# Web Advanced -
# Technische Documentatie

Naam auteur:
Mustafa Demir – 542812
Naam docent: Evert Duipmans
Vak: Web Advanced
Klas: DHI2V.SA
Datum: 20-10-2024
Versie: 1

# Inhoudsopgave

# Inleiding

Ik heb een webapplicatie gemaakt waarbij het mogelijk is om veilingen van games te kunnen zien. Deze documentatie beschrijft hoe deze applicatie is ontwikkelt. De applicatie heeft een user-friendly homepagina waarbij het mogelijk is om alle veilingen te zien, te kunnen filteren en te kunnen zoeken op user-input. Zodra je op een veiling klikt, zie je details van de veiling waarbij je de foto, beschrijving en de biedingen ziet op die specifieke veiling. Je kunt niet een bod plaatsen zonder dat je bent ingelogd. Je kunt je registreren door op de ‘login’ knop te klikken en je gebruikersnaam en wachtwoord in te vullen. Als je nog geen account hebt, kun je op ‘registreer hier’ klikken en je registreren. Vervolgens wordt je na het registreren gelijk doorverwezen naar de veilingen pagina en is het mogelijk om een bod te plaatsen.

Op het beheerportaal van deze veilingen website worden alle games weergeven die worden geveild. Het is voor beheerders mogelijk om games toe te voegen, te bewerken of te verwijderen.

Deze applicatie is gebouwd met Node.js en Express voor de backend, met Svelte voor in het front-end. Data wordt bijgehouden d.m.v. arrays en er is geen database hiervoor gebruikt. Dit project is opgezet om mijn programmeervaardigheden verder te ontwikkelen en biedt een compleet systeem voor het beheren van veilingen, inclusief gebruikersauthenticatie en administratie functionaliteiten voor het beheren van de veilingen.

# REST API

## GET requests

| **GET** | **/games** | | |
|---|---|---|---|
| Bekijk alle games. Er kunnen filters toegepast worden op naam, genre en uitgeverij en op titel van de game gezocht worden. | | | |
| | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| | Naam | Query | Zoek op titel |
| | Genre | Query | Filter op genre |
| | Uitgeverij | Query | Filter op uitgeverij |
| | Prijs | Query | Filter op prijs |
| | Aantal | Query | Filter op maximale games die er getoond wil worden |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 200 | Games gevonden | |
| | 404 | Geen game(s) gevonden | |

| **GET** | **/games/{id}** | | |
|---|---|---|---|
| Bekijk een specifieke veiling. | | | |
| | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | {id}* | path | Zoek naar een game met ID |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 200 | Games gevonden | |
| | 404 | Geen game gevonden | |

| **GET** | **/games/{id}/bids** | | |
|---|---|---|---|
| Bekijk alle biedingen van een specifieke veiling. | | | |
| | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | {id}* | path | Zoek naar een game met ID |
| | | | |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 200 | Biedingen gevonden | |
| | 404 | Geen biedingen gevonden | |

| **GET** | /won-auctions | | |
|---|---|---|---|
| Bekijk alle als user alle gewonnen veilingen. | | | |
| | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | | | |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 200 | Gewonnen veilingen gevonden | |
| | 404 | Geen gewonnen veilingen gevonden | |

| **GET** | **/users/**admin | | |
|---|---|---|---|
| Bekijk alle veilingen en beheer deze in de admin paneel. | | | |
| | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | | | |
| | | | |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 200 | Geauthentiseerd: toegang verleend tot admin paneel. | |
| | 404 | Geen veilingen gevonden. | |
| | 401 | Niet geautoriseerd: toegang geweigerd zonder geldige token of admin rechten. | |

## POST requests

| **POST** | **/games** | | |
|---|---|---|---|
| Voeg een nieuwe veiling toe. | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | | body | { "gameId": "Integer", "title": "String", "description": "String", "publisher": "String", "category": "String", "startingPrice": "Integer", "auctionEndDate": "Date", "bids": "Array" } |
| | **Code** | **Description / example if successful** | |
| **Responses:** | | | |
| | 201 | Nieuwe game aangemaakt | |
| | 400 | Ongeldige input | |
| | | | |
| | | | |

| **POST** | **/users** | |
|---|---|---|
| Voeg een gebruiker toe | | |
| **Parameters:** | **Name** | **Type** |
| *Add a * to the name of required parameters.* | body* | {"userId": "Integer", "username": "String", "email": "String", "password": "String"} |
| **Responses:** | **Code** | **Description / example if successful** |
| | 200 | Succesvol ingelogd |
| | 401 | Ongeldige invoer |
| | 409 | Gebruiker bestaat al |

| POST | /games/{id}/bids | | |
|---|---|---|---|
| Voeg een bod toe op een specifieke veiling. | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | Auction{id}* | Path | Zoeken naar een specifieke veiling |
| | body | body | "bidId": "Integer", "bidAmount": "Integer", "bidTime": "DateTime", "userId": "Integer", "gameId": "Integer" |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 201 | Bod toegevoegd | |
| | 400 | Fout in de input van bod | |
| | 404 | Game niet gevonden | |
| | | | |

| POST | /users/login | |
|---|---|---|
| Log in op de veilingwebsite met een gebruikers naam en wachtwoord. | | |
| **Parameters:** | **Name** | **Type** |
| *Add a * to the name of required parameters.* | username* | String |
| | password* | String |
| **Responses:** | **Code** | **Description / example if successful** |
| | 200 | Succesvol ingelogd |
| | 401 | Wachtwoord onjuist |
| | 404 | Gebruiker niet gevonden |

## PUT requests

| **PUT** | **/games/{id}** | | |
|---|---|---|---|
| Bewerk de attributen van een specifieke veiling. | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | auction {id}* | Path | Zoek naar een veiling met ID |
| | | body | "gameId": "Integer", "title": "String", "description": "String", "publisher": "String", "category": "String", "startingPrice": "Integer", "auctionEndDate": "Date" |

## DELETE requests

| **DELETE** | **/games/{id}** | | |
|---|---|---|---|
| Verwijder een specifieke veiling. | | | |
| **Parameters:** | **Name** | **Type** | **Description** |
| *Add a * to the name of required parameters.* | veiling{id}* | path | Zoek naar een veiling met ID |
| **Responses:** | **Code** | **Description / example if successful** | |
| | 200 | Game is succesvol verwijderd | |
| | 404 | Game niet gevonden | |
| | | | |

# Implementatiekeuzes

## 1. Systeem Beschrijving

Ik heb mijn front-end gebouwd met Svelte, de backend met Node.js en Express die de REST standaarden volgen. Het systeem haalt de data op d.m.v. Arrays met als JSON formaat om veilingen op te kunnen halen en deze te beheren. De structuur is opgezet met de richtlijnen van 'Seperations of Concerns'. Dit zorgt ervoor dat alle modules worden gescheiden zoals controllers, routers en data directories.

## 2. Implementatie Keuzes

1. **Node.js en Express (Backend)**:
    - **Waarom gekozen**: Node.js is een framework van javascript waarmee je snelle en efficiënte servers mee kan opbouwen. Express zorgt ervoor dat je makkelijker kan omgaan met routers.
    - **Voordelen**: Asynchrone verwerking en goede prestaties.
2. **Svelte (Front-end):**
    - **Waarom gekozen**: Svelte is een javascript framework voor aan het front-end die ervoor zorgt om interactieve webpagina's te kunnen componenten. Het genereert uiterst efficiënte Javascript code en dit zorgt voor snelle laattijden.
    - **Voordelen**: Omdat Svelte gebruik maakt van componenten, is het mogelijk om je website makkelijk te onderhouden en aan te passen. Je scheidt elke onderdeel van je website op in delen en als er iets gebeurd, hoef je niet je hele code-base aan te passen.
3. **Structuur van de Applicatie**:
    - **Controllers**: controller bevatten de logica die de inkomende verzoeken verwerken en geven een juist antwoord terug.
    - **Routes**: De routes zijn de eindpunten van de gedefinieerde API's en deze wijzen naar de juiste functies in de controllers
    - **In-Memory Data Management**: Veilingen en gebruikersgegevens worden in arrays beheerd. Dit is voor een licht gewicht applicatie ideaal.
4. **Error Handling**:
    - **Waarom gekozen**: Een juiste afhandeling van het opvangen van fouten zorgt voor een robuuste applicatie.
    - **Implementatie**: Algemene error handlers en specifieke foutmeldingen per route aangemaakt die aantonen wat er mis is gegaan.