# The-Featured-Bugz Auction Docker Desktop Guide

## catalog-service

This service handles the creation and management of items in the company's product catalog.
It communicates with an associated MongoDB database and identifies items that need to be converted into active auctions, which is done in the Auction service.

## auction-service

This service is the heart of the auction process as it receives items to be turned into active auctions based on unique IDs.
It is connected to a cache service that holds the auctions and bids from the system users along with their unique IDs.
Placing a bid is not possible if the bid is too low or if a JWT token has not been received.
If a bid is approved, rabbitMQ ensures that it is sent to the Bid service with the user's ID obtained from the JWT token.

## authentication-service

This service is responsible for generating JWT tokens for user authentication.
When a user logs in with a username and password, validated by the user service, they will receive a JWT token.
This token is necessary to perform authorized HTTP calls in the system.

## user-service

This service manages user administration and is connected to its own MongoDB database,
which stores user data. When a user logs in, they receive a JWT token,
which can be used by the Auction service to confirm their logged-in status and retrieve the user's unique ID.

## bid-service

This service handles the auction history by receiving data via RabbitMQ.
The data represents the bidding history from ongoing and previous auctions,
which is stored in a MongoDB database.

## Link to yml file

## Step 1: Create an item in the catalog-service

To start an auction, you need to create an item in the catalog-service.
Make an HTTP `POST` request to the endpoint `http://localhost:5098/item/postItem` with the following `JSON-data`:

```
{
  "ItemID": 1,
  "ItemName": "Chair",
  "ItemDescription": "Cool chair",
  "ItemStartPrice": 100,
  "ItemCurrentBid": -1,
  "ItemSellerID": 1,
  "ItemStartDate": "2023-05-25T00:00:00Z",
  "ItemEndDate": "2023-05-30T00:00:00Z"
}
```

This will add the item to the MongoDB database.

## Step 2: Create auctions from catalog-service

After creating an item in the catalog-service, you can proceed to create auctions from the catalog.
This requires a `JSON-array af items`, containing one or more items. In this case, we have added only one item,
which was created in step 1. Make an HTTP `POST` request to the endpoint `http://localhost:5098/item/postItemsToAuction` with the following `JSON-data`:

```
[
  {
    "ItemID": 1,
    "ItemStartPrice": 100,
    "ItemEndDate": "2023-05-25T00:00:00Z"
  }
]
```

This will create the auction for the items and add it to the cache (redis service) using the auction-service.

## Step 3: Create a user in the User service

To place a bid, you first need to create a user in the User service.
Make an HTTP `POST` request to the endpoint `http://localhost:5081/user/postuser` with the following `JSON-data`:

```
{
  "UserID": 1,
  "UserName": "John",
  "UserPassword": "password123",
  "UserEmail": "john@example.com",
  "UserAddress": "123 Main Street"
}
```

This will create the auction for the items and add it to the cache (redis service) using the auction-service.

## Step 4: Receive a JWT token from the authentication-service

To place a bid, you need to have a JWT token. To obtain this, you need to log in and receive a token.
Make an HTTP `POST` request to the endpoint `http://localhost:5152/auth/login` with the following `JSON-data` containing the username and password:

```
{
  "UserName": "John",
  "UserPassword": "password123"
}
```

This will return a JSON response containing user information and a token:

```
{
  "userID": 1,
  "userName": "John",
  "userPassword": "password123",
  "userEmail": "john@example.com",
  "userAddress": "123 Main Street",
  "token": "<JWT TOKEN>"
}
```

## Step 5: Place a bid in the auction-service

With the JWT token received in step 4, you can now place a bid.
Make an HTTP `POST` request to the endpoint `http://localhost:5158/auction/PostAuctionBid` with following `JSON-data`:

```
{
  "bidID": 1,
  "auctionID": 1,
  "bidPrice": 150,
  "bidUserID": 2
}
```

Replace the values according to your bidding information. This will submit the bid to the Auction service, and if the bid is approved, it will be sent to the Bid service via RabbitMQ, where the Bid service worker will store it in a MongoDB database.