

# PlantUML

Thierry Vaira [tvaira@free.fr](mailto:tvaira@free.fr)

2020 rev 1

## Table des matières

<b>PlantUML</b>	<b>1</b>
Présentation . . . . .	1
Exemples . . . . .	1
Paramètres . . . . .	4
Doxygen . . . . .	4
Markdown . . . . .	5
Voir aussi . . . . .	6

## PlantUML

Lien : [PlantUML](#)

Documentation : [plantuml-documentation.readthedocs.io](http://plantuml-documentation.readthedocs.io)

## Présentation

[PlantUML](#) est un outil open source permettant aux utilisateurs de créer des diagrammes UML à partir d'un langage de texte brut. Il utilise le logiciel [Graphviz](#) pour disposer ses diagrammes.

[PlantUML](#) permet de dessiner rapidement des :

- diagrammes de séquence
- diagrammes de cas d'utilisation
- diagrammes de classes
- diagrammes d'activité (ici l'ancienne syntaxe)
- diagrammes de composant
- diagrammes d'état
- diagrammes d'objet
- diagrammes de déploiement
- diagrammes de temps

Certains autres diagrammes (hors UML) sont aussi possibles :

- schéma réseau
- maquette d'interface graphique
- diagrammes de Gantt
- diagrammes d'idées
- ...

**Les diagrammes sont définis à l'aide d'un langage simple et intuitif.** Les images peuvent être générées au format PNG, SVG ou LaTeX.

Serveur pour réaliser des diagramme en ligne : <http://www.plantuml.com/plantuml>

## Exemples

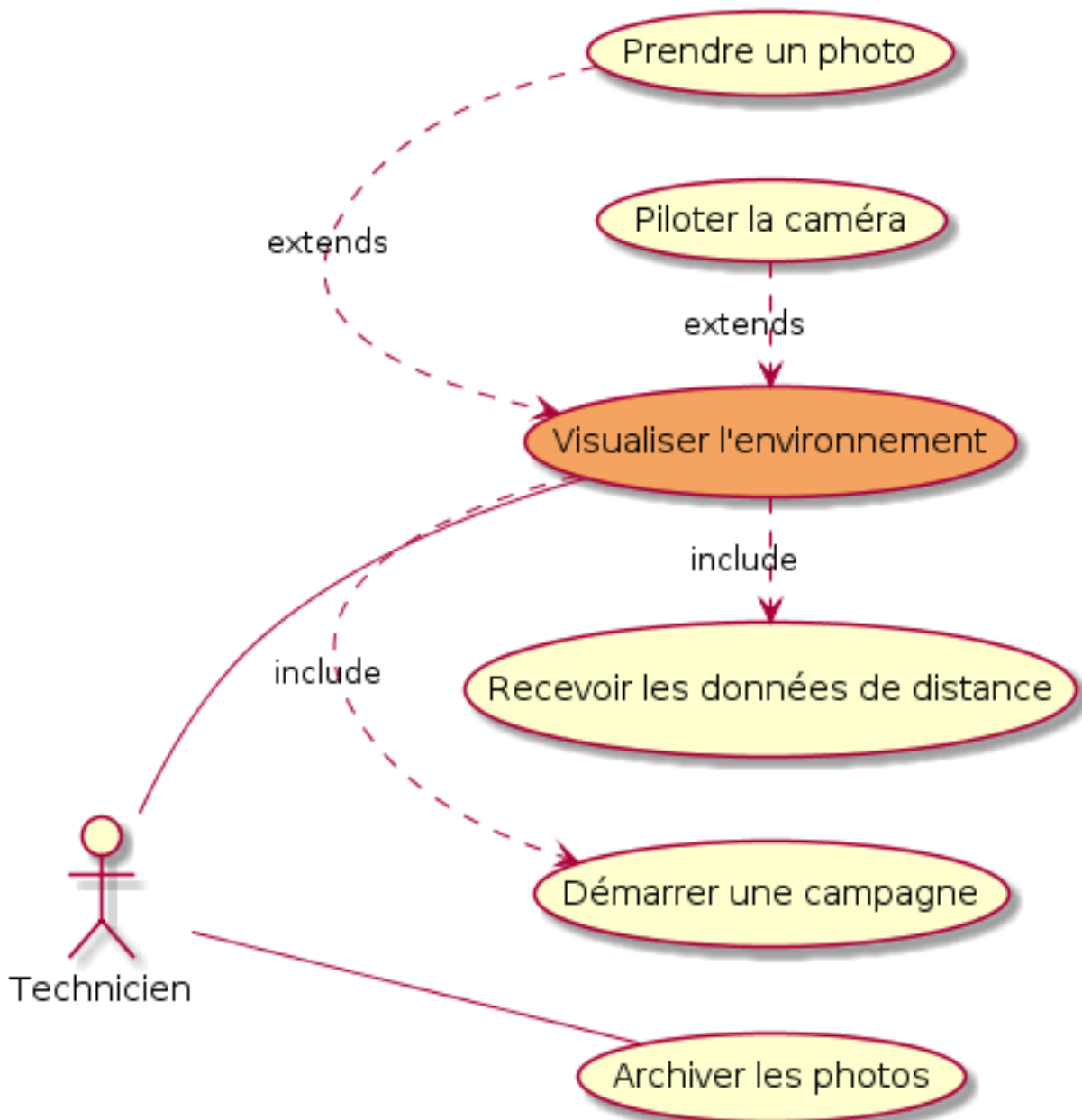
- [diagramme de cas d'utilisation](#)

Documentation : [plantuml-documentation.readthedocs.io](http://plantuml-documentation.readthedocs.io)

```

@startuml
left to right direction
hide stereotype
skinparam packageStyle rectangle
skinparam usecase {
  BackgroundColor<< perso >> SandyBrown
}
actor Technicien
rectangle roynet {
  (Visualiser l'environnement) << perso >>
  Technicien -- (Visualiser l'environnement) << perso >>
  (Visualiser l'environnement) .> (Démarrer une campagne) : include
  (Visualiser l'environnement) .> (Recevoir les données de distance) : include
  (Prendre un photo) .> (Visualiser l'environnement) : extends
  (Piloter la caméra) .> (Visualiser l'environnement) : extends
  Technicien -- (Archiver les photos)
}
@enduml

```



Un diagramme cas d'utilisation généré par PlantUML

## — diagramme de séquence

Documentation : [plantuml-documentation.readthedocs.io](http://plantuml-documentation.readthedocs.io)

```
@startuml
hide footbox
actor Utilisateur
participant Ihm
participant Supervision
database Bdd

Utilisateur ->> Ihm: Clic sur le bouton "Rechercher"
note over Utilisateur, Ihm #aqua: L'utilisateur a saisi un nom d'article\nà rechercher
activate Ihm

Ihm-> Ihm: clicked() <<signal>>
Activate Ihm

Ihm->Ihm: rechercherArticle() <<slot>>
Activate Ihm

opt recherche valide
  Ihm->Supervision: rechercherArticle()
  Activate Supervision
  Deactivate Ihm

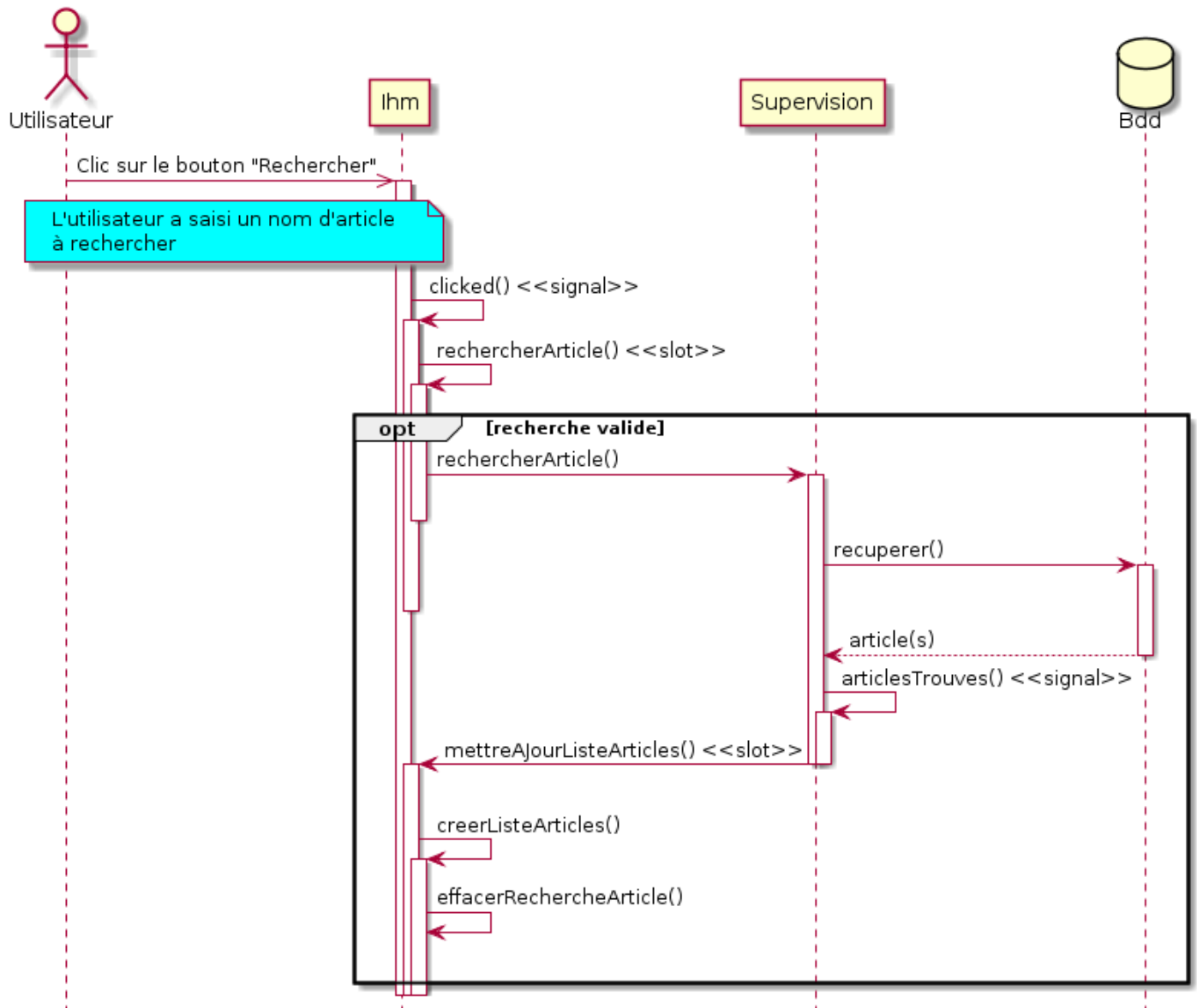
  Supervision->Bdd: recuperer()
  Activate Bdd
  Deactivate Ihm

  Bdd-->Supervision: article(s)
  Deactivate Bdd

  Supervision->Supervision: articlesTrouves() <<signal>>
  Activate Supervision

  Supervision->Ihm: mettreAJourListeArticles() <<slot>>
  Deactivate Supervision
  Activate Ihm
  |||
  Deactivate Supervision
  Ihm->Ihm: creerListeArticles()
  Activate Ihm

  Ihm->Ihm: effacerRechercheArticle()
  |||
end
@enduml
```



Un diagramme de séquence généré par PlantUML

## Paramètres

<https://plantuml.com/fr/skinparam>

## Doxygen

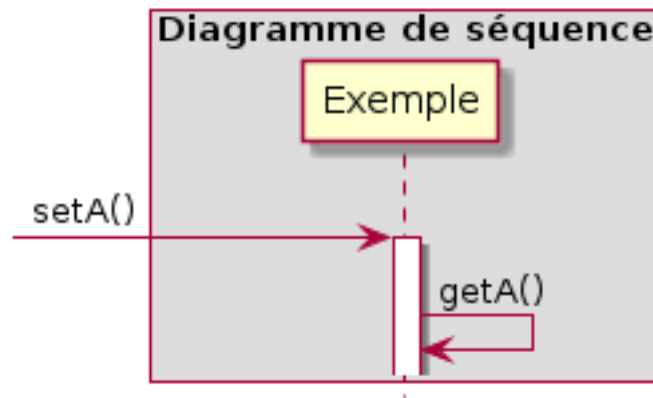
Il est possible d'intégrer des diagrammes PlantUML dans un document généré par Doxygen.

```

/**
 * ...
 *
 * \startuml
 *   hide footbox
 *   skinparam BoxPadding 50
 *   box "Diagramme de séquence"
 *     participant Exemple
 *   end box
 *   [-> Exemple: setA()
 *   Activate Exemple
 *   Exemple->Exemple: getA()
 * \enduml

```

\*/



Un diagramme de séquence généré par PlantUML

Pour cela, il suffit d'installer :

```
$ sudo apt-get install plantuml
```

Et d'ajouter le chemin dans le fichier Doxyfile :

```
PLANTUML_JAR_PATH      = /usr/share/plantuml/
```

## Markdown

Si vous utilisez VSCode, il existe une extension pour intégrer des diagrammes PlantUML directement dans vos documents .md : [jebbs.plantuml](https://github.com/jebbs/plantuml).

```
````plantuml
@startuml
hide footbox
actor User
participant "First Class" as A
participant "Second Class" as B
participant "Last Class" as C

User -> A: DoWork
activate A

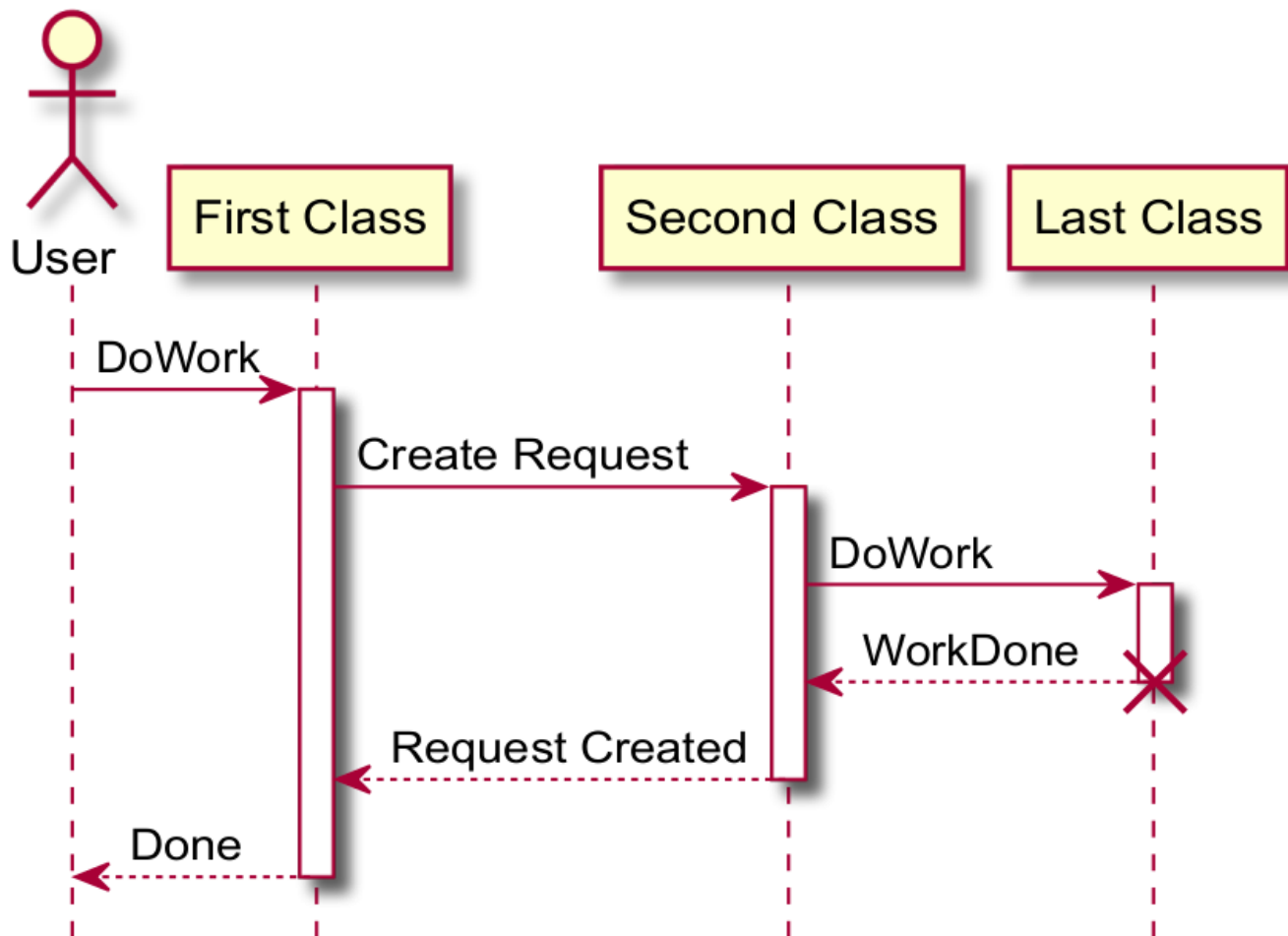
A -> B: Create Request
activate B

B -> C: DoWork
activate C
C --> B: WorkDone
destroy C

B --> A: Request Created
deactivate B

A --> User: Done
deactivate A

@enduml
````
```



Un diagramme de séquence généré par PlantUML

### Voir aussi

- [Mermaid](#) : diagrammes et organigrammes
- [Graphviz](#) : création de graphiques