

# Ajdin's GameLibrary API Overview

## Introductie.

Deze api heb ik gemaakt als een project voor school. Het is bedoelt als een kleine game library te dienen waarin ik mijn eigen scoren kan toevoegen.

In deze document leg ik het in mijn worden uit maar als er nog hulp nodig is is er een swagger UI te beschikking.

</swagger/index.html>

## Request Formats

`/game`

Dit is mijn "main" api gedeelte waarin jij games kunt posten deleten updaten enz.

**GET** `/game`

Response:

Alles games samen met dev informatie en gamescores.

Formaat

```
[
  {
    "id": 1,
    "title": "Warframe",
    "developer": {
      "id": 1,
      "devName": "Digital Extremes"
    },
    "gameScores": {
      "id": 1,
      "generalScore": 0,
      "action": 0,
      "replayability": 0,
      "fun": 0
    }
  },
]
```

## POST /game

Hier mee kunnen er games worden toegevoegt tot de game library. De gamescores worden er automatisch aangemaakt en toegevoegd tot de db.

De format:

```
{
  "title": "string",
  "developer": {
    "devName": "string"
  },
  "gameScores": {
    "generalScore": 10,
    "action": 10,
    "replayability": 10,
    "fun": 10
  }
}
```

## PUT /game

Hier mee kan er een game dat al in de library bestaat worden geupdate.

```
{
  "id": 0,
  "title": "string",
  "developer": {
    "id": 0,
    "devName": "string"
  },
  "gameScores": {
    "id": 0,
    "generalScore": 10,
    "action": 10,
    "replayability": 10,
    "fun": 10
  }
}
```

## GET /ID

Vervang de "ID" met de game ID om informatie over die game ID te verkrijgen.

## Response example

```
{
  "game": {
    "id": 5,
    "title": "Call of duty",
    "developer": {
      "id": 4,
      "devName": "Activision"
    },
    "gameScores": {
      "id": 5,
      "generalScore": 0,
      "action": 0,
      "replayability": 0,
      "fun": 0
    }
  },
  "tags": [
    null,
    null
  ]
}
```

## DELETE /ID

Vervang de "ID" met de game ID om de game met het zelfde ID te verwijderen uit de database. De gamescore en taglink word mee verwijderd maar de developer data en tags data word nog steeds bewaard.

## GET /search?title=XX&dev=XX&tags=XX&sort=title&dir=asc&page=0&length=1

Dit kan gebruikt worden om naar specifieke titles, devs, tags te zoeken. Het heeft sorting, paging, dir en length mogelijkheden. Vervang de XX. Niet alles moet ingevuld worden om resultaten te krijgen.

Sorting:

- Title

- Score

Dir:

- Asc
- desc

Typische response is voor `/search?title=Overwatch`

```
[
  {
    "id": 3,
    "title": "Overwatch",
    "developer": {
      "id": 3,
      "devName": "Blizzard"
    },
    "gameScores": {
      "id": 3,
      "generalScore": 0,
      "action": 0,
      "replayability": 0,
      "fun": 0
    }
  }
]
```

## Tags categorie.

Hier worden alle tags bewaard. Die worden verbonden met taglink naar een game.

**GET** `/game/tag`

Response:

Alle tags dat bestaan op de DB. Samen met ID en tagname.

Example.

```
{
  "id": 1,
  "tagName": "Action"
},
```

## POST /game/tag

Met dit kan er tags worden geupdated. Dit word niet zovaak gebruikt.

Format

```
{
  "id": 0,
  "tagName": "string"
}
```

## DELETE /game/tag/remove/ID

Hiermee kan er een tag worden verwijderd uit de database met behulp van zijn ID. Alles tagslink die verbonden zijn met deze tag worden mee verwijderd.

## TagsLinks

TagsLink. In dit db worden er links gelegd tussed de tag en de games. Dit gebeurd via een veel op veel relatie.

## GET /game/tagslink

Responce:

Alle tagslink dat bestaan op de DB. Samen met de game en de tag.

## PUT /game/tagslink

Hiermee kan er tagslinks worden geupdaten naargelange nodig.

```
{
  "id": 0,
  "tag": {
    "id": 0,
    "tagName": "string"
  },
  "game": {
    "id": 0,
```

```
"title": "string",
"developer": {
  "id": 0,
  "devName": "string"
},
"gameScores": {
  "id": 0,
  "generalScore": 10,
  "action": 10,
  "replayability": 10,
  "fun": 10
}
}
```

**DELETE** /game/tagslink/remove/**ID**

Hiermee kan er een tagslink worden verwijderd uit de db.

**POST** /game/tagslink/**ID1**/**ID2**

Hierin kan er een link worden gezet tussen ID1 game id, en ID2 tag id.

**DELETE** /game/tag/remove/**ID**

Hiermee kan er een tag worden verwijderd uit de database. Alles tagslink die verbonden zijn met deze tag worden mee verwijderd.

## Developer categorie

Hier worden alle Developer informatie bewaard.

**GET** /dev

Response:

Alle developers dat bestaan op de DB. Samen met ID en tagname.

Example.

```
{  
  "id": 1,  
  "devName": "Digital Extremes"  
},
```

**POST** /dev

Met dit kan er tags worden geupdated. Dit word niet zovaak gebruikt.

Format

```
{  
  "id": 0,  
  "devName": "string"  
}
```

**DELETE** /dev/remove/**ID**

Hiermee word er een Developer uit de db verwijderd met behulp van zijn ID. Alle games die met de Developersdat geconnecteerd zijn gaan well null als hun dev hebben.