Anggota kelompok:

217116641 - Paqsy Jalasukma

217116642 - Patrick Tjendika

217116656 - Stefan Ruben A.

217116664 - Vincent Chung

**Dokumentasi Endpoint**

**#User**

|  |  |  |
| --- | --- | --- |
| Activity | Verb/Noun Mapping | Participant |
| Register User | POST/user/register | Free & premium user |
| Login User | POST/user/login | Free & premium user |
| Update User | PUT/user/UpdateUser/:username | Free & premium user |
| Delete User | DELETE/user/:username | Free & premium user |
| Upgrade User | PUT/user/upgrade | Free & Premium user |
| Get User | GET/user/:keyword | Free & Premium user |

1. Register User
2. User memasukan username,password,nama,nomor telepon,dan foto profil
3. Sistem akan mengecek apakah ada username yang sama dengan yang ingin di daftarkan jika tidak ada maka user akan terdaftar kedalam sistem dengan data-data yang sudah diinputkan

Detail Endpoint:

/user/register (POST)

Body:

* username : Username dari pengguna
* password : Password pengguna
* name : Nama pengguna
* phone\_number : Nomor telepon pengguna
* profile\_picture : Profile picture pengguna

1. Login User
2. User memasukan username dan password
3. Sistem lalu akan mengecek apakah ada data user dengan status aktif yang memiliki username dan password yang sesuai dengan inputan user,jika ada maka sistem akan mengembalikan token JWT

Detail Endpoint :

/user/login (POST)

Body:

* username : Username dari pengguna
* password : Password pengguna

1. Update User
2. User memasukan token JWT
3. User memasukan username di parameter
4. User memasukan data-data yang ingin dirubah
5. Sistem akan mengecek token dan data user yang didapat dari username apakah sesuai atau tidak, dan jika sesuai maka data user akan di update dengan data-data inputan dari user yang baru

Detail Endpoint:

/user/updateUser/:username (PUT)

Params:

* username : Username pengguna

Header:

* x-auth-token : JWT token yang didapat saat login

Body:

* password : Password baru dari pengguna
* name : Nama baru dari pengguna
* phone\_number : Nomor telepon baru dari pengguna

1. Delete User
2. User memasukan Token JWT
3. User memasukan username di parameter
4. Sistem akan mengecek token dan data user yang didapat dari username apakah sesuai atau tidak, dan jika sesuai maka akun user akan di nonaktifkan dari sistem

Detail Endpoint :

/user/:username (DELETE)

Params :

* username : Username dari pengguna

Header :

* x-auth-token : JWT Token yang didapat saat login

1. Upgrade User
2. User memasukan token JWT
3. Sistem akan mengecek token yang dimasukan dengan data user di database sistem jika ditemukan maka akan dicek apakah ada id transaksi, jika ada maka sistem akan menampilkan status transaksi,jika tidak ada maka sistem akan memberikan virtual account BCA dan id transaksi akan tersimpan di data user

Detail Endpoint :

/user/upgrade (PUT)

Header :

* x-auth-token : JWT Token yang didapat saat login

1. Get User
2. User memasukan keyword username yang ingin dicari
3. Sistem akan mencari data user dari username yang mengandung huruf atau kata dari keyword yang diberikan.
4. Jika ditemukan maka akan ditampilkan data user

Detail Endpoint :

/user/:keyword (GET)

Params :

* keyword : Keyword yang digunakan untuk mencari daftar username yang memiliki huruf atau kata seperti keyword

**#Book**

|  |  |  |
| --- | --- | --- |
| Activity | Verb/Noun Mapping | Participant |
| Get book detail by id | GET/search/id/:id\_book | Free & premium user |
| Search book by query | GET/search/ | Free & premium user |

1. Get book detail by id
2. User memasukkan token jwt.
3. User memasukkan id buku goodreads pada parameter.
4. Sistem akan mengecek apakah token jwt valid.
5. Jika token jwt valid dan id buku ditemukan, maka sistem akan menampilkan detail buku dari id buku tersebut.

Detail endpoint:

/search/id/:id\_book (GET)

Header:

* x-auth-token : JWT token yang didapatkan saat login

1. Search book by query
   1. User memasukkan token jwt.
   2. User memasukkan query berdasarkan judul, penulis buku dan nomor halaman (default=1).
   3. Sistem akan mengecek apakah token jwt valid.
   4. Jika token jwt valid, maka sistem akan menampilkan daftar buku beserta informasi berdasarkan query dan nomor halaman.

Detail endpoint:

/search/ (GET)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Query:

* query: Query berdasarkan judul dan penulis buku
* page: Nomor halaman yang akan ditampilkan (default=1)

**#Review**

|  |  |  |
| --- | --- | --- |
| Activity | Verb/Noun Mapping | Participant |
| Create review | POST/review/:id\_book | Free & premium user |
| Get review | GET/review/ | Free & premium user |
| Update review | PUT/review/:id\_book | Free & premium user |
| Delete review | DELETE/review/:id\_book | Free & premium user |

1. Create review
2. User memasukkan token jwt.
3. User memasukkan id buku goodreads pada parameter dan memasukkan rating (1-5) dan comment pada body.
4. Sistem akan mengecek apakah token jwt valid.
5. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter dan body yang diperlukan sudah terpenuhi.
6. Jika semua kriteri sudah terpenuhi, maka sistem akan mengecek apakah buku dengan id tersebut ditemukan.
7. Jika buku ditemukan, maka sistem akan mengecek apakah user belum pernah mereview buku tersebut.
8. Jika user belum pernah mereview buku tersebut, maka sistem akan memasukkan data review ke database.

Detail endpoint:

/review/:id\_book (POST)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Body:

* rating: Rating buku (1-5)
* comment: Isi comment

1. Get review
   1. User memasukkan token jwt.
   2. User memasukkan query berdasarkan id buku, username (pembuat review), dan rating (1-5).
   3. Sistem akan mengecek apakah token jwt valid.
   4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter yang diperlukan sudah terpenuhi.
   5. Jika kriteria sudah terpenuhi, maka sistem akan menampilkan data review sesuai dengan query.

Detail endpoint:

/review/ (GET)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Query:

* id\_book: Query berdasarkan id buku dari goodreads (Optional)
* username: Query berdasarkan username (pembuat review) (Optional)
* rating: Query berdasarkan rating buku (Optional)

1. Update review
   1. User memasukkan token jwt.
   2. User memasukkan id buku goodreads yang akan diperbarui pada parameter dan memasukkan rating (1-5) dan comment pada body.
   3. Sistem akan mengecek apakah token jwt valid.
   4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter dan body yang diperlukan sudah terpenuhi.
   5. Jika semua kriteri sudah terpenuhi, maka sistem akan mengecek apakah buku dengan id tersebut ditemukan.
   6. Jika buku ditemukan, maka sistem akan mengecek apakah user pernah mereview buku tersebut.
   7. Jika user pernah mereview buku tersebut, maka sistem akan memperbarui data review dari id buku tersebut.

Detail endpoint:

/review/:id\_book (PUT)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Body:

* rating: Rating buku (1-5)
* comment: Isi comment

1. Delete review
   1. User memasukkan token jwt.
   2. User memasukkan id buku goodreads yang akan dihapus pada parameter.
   3. Sistem akan mengecek apakah token jwt valid.
   4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter yang diperlukan sudah terpenuhi.
   5. Jika semua kriteri sudah terpenuhi, maka sistem akan mengecek apakah buku dengan id tersebut ditemukan.
   6. Jika buku ditemukan, maka sistem akan mengecek apakah user pernah mereview buku tersebut.
   7. Jika user pernah mereview buku tersebut, maka sistem akan menghapus review tersebut.

Detail endpoint:

/review/:id\_book (DELETE)

Header:

* x-auth-token : JWT token yang didapatkan saat login

**#Bookmark**

|  |  |  |
| --- | --- | --- |
| Activity | Verb/Noun Mapping | Participant |
| Create bookmark | POST/bookmark/:id\_book | Free & premium user |
| Get bookmark | GET/bookmark/:id\_book | Free & premium user |
| Update bookmark | PUT/bookmark/:id\_book | Free & premium user |
| Delete bookmark | DELETE/bookmark/:id\_book | Free & premium user |

* 1. create bookmark
     1. User memasukkan token jwt.
     2. User memasukkan id buku goodreads pada parameter dan memasukkan page number dan note pada body.
     3. Sistem akan mengecek apakah token jwt valid.
     4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter dan body yang diperlukan sudah terpenuhi.
     5. Jika semua kriteri sudah terpenuhi, maka sistem akan mengecek apakah buku dengan id tersebut ditemukan.
  2. Jika buku ditemukan, maka sistem akan mengecek apakah user belum pernah membookmark buku tersebut.
  3. Jika user belum pernah membookmark buku tersebut, maka sistem akan memasukkan data bookmark ke database.

Detail endpoint:

/bookmark/:id\_book (POST)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Body:

* Page\_number: halaman buku yang di bookmark
* Note: catatan bookmark
  1. get bookmark
     1. User memasukkan token jwt.
     2. User memasukkan id buku goodreads pada parameter.
     3. Sistem akan mengecek apakah token jwt valid.
     4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter yang diperlukan sudah terpenuhi.
     5. Jika semua kriteri sudah terpenuhi, maka sistem akan menampilkan data bookmark sesuai dengan parameter.

Detail endpoint:

/bookmark/:id\_book (GET)

Header:

* x-auth-token : JWT token yang didapatkan saat login
  1. update bookmark
     1. User memasukkan token jwt.
     2. User memasukkan id buku goodreads pada parameter dan memasukkan page number dan note pada body.
     3. Sistem akan mengecek apakah token jwt valid.
     4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter dan body yang diperlukan sudah terpenuhi.
     5. Jika semua kriteri sudah terpenuhi, maka sistem akan mengecek apakah buku dengan id tersebut ditemukan.
     6. Jika buku ditemukan, maka sistem akan mengecek apakah user pernah membookmark buku tersebut.
     7. Jika user pernah membookmark buku tersebut, maka sistem akan memperbarui data bookmark dari id buku tersebut.

Detail endpoint:

/bookmark/:id\_book (PUT)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Body:

* Page\_number: halaman buku yang di bookmark

Note: catatan bookmark

* 1. delete bookmark
     1. User memasukkan token jwt.
     2. User memasukkan id buku goodreads yang akan dihapus pada parameter.
     3. Sistem akan mengecek apakah token jwt valid.
     4. Jika token jwt valid, maka sistem akan mengecek apakah kriteria parameter yang diperlukan sudah terpenuhi.
     5. Jika semua kriteri sudah terpenuhi, maka sistem akan mengecek apakah buku dengan id tersebut ditemukan.
     6. Jika buku ditemukan, maka sistem akan mengecek apakah user pernah membookmark buku tersebut.
     7. Jika user pernah membookmark buku tersebut, maka sistem akan menghapus bookmark tersebut.

Detail endpoint:

/bookmark/:id\_book (DELETE)

Header:

* x-auth-token : JWT token yang didapatkan saat login

1. /bookshelf/insert (POST)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Body:

* book\_id: Id buku dari goodreads

Tujuan: Menambahkan buku ke bookshelf user.

1. /bookshelf (GET)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Tujuan: Untuk melihat semua buku yang terdapat pada bookshelf user.

1. /bookshelf/search/:target (GET)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Params:

* target: Username dari user yang ingin dilihat bookshelf-nya

Tujuan: Untuk melihat bookshelf dari user lain, biasanya digunakan sebagai referensi.

1. /bookshelf/changePrivacy (PUT)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Tujuan: Untuk mengubah privasi bookshelf (bisa dilihat oleh orang lain apa tidak). Bookshelf bertipe 0 adalah public, sedangkan 1 adalah private.

1. /bookshelf/delete (DELETE)

Header:

* x-auth-token : JWT token yang didapatkan saat login

Body:

* book\_id: Id buku dari bookshelf user

Tujuan: Untuk menghapus buku dari bookshelf user.

1. /playlist/create (POST)

Header:

* x-auth-token: Token yang didapat saat login

Body:

* name: Nama untuk playlist baru

Tujuan: Untuk membuat playlist baru

1. /playlist/insert (POST)

Header:

* x-auth-token: Token yang didapat saat login

Body:

* playlist\_id: Id dari playlist yang tersedia
* book\_id: Id buku dari goodreads

Tujuan: Untuk menambahkan buku ke dalam playlist

1. /playlist (GET)

Header:

* x-auth-token: Token yang didapat saat login

Tujuan: Untuk melihat detail playlist yang telah dibuat user

1. /playlist/search/user/:username (GET)

Query:

* sort: Untuk mengurutkan data yang ditampilkan (date/name) bisa juga bila tidak diisi

Params:

* username: Username dari user yang ingin dilihat playlist-nya

Tujuan: Untuk mencari playlist berdasarkan username pembuatnya

1. /playlist/search/name/:name (GET)

Query:

* sort: Untuk mengurutkan data yang ditampilkan (date/name) bisa juga bila tidak diisi

Params:

* name: Nama dari playlist yang ingin dicari

Tujuan: Untuk mencari playlist berdasarkan nama playlist

1. /playlist/changeName (PUT)

Header:

* x-auth-token: Token yang didapat saat login

Body:

* playlist\_id: Id playlist yang telah terdaftar
* name: Nama baru untuk playlist

Tujuan: Mengubah nama dari playlist

1. /playlist/changePrivacy (PUT)

Header:

* x-auth-token: Token yang didapat saat login

Body:

* playlist\_id: Id playlist yang telah terdaftar

Tujuan: Untuk mengubah privasi dari playlist (bisa dilihat oleh orang lain atau tidak)

1. /playlist/delete/book/:book\_id (DELETE)

Header:

* x-auth-token: Token yang didapat saat login

Params:

* book\_id: Id buku yang terdapat pada playlist

Body:

* playlist\_id: Id playlist yang telah terdaftar

Tujuan: Untuk menghapus buku dari playlist

1. /playlist/delete (DELETE)

Header:

* x-auth-token: Token yang didapat saat login

Body:

* playlist\_id: Id playlist yang telah terdaftar

Tujuan: Untuk menghapus playlist yang telah dibuat