

Adjugé, Vendu !

Vous êtes chargé de développer une application de salle de vente électronique.

L'objectif est de proposer et documenter une API permettant aux commissaires priseurs de proposer des enchères en ligne et aux enchérisseurs d'interroger l'ensemble des enchères (ouvertes ou fermées) organisées par la salle de vente ainsi que de se joindre et de participer aux enchères ouvertes.

Les objets du domaine sont les suivants :

- **Auctioneer** : la personne qui crée les enchères (fr.: commissaire priseur)
- **Item** : quelque chose qui peut être identifié et acheté (fr.: l'objet d'une enchère)
- **Bidder** : une personne ou une organisation intéressée par l'achat d'un Item (fr.: enchérisseur)
- **Bid** : l'engagement d'un enchérisseur (Bidder) à payer un prix donné pour un Item (fr.: offre)
- **Current price** : l'offre (Bid) courante la plus haute proposée pour un Item (fr.: prix courant)
- **Auction** : le processus de gestion des offres pour un Item (fr.: enchère)
- **Auction house** : l'institution qui accueille et anime des enchères (fr.: salle de vente)

L'API doit permettre :

- aux commissaires priseurs (Auctioneers)
 - * de planifier de nouvelles enchères (Auctions) en précisant l'objet de l'enchère (Item), le prix de départ (offre plancher), la politique d'incrément à utiliser (constante, par tranches, etc.) ainsi que les dates (horodatées) de départ et de fin de l'enchère
 - * d'interroger l'historique d'une enchère en cours ou fermée
- aux enchérisseurs (Bidders)
 - * de connaître l'activité (passée, courante et à venir) de la salle de vente (notamment en ce qui concerne la situation de l'enchérisseur dans les différentes ventes auxquelles il participe ou a participé)
 - * de se joindre à une enchère (pré-requis à toute participation)
 - * de proposer une offre dans le cadre d'une enchère à laquelle il participe

Le cycle de vie d'une enchère est le suivant :

- Planned : l'enchère a été créée par un commissaire priseur mais n'est pas encore commencée ; elle est de ce fait encore modifiable (sa date d'ouverture peut être anticipée ou repoussée, son prix de départ corrigé, etc.) ; aucun enchérisseur ne peut se joindre à une enchère non ouverte (et donc a fortiori proposer une offre)
- Openned : l'enchère n'est plus modifiable par le commissaire priseur ; seul son prix courant est soumis à variation au gré des offres reçues des enchérisseurs ; ces offres peuvent être acceptées ou refusées en fonction de la politique d'incrément retenue lors de la définition de l'enchère ; seuls les enchérisseurs qui se sont joints à l'enchère peuvent y participer
- Closed : dès que la fin de l'enchère est atteinte, l'enchère est fermée et un gagnant est désigné

La prise en compte de la politique d'incrément est un plus : proposez un incrément constant par défaut. Pour rappel, l'incrément est l'augmentation minimale requise par rapport au prix actuel (ce mécanisme est destiné à éviter une surenchère de 1 centime par exemple).

Cette itération n'inclura :

- ni mécanisme de gestion de droits : l'API est ouverte sur le port 5000 et accessible en curl ou httpie par simple lancement d'un Main (le développeur joue tous les rôles)
- ni packaging de l'application : lancement direct du main dans l'IDE
- ni persistance : modèle in memory

Pile technique :

- langage : scala 2.12
- build : sbt
- API/routes : akka http

Attentes :

- API REST (respect des verbes et code retour)
- présence de tests (scalatest/scalacheck) afin de valider les invariants de l'application (ex.: une enchère en-deçà du prix courant sera toujours rejetée)

Pistes:

- la salle de vente peut être représentée par un acteur Akka
- les différentes enchères peuvent être représentées par des acteurs fils de cet acteur salle de vente
- le cycle de vie de l'enchère peut être modélisé par une state machine