

# Backend challenge

## Nordeus - Job Fair 2023

One of the core features in Top Eleven are auctions! Millions of users are simultaneously competing to sign the best players on the market. Can you make that feature?

Our players (or as we call them- football managers) try to build the most successful football team in their league by signing the best possible players. They look for such players on auctions, where they need to outbid other football managers in order to sign the wanted player. In this challenge you need to write the backend service (AuctionService) which would enable all the fun in the game!

### How do auctions work?

Initially there are 10 players on the auction, with one player in each auction. Users are able to join any active auction, and bid for a player with *Tokens*. Initial player price is *1 Token*, and each bid increases the next bid auction price for *1 Token*. Joined users are notified when a new bid is placed on a player.

Each auction lasts for *1 minute*. If a user places a bid in the last 5 seconds (for example: 2s before the end), the auction is prolonged instantly and new expiration time is set to 5 seconds. Auction ends if there are no bids in the last 5 seconds. When an auction ends, all joined users are notified about the winner of the auction.

Every minute *10 new auctions* are generated, regardless if all previous auctions were finished. Active auctions are these 10 new auctions and all the auctions that are prolonged due to users actively bidding on.

For a more vivid insight, why not set aside 15 minutes to play Top Eleven- Be a Football Manager and experience firsthand how auctions come to life?

### Requirements & Starting point

As a Backend Engineer, your job is to implement functionalities of AuctionService:

- Getting all active auctions
- Start of a new auctions
- Joining an auction
- Bidding on an auction
- Notifying users about:
  - end of auction and the winner
  - placed bid on an auction
  - refresh new auctions generated

The programming language employed for this challenge is **Java**. *You are free to utilize any integrated development environment (IDE) of your choice.*

**Java (Spring boot)** project with initially designed interfaces is prepared on GitHub ([LINK](#)) as a starting point. Fork (or download) project. There you will see:

- **AuctionService interface and dummy AuctionServiceImpl service.** Your job is to finish the implementation of the AuctionServiceImpl class. You have the liberty to change return types/arguments/exception of the methods.
- **Model classes (Auction, AuctionId, User, UserId, Bid)** are there for the purpose of defining interfaces. Feel free to modify and model the classes as you think would be best.
- **HttpController** is meant to define HTTP endpoints. That way (frontend) clients and other services could communicate with AuctionService. You can also test functionalities by manually invoking these endpoints. Define the rest of the endpoints and connect with the AuctionService.

In the event that certain aspects of this challenge are unclear, you have the liberty to decide how those ambiguous parts should function.

### Submission format

Please note that each challenge can be completed in less than 2 weeks (e.g. a few days) but we wanted to be mindful of your faculty obligations.

The submission should be sent via email to [jobfair@nordeus.com](mailto:jobfair@nordeus.com) with a link to your GitHub repository, or a zipped project. Email subject should be: **Backend Challenge**.

**Challenge is open until November 20, 2023** end of day. Good luck!