к импортируемым файлам
- **A** Прямой импорт файла
- Необходимо вставить `!include` в нужное место диаграммы-агрегата в вашем коде.
- **B** Вставка в функцию
- Вызываем заданную выше функцию BaseURL и передаём ей название файла или путь к нему.



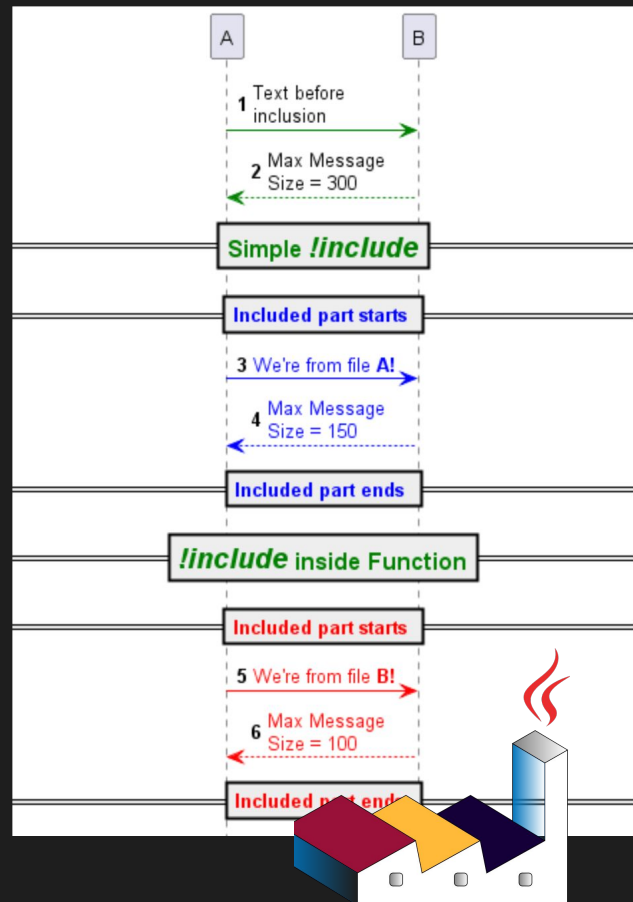
```
@startuml include_aggreagate
skinparam MaxMessageSize 300
autonumber

!function $BASEURL($filename)
!return "https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/" + $filename
!endfunction
```

```
A-[#green]>B: Text before inclusion
A<-[#green]-B: Max Message Size = 300
==== <color #green>Simple //!include//</color> ==
!include
https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/include_part_a.puml

==== <color #green>//!include// inside Function</color> ==
!include $BASEURL(include_part_b.puml)

... NEXT SLIDE ...
```



%LoadJSON

- **CL** Процедура %LoadJSON
- Внешняя процедура, позволяющая загрузить json по ссылке. Можно картинкой, можно кодом, но без json-форматирования.
- **SOS** Возможные трудности
- Alias'ы участников должны совпадать по всем диаграммам
- Будет применён тот параметр отображения, который встретился в коде последним (например, MaxMessageSize применяется = 100. Аналогично и для прочих, общих на весь файл параметров)
- Импорт не сработает, если безопасность не разрешает доступ из вашего контура вовне



... PREVIOUS SLIDE ...

```
==== <color #green>Procedure //loadJson//</color> ==  
!$link = $BASEURL(json_example_short.json)  
!$my_json = %loadJSON($link)
```

A-[#green]>B: Text after inclusion. Send message with json as a note

hnote across #white

```
{ {json  
    $my_json  
  } }
```

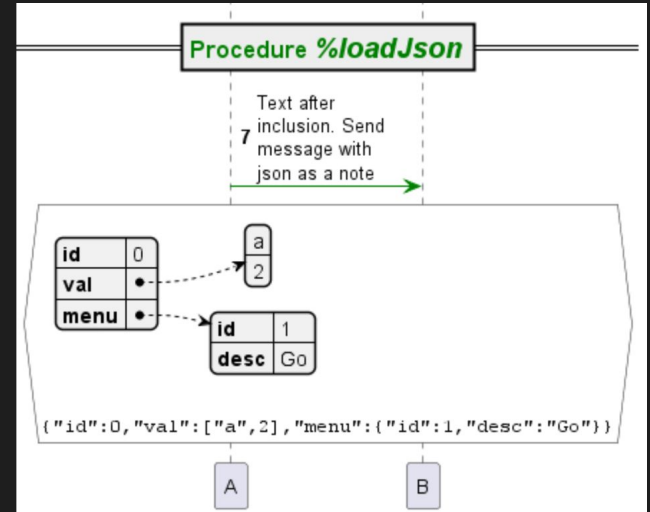
<code>

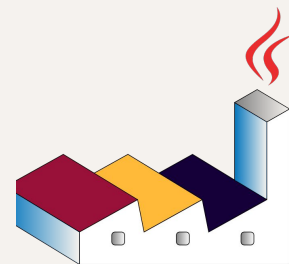
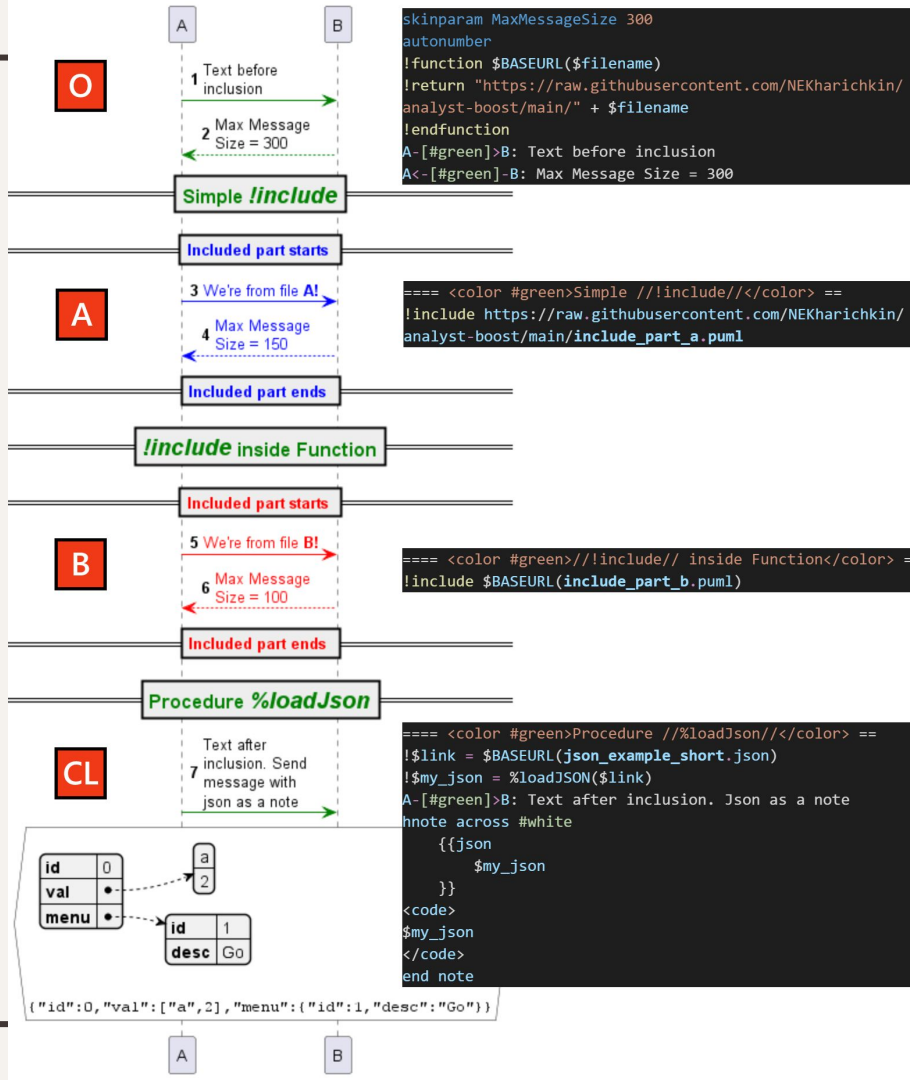
\$my_json

</code>

end note

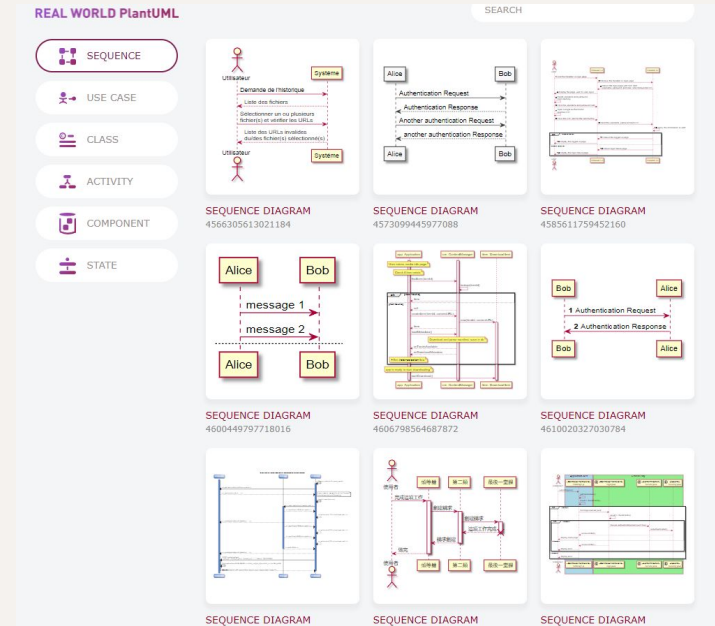
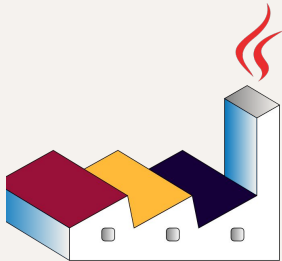
@enduml





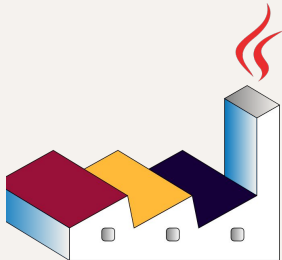
Real World PlantUML

- Реальные примеры из практики
- Real World PlantUML



The Hitchhiker's Guide

- The Hitchhiker's Guide to PlantUML



USING STANDARD LIBRARY

1. Create a Diagram of a Typical Network
2. Create Real Life AWS Diagrams
3. Kubernetes and Azure
4. Google Cloud Platform
5. Using C4 To Architect ACME Global Widget Production

PLANTUML FEATURES

1. Use Images in Diagrams
2. Scaling Sprites

3. Procedures and Keyword Arguments

- 3.1. Defines Are Deprecated
- 3.2. Default Values
- 3.3. Keyword Arguments

4. PassSpriteAsParameter

5. Colors
6. Layout
7. Tricked Out Diagrams (aka Diagram Annotation)

UNDERSTANDING STANDARD LIBRARY

1. PlantUML Stdlib Overview
2. PlantUML Stdlib Under The Hood
3. PlantUML Stdlib Under The Hood 2

STANDARDISING STANDARD LIBRARY

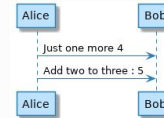
1. Standard Library - What We Have And What We Want

3.2. Default Values

Per `DefaultArgumentValue`, "In both procedure and return functions, you can define default values for arguments"

Tip

The default argument value can help to simplify definitions



Using default argument

```
1 @startuml
2 function $inc($value, $step=1)
3   return $value + $step
4   endfunction
5
6 Alice -> Bob : Just one more $inc(3)
7 Alice -> Bob : Add two to three : $inc(3, 2)
8 @enduml
```

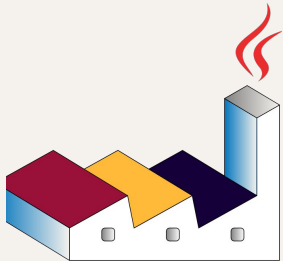
3.3. Keyword Arguments

Current macros are polymorphic e.g. macros are defined twice with a different number of arguments.

However, the ordering of the arguments is hardcoded e.g. first parameter is "alias".

Official guide

- Сайт
- PDF



Sequence Use Case Class Activity Activity **UML** Component State

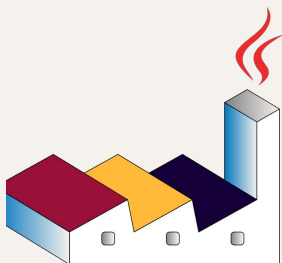
Home
What's New ?
Getting Started
Online Server
Running
F.A.Q.
Download
Forum
Theme
Preprocessing
Standard Library
Hitchhiker's Guide
PDF Guide

You can also have the following syntax:

```
@startuml
participant Alice
participant Bob #lightblue
Alice -> Bob
Bob -> Carol
...
[-> Bob
[o-> Bob
[o->o Bob
[x-> Bob
...
[<- Bob
[x<- Bob
...
Bob ->]
Bob ->o]
Bob o->o]
Bob ->x]
...
Bob <-]
Bob x<-]
```

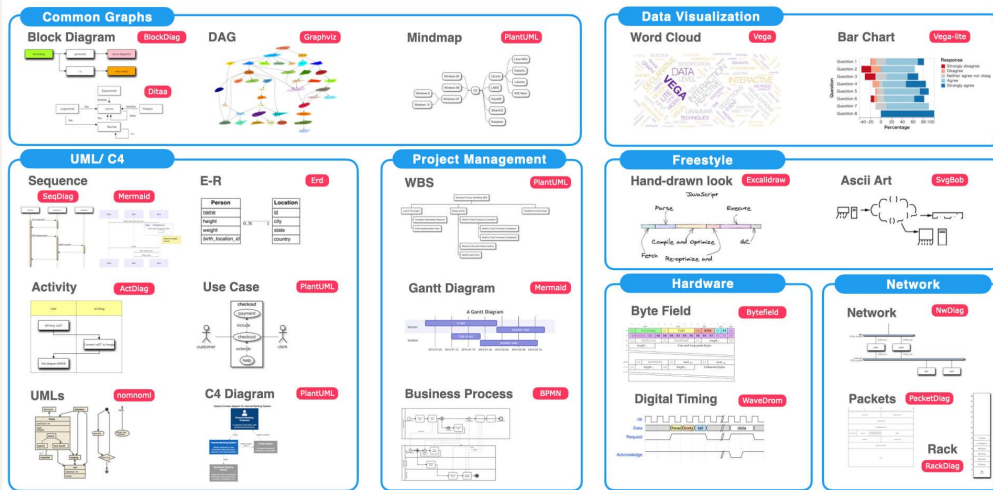
Другие

- Kroki!
- Text to diagram



Kroki Diagrams Zoo

creates diagrams from textual descriptions by Yuzu tech



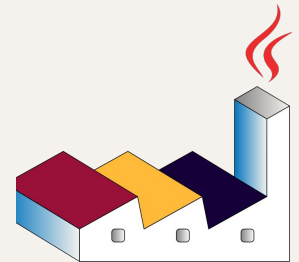
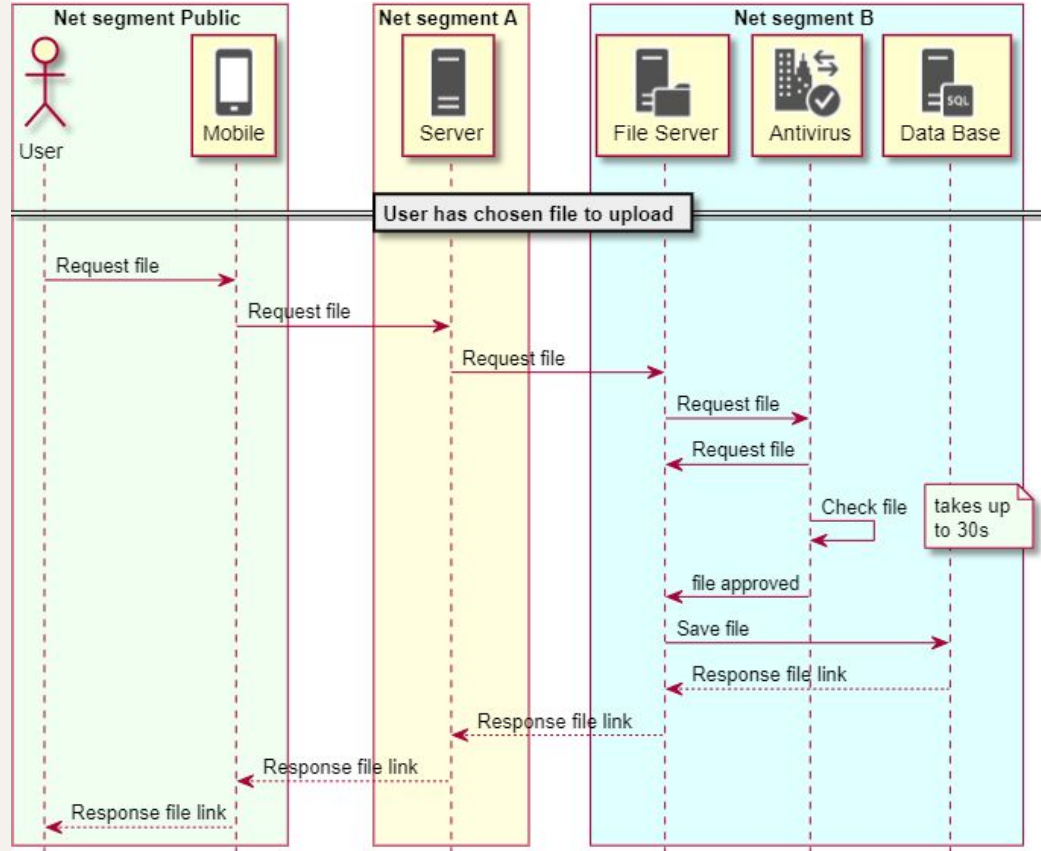
<https://kroki.io/> - <https://github.com/yuzutech/kroki>

Cheat sheet v.1.1 - 2021/05/15 - created by Tomo Masuda // @tomo-makes

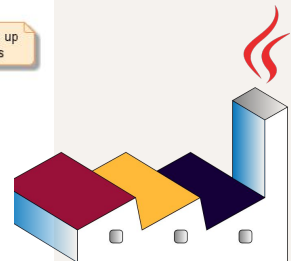
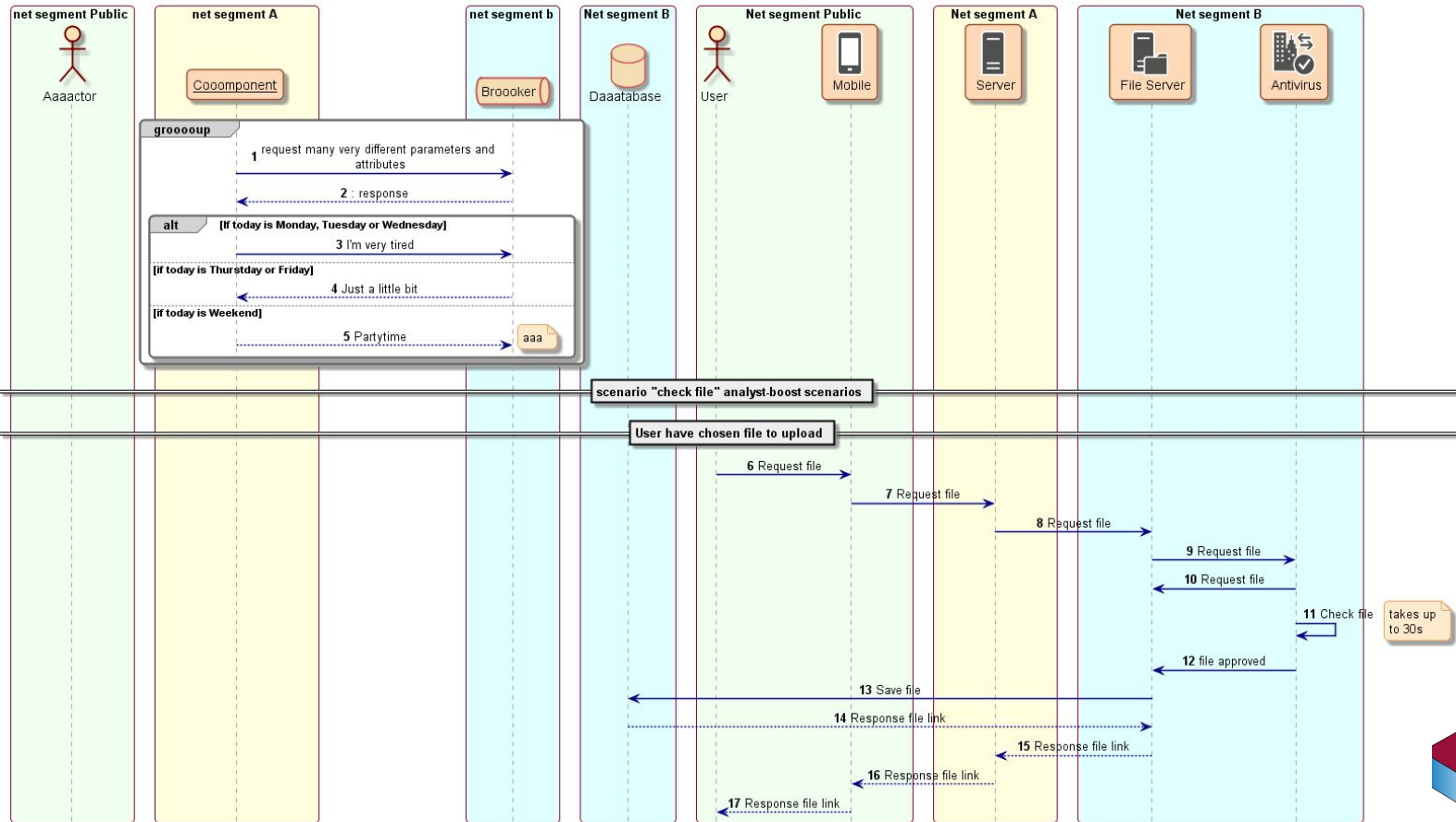
Community list of comparisons between Text to Diagram tools



Upload file and check via AntiVirus



Aggregated doc



```

@startuml aggregate
' == STYLE ==
' Import colour name from here:
' https://en.wikipedia.org/wiki/Web_colors#Extended_colors
'   e.g. "#lightGreen". Named colour helps to interpret
'   and understand the connection between uml code and the picture while editing
' Open style ".puml" file as "raw", then copy link and
' insert it under "!includeURL".
'   Make sure to include only 1 style, otherwise it may lead to style mix
'   Example of 3rd party style lib import below:
'!includeurl https://raw.githubusercontent.com/bschwarz/puml-themes/master/themes/cerulean/puml-theme-cerulean.puml

' For styles stored in one repo bt in multiple folders
' you may use define to shorten links for readability. See example below
'!define ICONURL https://raw.githubusercontent.com/Roemer/plantuml-office/master/office2014

' == Analyst Boost style ==
!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/style_plantuml_file.puml

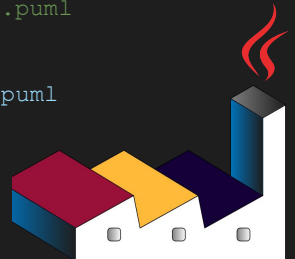
' == PROCEDURES LIBRARY ==
!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/procedures_library.puml

' == scenario 3rd party example ==
'!includeurl https://raw.githubusercontent.com/bschwarz/puml-themes/master/examples/sequence-ex.puml

== scenario "check file" analyst-boost scenarios ==
!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/feature_scenario.puml

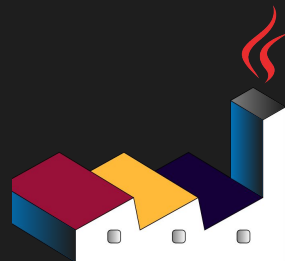
' If you want to override all the titles from the included, you must define it after inclusion
title Aggregated doc
@enduml

```



Всего 7 строчек

```
@startuml aggregate
!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/style plantuml file.puml
!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/procedures_library.puml
== scenario "check file" analyst-boost scenarios ==
!includeurl https://raw.githubusercontent.com/NEKharichkin/analyst-boost/main/feature_scenario.puml
title Aggregated doc
@enduml
```



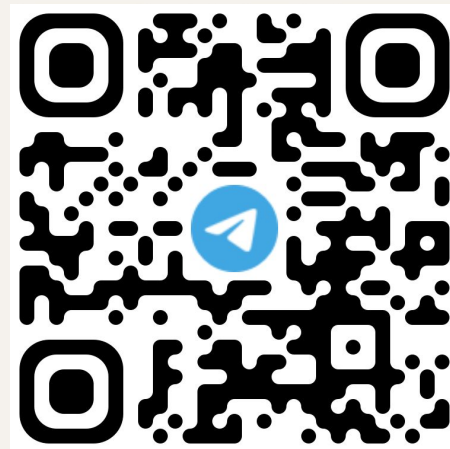
Контакты

Группа @plantuml



Mail: nekharichkin@yandex.ru

@Analyst_Boost



Спасибо!

Вопросы

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**