



# Votalisk

## Membres du projet

- Josian Chevalier
- Vladislav Fitz
- Quỳnh Nga Nguyễn
- Salmon Thomas
- Mohamed Skhiri

## Présentation

L'objectif de ce framework scala est de fournir un environnement de gestion électoral, il ne s'agit donc que d'un cœur.

L'intérêt étant que l'on puisse, à souhait, y ajouter facilement la méthode électorale (ie son algorithme) voulue pour une application.

## Utilisation

**Votalisk** laisse l'application, qui l'utilise, décider de la configuration des élections, aussi on attendra seulement les données et exécuter la stratégie fournie par l'utilisateur.

## Diagramme des classes

Le lien entre **Votalisk** et l'application qui l'embarque se fait par la classe *ElectionManager*, avec celle-ci on gère les permissions de votes ainsi que le stockage des scrutins, des candidats, des électeurs...

La partie **Election** du framework est un *Composite*, ceci afin de gérer les élections a tours successifs.

Les élections communales, régionales, nationales, etc sont toutes présentées par la partie **Districte**, également *Composite*. On peut donc présenter un territoire électorale ainsi que ses divisions.

**VotingPaper** permet de représenter les scrutins.

*VoteResult*, *VoteInput* et *VoteOuput* sont des types qui sont à définir

Le package statistique permet de voir les statistique sur un candidat (**CandidatResult**) ou sur un districte (**DistrictResult**)

C'est ce package qui permet de déterminer les résultats dans **ElectionResult**.