# Adatbázis dokumentáció:

## ’actings’ táblázat:

Adatkapcsolótáblázat az ’actors’ és ’games’ táblázathoz. Azt mutatja, hogy melyik játékokban milyen színészek játszottak, kiegészítve ezt a színész által eljátszott szereppel.

CREATE TABLE `actings` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `actorId` int(11) NOT NULL,

  `role` varchar(100) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. Az actorId színész azonosítója az actors táblázatból, ami egész szám, nem lehet üres, mert kötelező adt. A „role” a színész által eljátszott karakter, karakterlánc, lehet üres is, mert van, olyan eset mikor a színész fel van tűntetve, de nincs tisztázva a szerepe

ALTER TABLE `actings`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `actorId` (`actorId`);

Az id oszlopot elsődleges kulcs lesz, a gameId-hez és az actorId-hez hozzárendeljük, hogy kulcsok legyenek.

ALTER TABLE `actings`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0

ALTER TABLE `actings`

  ADD CONSTRAINT `actings\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `actings\_fk\_2` FOREIGN KEY (`actorId`) REFERENCES `actors` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. A gameId-hez és actorId-hez hozzárendeljük, hogy hogy idegen kulcsként működjenek, előbbinél a games táblázat id oszlopának, az utóbbi esetén az actors tábla id oszlopához. Mivel ezek az adatok feleslegesek, ha bármelyik rekordot töröljük, ezért megadjuk, hogy ezesetben ezek a sorok törlődjenek.

## ’actors’ táblázat:

Tárolja a játékokban megjelenő színészek nevét, illetve egy képet róluk.

CREATE TABLE `actors` (

  `id` int(11) NOT NULL,

  `firstName` varchar(100) DEFAULT NULL,

  `lastName` varchar(100) DEFAULT NULL,

  `profilePicture` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A firstName a színész keresztneve, ami karakterlánc, lehet üres. A lastName a színész vezetékneve, ami karakterlánc, lehet üres. A profilePicture a színész profilképének az elérési útvonala ami, karakterlánc, lehet üres.

ALTER TABLE `actors`

  ADD PRIMARY KEY (`id`);

Megadjuk az id-nek, hogy elsődleges kulcs legyen.

ALTER TABLE `actors`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje.

## ‘agerating’ táblázat:

Korhatár besorolás a játékokhoz, mivel több intézet foglalkozik ezzel, ezért egy játéknak lehet több, akár különböző besorolása több intézménytől.

CREATE TABLE `ageratings` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `rating` varchar(10) NOT NULL,

  `institution` varchar(4) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A rating a korhatár besorolás, ami karakterlánc, nem lehet üres, mert kötelező adt. Az institution a korhatár besorolást adó szervezet, ami karakterlánc, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat.

ALTER TABLE `ageratings`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`);

Megadjuk, hogy az id legyen elsődleges kulcs. A gameId-hez hozzá adjuk, hogy kulcs legyen.

ALTER TABLE `ageratings`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `ageratings`

  ADD CONSTRAINT `ageratings\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE;

COMMIT;

Az azonosítónak megadjuk, hogy az értékét automatikusan növelje. A gameid oszlopot pedig idegen kulcsé alakítjuk a games tábla id oszlopához és azt is megadjuk, hogy a rekord törölve legyen, ha a játékot töröljék, hiszen akkor nincs szükség erre az adatra.

## ’blacklistedtokens’ táblázat:

A kijelentkezett felhasználók token-jeit tartalmazza és annak dátumát és a felhasználó azonosítóját, abból célból, hogy ezt a token-t ne lehessen újra felhasználni. Az itt tárolt token-eket a backend rendszeresen törli, ami egy óra hosszánál régebbi, mert azok a token-ek alapértelmezetten érvénytelenek lesznek.

CREATE TABLE `blacklistedtokens` (

  `id` int(11) NOT NULL,

  `userId` int(11) NOT NULL,

  `token` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci NOT NULL,

  `date` datetime NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A userId a tokenhez tartozó felhasználó azonosítója a ’users’ táblából, egész szám, nem lehet üres, mert kötelező adat. A token mező magát a token-t tartalmazza, ami karakterlánc, nem lehet üres, kötelező adat. A date az a dátum mikor a token a listába került, dátum, nem lehet üres, mert kötelező adat.

ALTER TABLE `blacklistedtokens`

  ADD PRIMARY KEY (`id`),

  ADD KEY `userId` (`userId`);

Megadjuk, hogy az id legyen elsődleges kulcs. A userId-hoz hozzá rendeljük, hogy kulcs legyen.

ALTER TABLE `blacklistedtokens`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `blacklistedtokens`

  ADD CONSTRAINT `blacklistedtokens\_fk\_1` FOREIGN KEY (`userId`) REFERENCES `users` (`id`);

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. A megkötésben az userId oszlopát idegen kulcsá alakítjuk az user tábla id oszlopához.

## ’creations’ táblázat:

A ’creators’ és ’games’ táblázatot köti össze kiegészítve ezt azzal, hogy a fejlesztő milyen területen járult hozzá a játék elkészítéséhez.

CREATE TABLE `creations` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `creatorId` int(11) NOT NULL,

  `field` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, egész szám, nem lehet üres, mert kötelező adat. A creatorId a készítő azonosítója a ’creator’ táblázatból, egész szám, nem lehet üres, kötelező adat. A field azt mutatja, hogy a készítő milyen területen dolgozott az adott játékon, karakterlánc, lehet üres az olyan esetek miatt, ahol nem tisztázott a személy szerepe.

ALTER TABLE `creations`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `creatorId` (`creatorId`);

Megadjuk, hogy az azonosító legyen az elsődleges kulcs és kulcsokká alakítjuk a gameId és creatorId oszlopokat.

ALTER TABLE `creations`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `creations`

  ADD CONSTRAINT `creations\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `creations\_fk\_2` FOREIGN KEY (`creatorId`) REFERENCES `creators` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötéseket hozunk létre, hogy a gameId oszlop a games tábla id oszlopának az idegen kulcsa legyen. Hasonlóan a creatorId oszlop pedig a creators tábla id oszlop idegen kulcsa legyen.

## ’creators’ táblázat:

A játék fejlesztőit tartalmazza, és hogy milyen területen dolgoznak, bele értve a producereket, rendezőket és zeneszerzőket.

CREATE TABLE `creators` (

  `id` int(11) NOT NULL,

  `firstName` varchar(100) DEFAULT NULL,

  `lastName` varchar(100) DEFAULT NULL,

  `profilePicture` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A firstName a személy keresztneve, ami karakterlánc, lehet üres. A lastName a személy vezetékneve, ami karakterlánc, lehet üres. A profilePicture a profilkép elérési útvonala, ami karakterlánc, lehet üres.

ALTER TABLE `creators`

  ADD PRIMARY KEY (`id`);

ALTER TABLE `creators`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

COMMIT;

Az id oszlopot elsődleges kulcsra alakítjuk, majd megadjuk, hogy az értékét automatikusan növelje.

## ’favourites’ táblázat:

A felhasználók által kedvencekbe elmentett játékok.

CREATE TABLE `favourites` (

  `id` int(11) NOT NULL,

  `userId` int(11) NOT NULL,

  `gameId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A userId a felhasználó azonosítója, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója, ami egész szám, nem lehet üres, mert kötelező adat.

ALTER TABLE `favourites`

  ADD PRIMARY KEY (`id`),

  ADD KEY `userId` (`userId`),

  ADD KEY `gameId` (`gameId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A userId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `favourites`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `favourites`

  ADD CONSTRAINT `favourites\_fk\_1` FOREIGN KEY (`userId`) REFERENCES `users` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `favourites\_fk\_2` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk hogy a userId idegen kulcs legyen a users tábla id oszlopának, illetve a gameId is idegen kulcs legyen a games tábla id oszlopának.

## ’forumcomments’ táblázat:

A fórumon lévő posztok alatti kommenteket tartalmazza és fontosabb információkat róla.

CREATE TABLE `forumcomments` (

  `id` int(11) NOT NULL,

  `postId` int(11) NOT NULL,

  `userId` int(11) NOT NULL,

  'username' varchar(255), NOT NULL,

  `content` text NOT NULL,

  `creation` datetime NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A userId a komment írójának azonosítója, ami egész szám, nem lehet üres, mert kötelező adat. A username a komment írójának a neve, karakterlánc, nem lehet üres, mert kötelező adat. A content a komment tartalma, ami szöveg típusú, nem lehet üres, mert kötelező adat. A creation a komment létrehozásának ideje, ami típusa dátum idő, nem lehet üres, mert kötelező adat.

ALTER TABLE `forumcomments`

  ADD PRIMARY KEY (`id`),

  ADD KEY `postId` (`postId`),

  ADD KEY `userId` (`userId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A userId és postId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `forumcomments`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `forumcomments`

  ADD CONSTRAINT `forumcomments\_fk\_1` FOREIGN KEY (`postId`) REFERENCES `forumposts` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `forumcomments\_fk\_2` FOREIGN KEY (`userId`) REFERENCES `users` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk hogy a userId idegen kulcs legyen a users tábla id oszlopának, illetve a postId is idegen kulcs legyen a forumposts tábla id oszlopának.

## ’forumposts’ táblázat:

A fórumon lévő posztokat tartalmazza, ehhez hozzá járul a posztoló neve és pár infó a posztról.

CREATE TABLE `forumposts` (

  `id` int(11) NOT NULL,

  `userId` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `title` varchar(255) NOT NULL,

  `content` text NOT NULL,

  `story` tinyint(1) NOT NULL DEFAULT 0,

  `gameplay` tinyint(1) NOT NULL DEFAULT 0,

  `creation` datetime NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat.A userId a posztot létrehozó felhasználó azonosítója, egész szám, nem lehet üres, kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. A title a poszt címe, ami karakterlánc, nem lehet üres, mert kötelező adat. A content a poszt tartalma, ami szöveg típusú, nem lehet üres, mert kötelező adat. A story az mutatja, hogy a poszt a játék történetéhez kapcsolódik-e, ami logikai, nem lehet üres, alapvetően hamis. A gameplay azt mutatja, hogy poszt az adott játék játékmenetéhez kapcsolódik-e, ami logikai, nem lehet üres, alapvetően hamis. A creation a poszt létrehozásának ideje, ami típúsa dátum idő, nem lehet üres, kötelező adat.

ALTER TABLE `forumposts`

  ADD PRIMARY KEY (`id`),

  ADD KEY `userId` (`userId`),

  ADD KEY `gameId` (`gameId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A userId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `forumposts`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

ALTER TABLE `forumposts`

  ADD CONSTRAINT `forumposts\_fk\_1` FOREIGN KEY (`userId`) REFERENCES `users` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `forumposts\_fk\_2` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk hogy a userId idegen kulcs legyen a users tábla id oszlopának, illetve a gameId is idegen kulcs ami legyen a games tábla id oszlopának.

## ’gamepictures’ táblázat:

Ez a tábla a játékokhoz tartozó képeknek az elérési útját tárolják játék betekintésére szánt képeket.

CREATE TABLE `gamepictures` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `url` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblából, ami egész szám, nem lehet üres, mert kötelező adat. Az url a kép elérési útvonala, ami karakterlánc, nem lehet üres, mert kötelező adat.

ALTER TABLE `gamepictures`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A userId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `gamepictures`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

ALTER TABLE `gamepictures`

  ADD CONSTRAINT `gamepictures\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának.

## ’games’ táblázat:

Ez a fő táblázat, itt tárolódnak a játék egyedi adatai, amik tovább nem bonthatók.

CREATE TABLE `games` (

  `id` int(11) NOT NULL,

  `gameTitle` varchar(255) NOT NULL,

  `altGameTitle` varchar(255) DEFAULT NULL,

  `description` text NOT NULL,

  `release` date NOT NULL,

  `boxart` varchar(255) DEFAULT NULL,

  `promoArt` varchar(255) DEFAULT NULL,

  `controllerSupport` int(11) DEFAULT NULL,

  `crossplatform` int(11) DEFAULT NULL,

  `crossPlatformException` text DEFAULT NULL,

  `steamdeckComp` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameTitle a játék neve, ami karakterlánc, nem lehet üres, mert kötelező adat. Az altGameTitle a játék alternatív neve, ami jöhet onnan, hogy más régióban más nevet vagy számozást kapott, illetve egy kisegítő, ha az eredeti neve megegyezik egy másik játék nevével, karakterlánc típusú, lehet üres. A description A Játék rövid leírása, ami karakterlánc, nem lehet üres, mert kötelező adat. A release a játék legelső megjelenési ideje, ami dátum típusú, nem lehet üres, mert kötelező adat. A boxart a játék borítóképének az elérési útvonala, ami karakterlánc, lehet üres. A promoart a játékhoz tartozó nagyobb méretű promóciós képe, ami karakterlánc, lehet üres. A controllerSupport azt mutatja, hogy támogatja egy kontroller használatot számítógépen, egész szám, ami ezeket az eseteket jelöli: 0, vagyis nincs natív kontroller támogatás. 1, vagyis csak Xbox kontroller támogatása van. 2, vagyis Playstation kontroller esetén nincs teljes támogatás, ami során a kontroller működik, de nem a megfelelő gombkiosztás jelenik meg. 3, vagyis Xbox és Playstation kontrollert egyaránt támogat. A crossplatform azt mutatja meg hogy támogatja e, hogy a játékokat különböző platformokon egymással tudjanak-e játszani, egész szám, ami ezeket az eseteket tárolja: 0, vagyis nincs crossplatfrom. 1, vagyis csak konzolok között. 2, vagyis konzolok és számítógépek között egyaránt. A crossPlatformException az elöbbi esetet bővití ki, ha szükséges, a félreértéseket elkerülve, karakterlánc típusú, lehet üres. A steamdeckComp azt mutatja, hogy kompatibilis-e a steamdeck kézi konzollal, egész szám, ami ezeket a lehetőségeket tartalmazza: 0, vagyis nem kompatibilis. 1, vagyis játszható, de nem támogatott. 2, vagyis támogatott. Lehet üres ami azt jelenti hogy a steamdeck-en nem elérhető a játék.

ALTER TABLE `games`

  ADD PRIMARY KEY (`id`);

ALTER TABLE `games`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

COMMIT;

A games táblának az id oszlopát elsődleges kulcsnak adjuk meg. Majd megadjuk, hogy értékét automatikusan növelje.

## ’gamesawards’ táblázat:

Összekötő tábla a games és awards tábla között, ami a díjakat a játékokkal összeköti, hozzáadva hogy melyik évben jelölték a díjra és hogy megnyerte-e.

CREATE TABLE `gamesawards` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `awardId` int(11) NOT NULL,

  `year` int(11) NOT NULL,

  `result` tinyint(1) DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblából, ami egész szám, nem lehet üres, mert kötelező adat. Az awardId a díj azonosítója az ’awards’ táblából, ami egész szám, nem lehet üres, mert kötelező adat. A year a díjazás éve, ami egész szám, nem lehet üres, mert kötelező adat. A result a díjazás eredménye, ami logikai, ha igaz, akkor a játék megnyerte a díjat, ha hamis, akkor csak jelölve volt, alapértelmezetten hamis.

ALTER TABLE `gamesawards`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `awardId` (`awardId`);

A games táblának az id oszlopát elsődleges kulcsnak adjuk meg. A awardId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `gamesawards`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `gamesawards`

  ADD CONSTRAINT `gamesawards\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `gamesawards\_fk\_2` FOREIGN KEY (`awardId`) REFERENCES `awards` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az awardId legyen az award tábla id oszlopának az idegen kulcsa.

## ’gameslanguages’ táblázat:

Összekötő tábla a ’games’ és ’languages’ között, hozzá adva hogy a játékhoz tartozik e szinkron az adott nyelven.

CREATE TABLE `gameslanguages` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `languageId` int(11) NOT NULL,

  `dub` tinyint(1) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. A languageId a nyelv azonosítója a ’languages’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. A dub azt mutatja hogy tartozik-e a játékhoz szinkron is, ami logikai, ha igaz, akkor a játék az adott nyelven szinkronizálva is van, ha hamis, akkor csak felirat érhető el, nem lehet üres, kötelező adat, alapvetően hamis.

ALTER TABLE `gameslanguages`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `languageId` (`languageId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A languageId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `gameslanguages`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=20;

ALTER TABLE `gameslanguages`

  ADD CONSTRAINT `gameslanguages\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `gameslanguages\_fk\_2` FOREIGN KEY (`languageId`) REFERENCES `languages` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az languageId legyen az languages tábla id oszlopának az idegen kulcsa.

## ’gameonlineplatforms’ táblázat:

Összekötő tábla a ’games’ és az ’onlineplatforms’ között, nem tartalmaz további információt.

CREATE TABLE `gamesonlineplatforms` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `platformId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. platformId a platform azonosítója a ’onlineplatforms’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat.

ALTER TABLE `gamesonlineplatforms`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `platformId` (`platformId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A platformId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `gamesonlineplatforms`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `gamesonlineplatforms`

  ADD CONSTRAINT `gamesonlineplatforms\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `gamesonlineplatforms\_fk\_2` FOREIGN KEY (`platformId`) REFERENCES `onlineplatforms` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az platformId legyen az onlineplatforms tábla id oszlopának az idegen kulcsa.

## ’gamesplatforms’ táblázat:

Összekötő tábla a ’games’ és az ’platforms’ között, további információt nem tartalmaz.

CREATE TABLE `gamesplatforms` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `platformId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. A platformId a platform azonosítója a ’platforms’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat.

ALTER TABLE `gamesplatforms`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `platformId` (`platformId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A platformId és gameId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `gamesplatforms`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

ALTER TABLE `gamesplatforms`

  ADD CONSTRAINT `platforms\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `platforms\_fk\_2` FOREIGN KEY (`platformId`) REFERENCES `platforms` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az platformId legyen az platforms tábla id oszlopának az idegen kulcsa.

## ’gametags’ táblázat:

Összekötő tábla a ’games’ és az ’tags’ között, további információt nem tartalmaz

CREATE TABLE `gamestags` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `tagId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;.

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId – A játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. A tagId a tag azonosítója a ’tags’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat.

ALTER TABLE `gamestags`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `tagId` (`tagId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A platformId és tagId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `gamestags`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `gamestags`

  ADD CONSTRAINT `tags\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `tags\_fk\_2` FOREIGN KEY (`tagId`) REFERENCES `tags` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az tagId legyen az tags tábla id oszlopának az idegen kulcsa.

## ’languages’ táblázat:

Ez a tábla a világ nyelveit tartalmazza, amik játékok esetén előfordulnak.

CREATE TABLE `languages` (

  `id` int(11) NOT NULL,

  `language` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A language a nyelv neve, ami karakterlánc, nem lehet üres, mert kötelező adat.

ALTER TABLE `languages`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `unique\_language` (`language`);

ALTER TABLE `languages`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

COMMIT;

Az id-nek megadjuk, hogy elsődleges kulcs és hogy a language egyedi adat legyen, szóval két megegyező nevű rekord ne legyen lehetséges. Végül megadjuk, hogy az id az értékét automatikusan növelje.

## ‘onlineplatforms’ táblázat:

A játékokat árusító platformok listája.

CREATE TABLE `onlineplatforms` (

  `id` int(11) NOT NULL,

  `platform` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A platform az online platform neve, ami karakterlánc, nem lehet üres, mert kötelező adat.

ALTER TABLE `onlineplatforms`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `unique\_onlineplatform` (`platform`);

ALTER TABLE `onlineplatforms`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

COMMIT;

Az id-nek megadjuk, hogy elsődleges kulcs és hogy a platform egyedi adat legyen, szóval két megegyező nevű rekord ne legyen lehetséges. Végül megadjuk, hogy az id az értékét automatikusan növelje.

## ’pcspecs’ táblázat:

A játék minimum és ajánlott gépigényét tárolja. Az eszközök általános rövidítéseit használja, és ami a minimumhuz tartozik, ott szerepel a „min” szócska.

CREATE TABLE `pcspecs` (

  `minop` varchar(20) NOT NULL,

  `mincpu` varchar(100) NOT NULL,

  `minram` varchar(10) NOT NULL,

  `mingpu` varchar(100) NOT NULL,

  `mindirectx` varchar(20) DEFAULT NULL,

  `cpu` varchar(100) DEFAULT NULL,

  `ram` varchar(10) DEFAULT NULL,

  `gpu` varchar(100) DEFAULT NULL,

  `directx` varchar(20) NOT NULL,

  `op` varchar(20) DEFAULT NULL,

  `storage` varchar(10) NOT NULL,

  `sidenote` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_hungarian\_ci DEFAULT NULL,

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

id – Azonosító, egész szám, nem lehet üres, kötelező adat

minop – A minimum szükséglet operációsrendszer terén, karakterlánc, nem lehet üres, kötelező adat

mincpu - A minimum szükséglet a processzor terén, karakterlánc, nem lehet üres, kötelező adat

minram - A minimum szükséglet a memória terén, karakterlánc, nem lehet üres, kötelező adat

mingpu - A minimum szükséglet a videókártya terén, karakterlánc, nem lehet üres, kötelező adat

mindirectx - A minimum szükséglet a directx verzió terén, karakterlánc, nem lehet üres, kötelező adat

cpu – Az ajánlott szükséglet a processzor terén, karakterlánc

ram - Az ajánlott szükséglet a memória terén, karakterlánc

gpu - Az ajánlott szükséglet a videókártya terén, karakterlánc

directx - Az ajánlott szükséglet a directx verzió terén, karakterlánc

op - Az ajánlott szükséglet operációsrendszer terén, karakterlánc

storage – Szükséges tárhely, karakterlánc, nem lehet üres, kötelező adat

sidenote – Egyéb megjegyzések, karakterlánc

id – Azonosító, egész szám

gameId – A játék azonosítója a ’games’ táblából, egész szám

ALTER TABLE `pcspecs`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId\_fk\_1` (`gameId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A gameId oszlopot kulcsra alakítjuk.

ALTER TABLE `pcspecs`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

ALTER TABLE `pcspecs`

  ADD CONSTRAINT `pcspecs\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának.

## ’platforms’ táblázat:

Ez a táblázat tartalmazza játék platformokat/konzolokat tartalmaz.

CREATE TABLE `platforms` (

  `id` int(11) NOT NULL,

  `platform` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosító a ’games’ táblából, ami egész szám, nem lehet üres, mert kötelező adat. A platform mező a platform, avagy konzol neve, ami karakterlánc, nem lehet üres, mert kötelező adat.

ALTER TABLE `platforms`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `unique\_platform` (`platform`);

ALTER TABLE `platforms`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

COMMIT;

Az id-nek megadjuk, hogy elsődleges kulcs és hogy a platform egyedi adat legyen, szóval két megegyező nevű rekord ne legyen lehetséges. Végül megadjuk, hogy az id az értékét automatikusan növelje.

## ’rating’ táblázat:

A játékok értékeléseit tartalmazza, amit a regisztrált felhasználók adtak.

CREATE TABLE `ratings` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `positive` tinyint(1) NOT NULL DEFAULT 0,

  `userId` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója a ’games’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. A positive a felhasználó értékelése, ami logikai, ahol a hamis azt jelzi, hogy a felhasználó negatívan értékelte, az igaz meg hogy pozitívan értékelte, nem lehet üres, mert kötelező adat. A userId a felhasználó azonosítója a ’users’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat.

ALTER TABLE `ratings`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `userId` (`userId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A gameId és userId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `ratings`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=10;

ALTER TABLE `ratings`

  ADD CONSTRAINT `ratings\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `ratings\_fk\_2` FOREIGN KEY (`userId`) REFERENCES `users` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az userId legyen az users tábla id oszlopának az idegen kulcsa.

## ’studios’ táblázat:

A játék fejlesztő és kiadó stúdiókat tárolja és ezeknek a logóját.

CREATE TABLE `studios` (

  `id` int(11) NOT NULL,

  `name` varchar(255) NOT NULL,

  `logo` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A name a cég neve, ami karakterlánc, nem lehet üres, mert kötelező adat. A logo a stúdió logójának az elérési útvonala, ami karakterlánc, nem lehet üres, mert kötelező adat.

ALTER TABLE `studios`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `unique\_name` (`name`);

ALTER TABLE `studios`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=0;

COMMIT;

Az id-nek megadjuk, hogy elsődleges kulcs és hogy a name egyedi adat legyen, szóval két megegyező nevű rekord ne legyen lehetséges. Végül megadjuk, hogy az id az értékét automatikusan növelje.

## ’studiosgames’ táblázat:

Összeköti a ’games’ és a ’studios’ táblázatot azonosító alapján, kiegészítve ezt azzal, hogy az adott játék esetén fejlesztő és/vagy kiadói szerepet töltöttek be.

CREATE TABLE `studiosgames` (

  `id` int(11) NOT NULL,

  `gameId` int(11) NOT NULL,

  `studioId` int(11) NOT NULL,

  `isDeveloper` tinyint(1) NOT NULL DEFAULT 0,

  `isPublisher` tinyint(1) NOT NULL DEFAULT 0

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója, a ’games’ táblázatból kerül ide az ’id’ oszlop, ami egész szám, nem lehet üres, mert kötelező adat. A studioId a stúdió azonosítója a ’studios’ táblázatból, ami egész szám, nem lehet üres, mert kötelező adat. Az isDeveloper azt mutatja, hogy a stúdió fejlesztette-e a játékot, ami logikai, nem lehet üres, mert kötelező adat, alapvetően hamis. Az isPublisher azt mutatja hogy a stúdió kiadta-e a játékot, ami logikai, nem lehet üres, mert kötelező adat.

ALTER TABLE `studiosgames`

  ADD PRIMARY KEY (`id`),

  ADD KEY `gameId` (`gameId`),

  ADD KEY `studioId` (`studioId`);

Az id-nek megadjuk, hogy elsődleges kulcs. A studioId és tagId oszlopokat kulcsokká alakítjuk.

ALTER TABLE `studiosgames`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=9;

ALTER TABLE `studiosgames`

  ADD CONSTRAINT `studiosgames\_fk\_1` FOREIGN KEY (`gameId`) REFERENCES `games` (`id`) ON DELETE CASCADE,

  ADD CONSTRAINT `studiosgames\_fk\_2` FOREIGN KEY (`studioId`) REFERENCES `studios` (`id`) ON DELETE CASCADE;

COMMIT;

Az id-nek megadjuk, hogy az értékét automatikusan növelje. Megkötésekben megadjuk, hogy a gameId idegen kulcs legyen a games tábla id oszlopának, illetve az studioId legyen az users tábla id oszlopának az idegen kulcsa.

## ’tags’ táblázat:

A játékokhoz tartozó címkék, avagy ’tag’-ek, ami a játékhoz illenek.

CREATE TABLE `tags` (

  `id` int(11) NOT NULL,

  `tag` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A gameId a játék azonosítója, ami egész szám, nem lehet üres, mert kötelező adat. A tag a játék címkéje, ami stílust, illetve műfajt jelöl, karakterlánc típusú, nem lehet üres, mert kötelező adat.

ALTER TABLE `tags`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `unique\_tag` (`tag`);

ALTER TABLE `tags`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=22;

COMMIT;

Az id-nek megadjuk, hogy elsődleges kulcs és hogy a tag egyedi adat legyen, szóval két megegyező nevű rekord ne legyen lehetséges. Végül megadjuk, hogy az id az értékét automatikusan növelje.

## ’users’ táblázat:

Információk a felhasználókról.

CREATE TABLE `users` (

  `id` int(11) NOT NULL,

  `username` varchar(255) NOT NULL,

  `password` varchar(255) NOT NULL,

  `email` varchar(255) NOT NULL,

  `admin` tinyint(1) NOT NULL,

  `creation` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

Az id az azonosító, ami egész szám, nem lehet üres, mert kötelező adat. A username a felhasználó neve, ami karakterlánc, nem lehet üres, mert kötelező adat. A password a felhasználó jelszava titkosítva, ami karakterlánc, nem lehet üres, mert kötelező adat. Az email mező a felhasználó email címe, ami karakterlánc, nem lehet üres, mert kötelező adat. Az admin azt mutatja, hogy a felhasználó rendelkezik-e admin jogokkal, ami logikai, nem lehet üres, mert kötelező adat. A creation a profil létrehozásának időpontja, ami dátum típusú, nem lehet üres, mert kötelező adat.

ALTER TABLE `users`

  ADD PRIMARY KEY (`id`),

  ADD UNIQUE KEY `unique\_email` (`email`),

  ADD UNIQUE KEY `unique\_name` (`username`);

Az id-nek megadjuk, hogy elsődleges kulcs. Az email és username oszlopok egyedinek állítjuk be, hogy ne engedjen két ugyanolyan rekordot a táblába.

ALTER TABLE `users`

  MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

COMMIT;

Megadjuk, hogy az id az értékét automatikusan növelje.

# Backend dokumentáció

# Eszközök

A backend Node.js segítségével lett létrehozva az Express keretrendszerrel, továbbá a Sequelize ORM-et használjuk az adatbázis eléréséhez, ehhez telepítve van mellé a Mariadb csomag is. A Cors csomag lehetővé teszi, hogy kapcsolatot alakítsunk ki a frontend-hez.

A Nodemon csomagot használjuk, hogy megkönnyítse a fejlesztési oldalt azzal, hogy a szerver minden változtatás mentése után újra induljon, illetve biztosít tiszta kilépést hiba esetén. Az oldal token-eket alkalmaz a felhasználók kezelésére, ehhez a Jose könyvtár JWE token-eket alkalmazunk, ami az alapvető JWT-hez képest további titkosítással van ellátva. A felhasználók jelszavának titkosításához a Bcrypt csomagot használjuk.

## Mappa struktúra

main/backend

├── node\_modules

├── src

│ ├── config

│ ├── controllers

│ ├── models

│ ├── routes

│ ├── utilities

├── app.js

├── package-lock.json

├── package.json

A backend mappában három fájl látható, a package.json és a package-lock.json olyan fájlok amik egy Node.js projekt megkezdésekor létre jönnek és meta adatokat tartalmaznak, illetve a függőségeket, az app.js a backend fő fájlja ami egyesíti az előző programrészeket és ez az a fájl amelyet futtatnunk kell hogy elinduljon a backend szerver.

Az src mappa további öt mappát tartalmaz, a config mappa egy fájlt tartalmaz, amiben az adatbázis konfigurációját tartalmazza. A controllers mappában lévő fájlok kezelik a felhasználó által kezdeményezett kéréseket. A models mappa az adatbázis tábláinak a Sequelize modelljeit tartalmazza, minden tábla egy külön fájl, illetve tartalmaz egy index.js fájlt, ahol a modellek kapcsolatai vannak meghatározva, illetve a szinkronizálja a modelleket a tényleges adatbázissal. A routes mappa a controllers mappában lévő metódusokat társítja egy URL-hez és http metódussal, majd ezt exportálva fel lesz használva a fent említett app.js-ben. Az utilities olyan metódusokat tartalmaz, amelyek a projektben több helyen felvannak használva, így csökkentve a redundanciát és átláthatóbb lesz a kód, a különböző fájlok valamilyen csomagra specifikusak, például a bcrypt.methods.js a bcrypt csomaghoz tartozó műveleteket és változókat tartalmazza.

## HTTP kérések

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Metódus | Magyarázat | Státusz kódok |
| / | GET | GET: A főoldalon megjelennek a hamarosan megjelenő játékok, illetve további játékok, mint a legújabbak, illetve műfajokba sorolva. | 200 – Adatok sikeresen lekérdezve  500 – Ismeretlen hiba |
| PUT | A főoldalon egy működő kereső mező, ami a játékokat tud keresni az eredeti, illetve alternatív nevével is, a keresendő szöveget a frontend elküldi, amit a backend feldolgozz. | 200 – Keresés eredményei lekérve  500 – Ismeretlen hiba |
| POST | A főoldalon a bejelentkezett felhasználó ki tudd ezáltal jelentkezni. | 200 – Sikeres kijelentkezés  500 – Ismeretlen hiba |
| /game/:gameId | GET | A konkrét játék adatait kérjük le és jelenítjük meg. | 200 – Adatok sikeresen lekérdezve  404 – Az URL-ben megadott azonosítóhoz nincs játék |
| PUT | A felhasználó játék értékelését menti el, magát az értékelést a frontend küldi. | 201 – Az értékelés vagy a kedvencekbe tétel sikeresen lementve  403 – Nem engedélyezett a játék értékelés, mert a felhasználó nincs bejelentkezve  500 – Ismeretlen hiba |
| POST | A felhasználó kedvencekbe helyezheti a kívánt játékokat. | 201 – Az értékelés vagy a kedvencekbe tétel sikeresen lementve  403 – Nem engedélyezett a játék kedvencekbe tétele, mert a felhasználó nincs bejelentkezve  500 – Ismeretlen hiba |
| /registration | POST | Regisztráció az oldalhoz. | 201 – Az adatbázisba sikeresen feljuttattuk az új felhasználó adatokat  400 – Hiba a regisztrációs adatokkal  409 – Már van az adatbázisban olyan felhasználó, akinek ilyen felhasználóneve vagy email címe van.  500 – Ismeretlen hiba |
| /login | POST | Bejelentkezés az oldalhoz. | 201 – Sikeres bejelentkezés, ehhez a felhasználó kap egy token-t  400 – Hiányzik valamilyen adat vagy üres  404 – Nem létezik ilyen adatokkal felhasználó  500 – Ismeretlen hiba |
| /studio/:studioId | GET | A hozzá kapcsolható játékokat jeleníti meg, illetve a stúdió logóját. | 200 – Adatok sikeresen lekérdezve  404 – Az URL-ben megadott azonosítóhoz nincs studió  500 – Ismeretlen hiba |
| /actor/:actorId | GET | A hozzá kapcsolható játékokat jeleníti meg, illetve a stúdió logóját. | 200 – Adatok sikeresen lekérdezve  404 – Az URL-ben megadott azonosítóhoz nincs színész  500 – Ismeretlen hiba |
| /user/:user | GET | A felhasználó adatait jeleníti. | 200 – Adatok sikeresen lekérdezve  404 – Az URL-ben megadott azonosítóhoz nincs felhasználó  500 – Ismeretlen hiba |
| POST | A felhasználó megtudja változtatni a jelszavát, az új jelszót a frontend küldi el, a validációt a backend végzi. | 201 – Jelszó sikeresen megváltoztatva  400 – Probléma az új jelszó megadása során  500 – Ismeretlen hiba |
| user/:user/favourites | GET | A felhasználó a kedvencekben elmentett játékait mutatja meg. | 200 – Adatok sikeresen lekérve  500 – Internal Server Error |
| POST | A felhasználó eltávolíthatja a kedvencekben mentett játékait. | 200 – Adatok sikeres törölve  500 – Internal Server Error |
| /preforum | - | - | - |
| /forum | GET | Alapvetően a legfrissebb fórum posztokat mutatja „generals” -ban, és lehet konkrét játékok alapján is keresni. | 200 – Sikeres lekérdezés  500 – Ismeretlen hiba |
| DELETE | Az admin jogosultságú felhasználó bármelyik posztokat törölni tudja. | 200 – Sikeres törlés  500 – Ismeretlen hiba |
| /forum/creation | POST | A felhasználó posztját menti el | 201 – Poszt sikeresen létrehozva  400 – Nincs megadva minden szükséges adat  401 – Ha nem admin akkor nincs erre jogosultsága  500 – Ismeretlen hiba |
| /forum/:postId | GET | A posztokat és kommenteket jeleníti meg. | 200 – Sikeres lekérdezés  404 – Nincs ilyen azonosítójú poszt  500 – Ismeretlen hiba |
| POST | A felhasználó kommentjét menti el. | 201 – A komment sikeresen létrejött  401 – Nem engedélyezett a komment közéttele, mert a felhasználó nincs bejelentkezve vagy hibás a token  500 – Ismeretlen hiba |
| DELETE | Az admin jogosultságú felhasználó képes bármely felhasználó hozzászólását törölni. | 200 – A komment sikeresen törölve  401 – Nem engedélyezett a komment törlése, mert a felhasználó nem admin  500 – Ismeretlen hiba |