

## Use with `knitr`

### Register the plantuml knitr engine

The plantuml knitr engine `plantuml_knit_engine()` needs to be registered with `knitr`. The easiest is to use the helper function `plantuml_knit_engine_register` in the `setup` code chunk. It is important to load the package `plantuml` before, as some options need to be set.

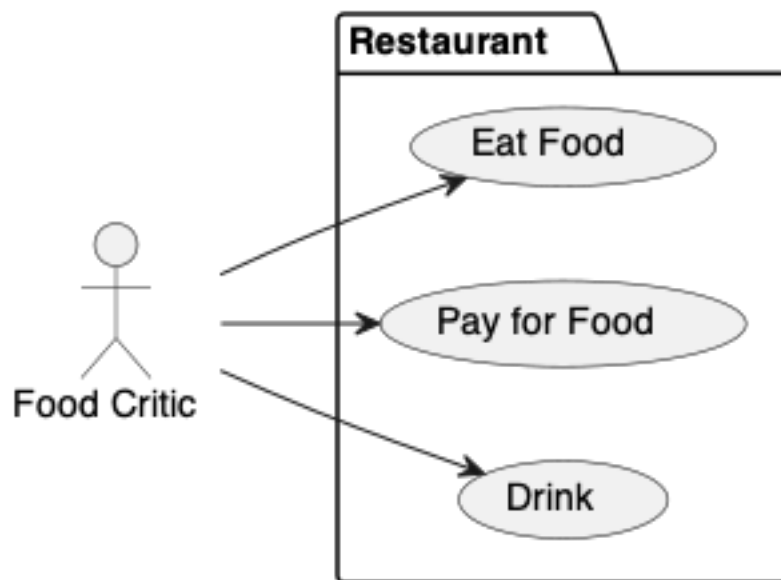
```
library(plantuml)
```

### PlantUML Code Chunk Options

The following options are available, in addition to the standard options: - `plantuml_format`: the format of the resulting file. At the moment, `png` (the default), `svg`, and `eps` are implemented. - `plantuml.path`: the path at which the resulting graphs should be saved. If the directory does not exist, it will be created. - `plantuml.preview`: if the graph should be shown as an inline preview in the Rmarkdown document in Rstudio. This option results in generating the image twice, which may lead to a substantial increase of the processing time. The preview is **always** a vector preview, irrespective of the file `plantuml_format`

### A simple chart as png

A png graph is nice, because it can be shown in all kinds of documents, including html, pdf and docx. But it is a bitmap format, i.e. when scaling in the graph becomes pixelated.



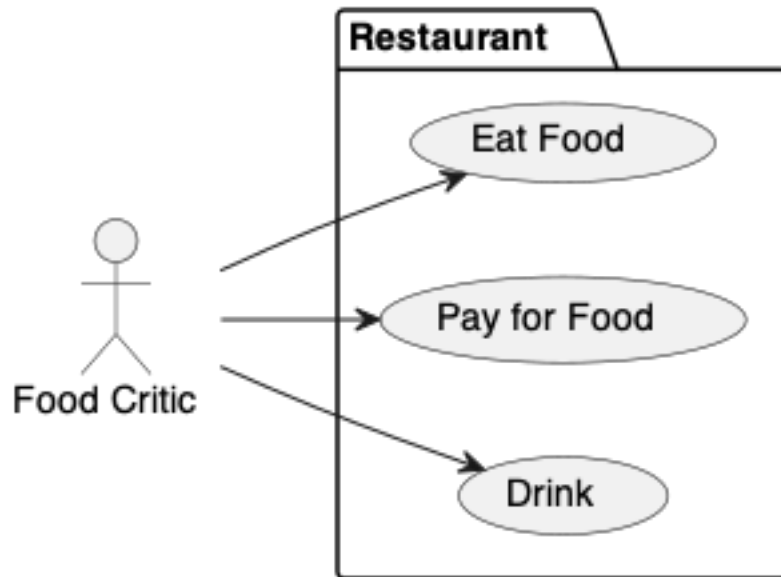
### A simple chart as png and with code

```
left to right direction
actor "Food Critic" as fc
package Restaurant {
```

```

usecase "Eat Food" as UC1
usecase "Pay for Food" as UC2
usecase "Drink" as UC3
}
fc --> UC1
fc --> UC2
fc --> UC3

```



## A simple chart using format auto with code

When specifying `plantuml.format = "auto"` the appropriate vector format is chosen for the output format:

- pdf output format: `pdf` as `plantuml.format`
- html output format: `svg` as `plantuml.format`
- docx output format: `pdf` as `plantuml.format`

Therefore, the optimal vector format is chosen for the output format.

```

robust "DNS Resolver" as DNS
robust "Web Browser" as WB
concise "Web User" as WU

```

```

@0
WU is Idle
WB is Idle
DNS is Idle

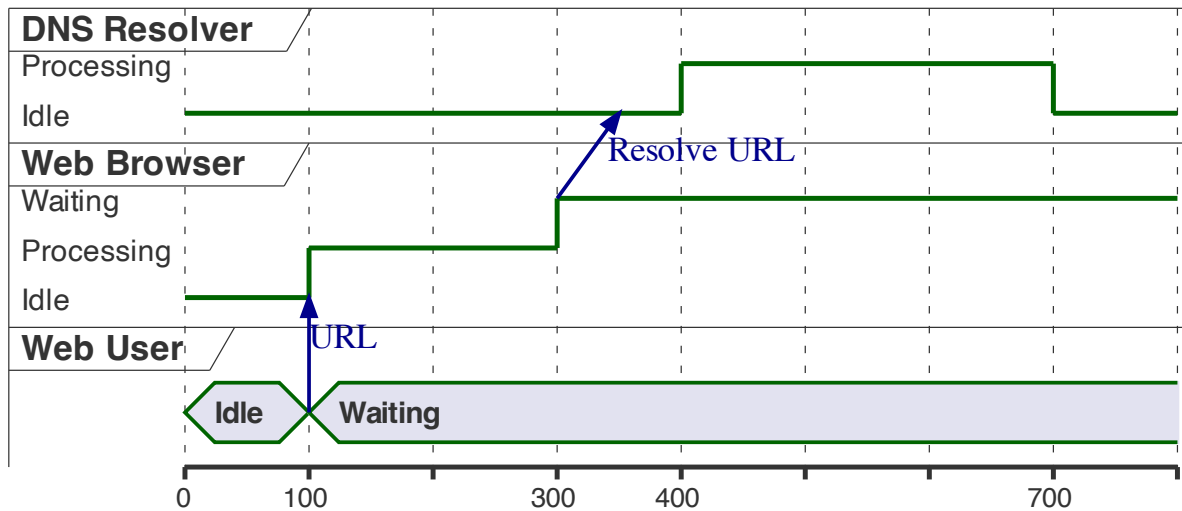
@+100
WU -> WB : URL
WU is Waiting
WB is Processing

@+200
WB is Waiting
WB -> DNS@+50 : Resolve URL

```

```
@+100
DNS is Processing
```

```
@+300
DNS is Idle
```



Now only the code

```
@startuml
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

Alice -> Bob: Another authentication Request
Alice <-- Bob: Another authentication Response
@enduml
```

And now as ASCII Art

This does not play nicely with LaTeX and therefore pdf due to unicode characters used. I have no idea at the moment how to make this nicer.

```
@startuml
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

Alice -> Bob: Another authentication Request
Alice <-- Bob: Another authentication Response
@enduml
```

Alice

Bob

Authentication Request

>

Authentication Response

<

Another authentication Request

>

Another authentication Response

<

Alice

Bob