

Proiect GlobalWaves - Etapa 2 - Pagination



- **Responsabili:** Rareș Constantin [mailto:rares.constantin2002@gmail.com], Alexandru Marinescu [mailto:alex24marinescu@gmail.com]
- **Colaboratori:** Sorina-Anamaria Buf [mailto:sorinabuf@gmail.com], Ștefan Cocioran [mailto:stefanccocioran@gmail.com], Florian-Luis Micu [mailto:miculuis1@gmail.com]
- **Data publicării:** 28.11.2023, ora 12:00
- **Deadline HARD:** 15.12.2023, ora 23:55
- **Ultima modificare a cerinței:** 08.12.2023, ora 17:10 - corectare output printCurrentPage pentru Host Page
- **Ultima modificare a scheletului:** 05.12.2023, ora 13:30 - update checkstyle 0 errors
- **Ultima modificare a testelor:** 04.12.2023, ora 21:15 - update ref test09, test15, test16
- **Schelet:** <https://github.com/oop-pub/oop-project-2023/tree/main/etapa2> [<https://github.com/oop-pub/oop-project-2023/tree/main/etapa2>]
- **Atenție!** Pentru partea de citire / afișare nu este necesară folosirea adnotărilor din librăria Jackson. Puteți utiliza o metodă similară cu cea prezentată de noi în schelet, și anume să vă folosiți de clasa ObjectMapper. **Nu există o metodă corectă sau greșită de a lucra cu input-ul / output-ul.** Scopul nostru în cadrul acestui proiect nu este să vă testăm abilitatea de a lucra cu fișierele, ci să vă testăm atât înțelegerea cunoștințelor de bază dobândite în cadrul cursurilor și a laboratoarelor, cât și modul de gândire într-un limbaj de programare orientat obiect.

Dacă decideți să folosiți Generative AI (ex. ChatGPT) pentru implementare sau alte aspecte ale codului, treceți în README exact unde ați folosit această metodă.

De asemenea, scrieți în README și dacă ați folosit ca schelet rezolvarea oficială a etapei I.

Obiective

- Implementarea a cel puțin unui design pattern POO.
- Refactorizarea codului pentru a permite adăugarea funcționalităților noi.

Descriere

Pentru acest proiect o să aveți de implementat o aplicație asemănătoare ca funcționalități cu Spotify, simulând diferite acțiuni făcute de utilizatori, acum incluzând artiști și hosts (prezențați mai jos). Aceste acțiuni vor fi simulate folosind niște comenzi primite în fișierele de input. Perspectiva de admin asupra utilizatorilor și a elementelor aplicației se păstrează.

Comenzile de la etapa I și formatul input-ului și al output-ului nu se modifică. Tot ce ați implementat în etapa I, rămâne valid și în cadrul etapei curente. În această etapă va trebui să extindeți soluția de la etapa I cu funcționalități noi.

Sistem de pagini

Pentru că avem baza de date în picioare, acum putem simula și o interfață grafică minimală prin compartimentarea aplicației în pagini diferite pe care un user normal se poate afla la un moment dat. Astfel vor exista următoarele tipuri de pagini pentru un user individual, urmând ca în etapa III să putem adăuga mai multe pagini:

- HomePage - Pagina unde orice utilizator se află atunci când este adăugat prima dată pe platformă. Aici utilizatorul va putea vedea static "recomandări", însemnând că va putea vedea primele maxim 5 melodii la care a dat like după numărul de like-uri și primele 5 playlisturi urmărite ca număr de like-uri total pe toate melodiile din acel playlist.
- LikedContentPage - Pagina unde orice utilizator poate vedea toate melodiile și toate playlisturile apreciate/urmărite de acesta.

Aceste două pagini sunt accesibile de oriunde din aplicație și pot fi accesate la orice moment de timp, mai puțin când utilizatorul normal este offline.

Artist page

Fiecare artist va avea propria pagină atunci când este adăugat pe platformă și va putea manipula anumite detalii despre aceasta. Un utilizator normal poate ajunge pe pagina artistului prin căutarea acestuia și apoi selectarea lui din rezultatele căutării, care va goli lista de rezultate după căutare. Acesta va putea vedea pe pagină albumele, articolele de vânzare și evenimentele artistului. Orice modificare adusă de artist asupra albumelor, merch-ului și a evenimentelor va trebui să se regăsească la același moment de timp pentru toți utilizatorii care se află pe pagina acestuia.

Host page

Asemănător cu pagina de artist, un utilizator normal va putea vedea pe această pagină podcasturile și anunțurile hostului.

Comenzi pentru sistemul de pagini

Doar utilizatorii normali vor putea folosi comenzile pentru pagini!

ChangePage

Mesaje posibile pentru această comandă:

- <username> accessed <next page> successfully.
- <username> is trying to access a non-existent page.

<pre>{ "command": "changePage", "username": "alice22", "timestamp": 100, "nextPage": "Home" }</pre>
<pre>{ "command": "changePage", "username": "alice22", "timestamp": 100, "nextPage": "LikedContent" }</pre>
<pre>{ "command" : "changePage", "user" : "alice22", "timestamp" : 100, "message" : "alice22 accessed Home successfully." }</pre>
<pre>{ "command" : "changePage", "user" : "alice22", "timestamp" : 100, "message" : "alice22 accessed LikedContent successfully." }</pre>

```
    "message" : "alice22 accessed LikedContent successfully."
  }
```

```
{
  "command" : "changePage",
  "user" : "alice22",
  "timestamp" : 100,
  "message" : "alice22 is trying to access a non-existent page."
}
```

PrintCurrentPage

Prin această comandă vom vedea pagina pe care se află un anumit user. Astfel vom folosi un format de string pentru fiecare pagină, după cum urmează:

- HomePage:

Liked songs:\n\t[songname1, songname2, ...]\n\nFollowed playlists:\n\t[playlistname1, playlistname2, ...]

- LikedContentPage

Liked Songs:\n\t[songname1 - songartist1, songname2 - songartist2, ...]\n\nFollowed Playlists:\n\t[playlistname1 - owner1, playlistname2 - owner2, ...]

- Artist page

Albums:\n\t[albumname1, albumname2, ...]\n\nMerch:\n\t[merchname1 - merchprice1:\n\t\tmerchdescription1, merchname2 - merchprice2:\n\t\tmerchdescription2, ...]\n\nEvent:\n\t[eventname1 - eventdate1:\n\t\tteventdescription1, eventname2 - eventdate2:\n\t\tteventdescription2, ...]

- Host page

[illegible]

```
{
  "command": "printCurrentPage",
  "username": "alice22",
  "timestamp": 420,
}
```

```
{
  "command": "printCurrentPage",
  "user": "alice22",
  "timestamp": 420,
  "message": "Liked songs:\n\t[songname1, songname2]\n\nFollowed Playlists:\n\t[playlistname1 - owner1, playlistname2 - owner2, ...]"
}
```

Entități noi față de prima etapă

Useri

Se adaugă două tipuri de useri, fiecare având comenzi specifice prezentate mai jos:

- artist - user care poate adăuga albume și are propria sa pagină în cadrul acestei etape
- host - user care poate adăuga podcast-uri și are propria sa pagină în cadrul acestei etape

Colectii de fisiere audio

Se adaugă următorul tip de colecție:

- album

Albumul este o colecție de melodii (Song) creată de către un artist. Utilizatorii normali pot să caute și să asculte albume, prin funcțiile de select și load. Melodiile nu pot fi adăugate în cadrul aplicației în afara unui album.

Recapitulare Entități

Library-ul reprezintă totalitatea melodiilor din platformă. Toți utilizatorii platformei au acces la toată biblioteca de melodii.

Playlist-ul este o colecție de melodii creată de un utilizator. Un playlist poate fi public sau privat. Acesta este format din fișiere audio din library în funcție de preferințele utilizatorului.

Podcast-ul este o colecție formată din mai multe episoade ce au legătură între ele. Toate podcasturile vor fi specificate în fișierul de intrare, fiind accesibile în biblioteca de la începutul simulării. Episoadele sunt ordonate în funcție de poziția inițială specificată la input.

Search bar

Noutăți search bar:

- Album:
 - by name → se verifică dacă numele albumului începe cu textul specificat
 - by owner → se verifică dacă artistul care a creat albumul începe cu textul specificat
 - by description → se verifică dacă descrierea albumului începe cu textul specificat
- Artist:
 - by username → se verifică dacă numele artistului începe cu textul specificat
- Host:
 - by username → se verifică dacă numele hostului începe cu textul specificat

Pentru fiecare tip de căutare garantăm că cel puțin un câmp o să fie specificat. Ordinea în care se obțin rezultatele ține de poziția conținutului în lista cu toate acele elemente (adică în lista de melodii, în lista cu podcast-uri sau în listele cu playlist-uri în cazul fiecărui utilizator).

Un utilizator normal va putea căuta acum un artist/un host și va putea aplica comanda "select" pe aceștia, urmând să fie trimis către pagina entry-ului selectat din rezultatele căutării. Eroarea "item number" se păstrează în acest caz, iar rezultatele vor fi trimise ca listă de String-uri formată din numele artiștilor/hostilor rezultați. După selecție, rezultatele vor fi șterse, exact ca în prima etapă.

Music player

Acum un utilizator normal va putea să dea select pentru un album și apoi load acestui tip de colecție. Albumul trebuie să suporte aceleași funcționalități ca un playlist, incluzând repeat și shuffle.

Recapitulare Timestamp

Pentru a simula caracterul real al aplicației, comenzile au un câmp numit "timestamp", care precizează la ce secundă s-au executat față de începutul testului (momentul t_0). Fiecare comandă este executată instantaneu într-un moment de timp. Timpul este comun pentru toți utilizatorii, și trece constant, indiferent de comenzile primite. Astfel, nu vom putea avea o comandă cu timestamp-ul "30" pentru "user1", urmată de o comandă cu timestamp-ul "20" pentru "user2" (o comandă cu timestampul "t" marchează faptul că toți utilizatorii se afla la momentul de timp "t"). De asemenea, trebuie simulat ce se întâmplă între momentele de timp, în cazul nostru, starea în care se află player-ul utilizatorului curent (la ce track a ajuns, dacă s-a oprit rularea etc.).

Comenzi search bar

Search

Se adaugă ca funcționalitate posibilitatea de a căuta un album, un artist sau un host.

```
{
  "command": "search",
  "username": "alice22",
  "timestamp": 100,
```

```
{
  "type": "album",
  "filters": {
    "name": "Th",
    "owner": "U",
    "description": "Best"
  }
}
```

```
{
  "command": "search",
  "username": "alice22",
  "timestamp": 100,
  "type": "artist",
  "filters": {
    "name": "The"
  }
}
```

```
{
  "command": "search",
  "username": "alice22",
  "timestamp": 100,
  "type": "host",
  "filters": {
    "name": "The"
  }
}
```

```
{
  "command" : "search",
  "user" : "alice22",
  "timestamp" : 10,
  "message" : "Search returned 3 results",
  "results" : [ "This Is Why", "The Mad Stone", "The Eminem Show" ]
}
```

Player

Player-ul unui user normal va putea avea ca sursă audio un album și va putea folosi toate comenzile pentru playlist.

Comenzi Admin

AddUser

Ca admin putem adăuga useri noi, fie ei useri normali, artiști sau hosts. Acest lucru se va face cu aceasta comandă, conținând doar detaliile de bază pentru un user, mai apoi lăsând fiecare user să își schimbe detaliile, cum ar fi adăugarea unui album pentru artist. Userii normali care există deja în teste nu vor fi modificați, iar toți artiștii și hostii vor fi adăugați pe parcursul fiecărui test.

Mesaje posibile pentru această comandă:

- The username <username> is already taken.
- The username <username> has been added successfully.

```
{
  "command": "addUser",
  "timestamp": 100,
  "type": "user/artist/host",
  "username": "anaaremere",
  "age": 20,
  "city": "New York"
}
```

```
{
  "command" : "addUser",
  "timestamp" : 100,
  "user": "anaaremere",
  "message" : "The username anaaremere has been added successfully"
}
```

DeleteUser

Ca admin putem să ștergem un user, iar toate legăturile sale cu alte entități din aplicație trebuie și ele șterse/updatate. Un user de orice tip nu poate fi șters dacă un alt user are interacțiuni în acel moment de timp în aplicație cu una din entitățile sale. De exemplu: vrem să ștergem un host, iar un user îi ascultă în acel moment unul din podcasturile sale. Toate cazurile de tratare pentru acest tip de comanda rămân la latitudinea voastră pentru a le trata.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> can't be deleted.
- <username> was successfully deleted.

```
{
  "command": "deleteUser",
  "username": "alice22",
  "timestamp": 120
}
```

```
{
  "command" : "deleteUser",
  "user" : "alice22",
  "timestamp" : 120,
  "message" : "alice22 was successfully deleted."
}
```

ShowAlbums

În cadrul acestei comenzi se vor afișa toate albumele unui artist. Se garantează că username-ul primit în input este un artist și acesta nu va fi invalid. Pentru fiecare album se va afișa numele lui și o listă cu numele melodiilor din album.

```
{
  "command": "showAlbums",
  "username": "Ed Sheeran",
  "timestamp": 400,
}
```

```
{
  "command": "showAlbums",
  "user": "Ed Sheeran",
  "timestamp": 400,
  "result": [ {
    "name": "Divide",
    "songs": [ "Shape of You", "Happier", "Castle on the Hill" ]
  } ]
}
```

ShowPodcasts

În cadrul acestei comenzi se vor afișa toate podcasturile unui host. Se garantează că username-ul primit în input este un host și acesta este valid. Pentru fiecare podcast se va afișa numele lui și o listă cu numele episoadelor din podcast.

```
{
  "command": "showPodcasts",
```

```

"username": "Joe Rogan",
"timestamp": 420,
}

```

```

{
  "command": "showPodcasts",
  "user": "Joe Rogan",
  "timestamp": 420,
  "result": [ {
    "name": "The Joe Rogan Experience",
    "episodes": [ "Elon Musk Returns", "Jordan Peterson", "Neil deGrasse Tyson", "Elon Musk on Mars", "The Art of MMA" ]
  } ]
}

```

Comenzi artist

AddAlbum

Această comandă permite unui artist să adauge un nou album în aplicație. Artistul specifică detaliile albumului, inclusiv numele, anul lansării, descrierea și lista de melodii care compun albumul.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not an artist.
- <username> has another album with the same name.
- <username> has the same song at least twice in this album.
- <username> has added new album successfully.

```

{
  "command": "addAlbum",
  "username": "Ed Sheeran",
  "timestamp": 160,
  "name": "Album1",
  "releaseYear": 2023,
  "description": "Primul album bro!"
  "songs": [
    {
      "name": "Shape of You",
      "duration": 233,
      "album": "Divide",
      "tags": [
        "#pop",
        "#mostlistenedthisyear",
        "#spotify"
      ],
      "lyrics": "The club isn't the best place to find a lover, So the bar is where I go (mm-mm)",
      "genre": "Pop",
      "releaseYear": 2017,
      "artist": "Ed Sheeran"
    },
    {
      "name": "Don't",
      "duration": 219,
      "album": "x",
      "tags": [
        "#pop",
        "#relationship",
        "#spotify"
      ],
      "lyrics": "I met this girl late last year, She said, 'Don't you worry if I disappear",
      "genre": "Pop",
      "releaseYear": 2014,
      "artist": "Ed Sheeran"
    }
  ]
}

```

```

{
  "command": "addAlbum",

```

```
{
  "user": "Ed Sheeran",
  "timestamp": 160,
  "name": "Ed Sheeran has added new album successfully."
}
```

RemoveAlbum

Un artist poate șterge unul din albumurile sale. În cazul în care există un album cu acel nume, acesta va putea fi șters dacă nu există niciun utilizator normal care să aibă albumul sau o melodie din acesta loaded (nu se ia în considerare dacă utilizatorul doar se află pe pagina artistului) sau un playlist care să conțină o melodie din album.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not an artist.
- <username> doesn't have an album with the given name.
- <username> can't delete this album.
- <username> deleted the album successfully.

```
{
  "command": "removeAlbum",
  "username": "Ed Sheeran",
  "timestamp": 180,
  "name": "Album1"
}
```

```
{
  "command": "removeAlbum",
  "username": "Ed Sheeran",
  "timestamp": 180,
  "message": "Ed Sheeran deleted the album successfully."
}
```

AddEvent

Artistul poate adăuga un eveniment. Evenimentul va avea o descriere și o dată care se consideră a fi invalidă în următoarele situații:

- Formatul corect este dd-mm-yyyy, însă în input acesta poate să fie și invalid, caz în care trebuie să afișați o eroare.
- Pentru luna februarie ziua este mai mare de 28 sau în general mai mare decât 31, luna este mai mare de 12, anul este mai mic de 1900 sau mai mare de 2023.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not an artist.
- <username> has another event with the same name.
- Event for <username> does not have a valid date.
- <username> has added new event successfully.

```
{
  "command": "addEvent",
  "username": "Ed Sheeran",
  "timestamp": 200,
  "name": "Event1",
  "description": "Primul Event adaugat!",
  "date": "01-01-2022"
}
```

```
{
  "command": "addEvent",
```



```
{
  "user": "Ed Sheeran",
  "timestamp": 200,
  "message": "Ed Sheeran has added new event successfully."
}
```

RemoveEvent

Această comandă permite unui artist să șteargă un eveniment existent.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not an artist.
- <username> doesn't have an event with the given name.
- <username> deleted the event successfully.

```
{
  "command": "removeEvent",
  "username": "Ed Sheeran",
  "timestamp": 220,
  "name": "Event1"
}
```

```
{
  "command": "removeEvent",
  "user": "Ed Sheeran",
  "timestamp": 220,
  "message": "Ed Sheeran deleted the event successfully."
}
```

AddMerch

Artistul are posibilitatea de a adăuga articole de merchandising în aplicație. Aceste articole includ descrieri și prețuri.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not an artist.
- <username> has merchandise with the same name.
- Price for merchandise can not be negative.
- <username> has added new merchandise successfully.

```
{
  "command": "addMerch",
  "username": "Ed Sheeran",
  "timestamp": 240,
  "name": "Merch1",
  "description": "Primul Merch adaugat!",
  "price": 100
}
```

```
{
  "command": "addMerch",
  "user": "Ed Sheeran",
  "timestamp": 240,
  "message": "Ed Sheeran has added new merchandise successfully."
}
```

Comenzi host

AddPodcast

Această comandă permite unui host să adauge un nou podcast în aplicație. Hostul specifică detaliile podcastului, inclusiv numele și lista de episoade cu detaliile aferente fiecărui episod (nume, durată, descriere).

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not a host.
- <username> has another podcast with the same name.
- <username> has the same episode in this podcast.
- <username> has added new podcast successfully.

```
{
  "command": "addPodcast",
  "username": "Mike",
  "timestamp": 260,
  "name": "Podcast1",
  "episodes": [
    {
      "name": "Elon Musk Returns",
      "duration": 11927,
      "description": "Elon Musk, CEO of SpaceX and Tesla, returns to discuss various topics."
    },
    {
      "name": "Jordan Peterson",
      "duration": 9916,
      "description": "Dr. Jordan Peterson joins Joe to discuss psychology, philosophy, and current events."
    }
  ]
}
```

```
{
  "command": "addPodcast",
  "user": "Mike",
  "timestamp": 260,
  "message": "Mike has added new podcast successfully."
}
```

RemovePodcast

Permite unui host să ștergă un podcast existent. În cazul în care există un podcast cu acel nume, acesta va putea fi șters dacă nu există niciun utilizator normal care are podcastul loaded (nu se ia în considerare dacă utilizatorul doar se află pe pagina hostului).

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not a host.
- <username> doesn't have a podcast with the given name.
- <username> can't delete this podcast.
- <username> deleted the podcast successfully.

```
{
  "command": "removePodcast",
  "username": "Mike",
  "timestamp": 280,
  "name": "Podcast1"
}
```

```
{
  "command": "removePodcast",
  "user": "Mike",
  "timestamp": 280,
  "message": "Mike deleted the podcast successfully."
}
```

AddAnnouncement

Hostul poate adăuga anunțuri pentru a informa audiența despre diferite subiecte sau evenimente. Acestea vor avea un nume și o descriere.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not a host.
- <username> has already added an announcement with this name.
- <username> has successfully added new announcement.

```
{
  "command": "addAnnouncement",
  "username": "Mike",
  "timestamp": 300,
  "name": "Announcement1",
  "description": "Primul anunt adaugat!"
}
```

```
{
  "command": "addAnnouncement",
  "user": "Mike",
  "timestamp": 300,
  "message": "Mike has successfully added new announcement."
}
```

RemoveAnnouncement

Această comandă permite unui host să șteargă un anunț existent.

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not a host.
- <username> has no announcement with the given name.
- <username> has successfully deleted the announcement.

```
{
  "command": "removeAnnouncement",
  "username": "Mike",
  "timestamp": 320,
  "name": "Announcement1"
}
```

```
{
  "command": "removeAnnouncement",
  "user": "Mike",
  "timestamp": 320,
  "message": "Mike has successfully deleted the announcement."
}
```

Comenzi user normal

SwitchConnectionStatus

Toți utilizatorii normali inițial vor fi considerați online, inclusiv atunci când este adăugat pe platformă, și au acces la toate comenzile lor, incluzând comenzile din etapa I. Această comandă va trece un utilizator normal din modul online, în modul offline sau invers, oprind orice acces la platformă și acesta nu va putea utiliza alte comenzi ce țin de utilizatorii normali, mai puțin cele care țin de date care sunt accesibile de către admin: playlist-urile, melodiile preferate, genul preferat. Când utilizator este offline, pentru acesta timpul nu va mai fi contorizat, player-ul său fiind

oprit până când revine online. În cazul în care utilizatorul este offline în momentul folosirii unei comenzi de mai jos, mesajul va fi "<username> is offline."

Listă comenzi oprite în modul offline: search, select, load, playPause, repeat, shuffle, forward, backward, like, next, prev, createPlaylist, addRemoveInPlaylist, switchVisibility, follow, changePage, printCurrentPage

Mesaje posibile pentru această comandă:

- The username <username> doesn't exist.
- <username> is not a normal user.
- <username> has changed status successfully.

```
{
  "command": "switchConnectionStatus",
  "username": "Mike",
  "timestamp": 320,
}
```

```
{
  "command": "switchConnectionStatus",
  "user": "Mike",
  "timestamp": 320,
  "message": "Mike has changed status successfully."
}
```

Statistici generale

getTop5Albums

Această comandă va afișa numele celor mai apreciate 5 albume din aplicație în funcție de numărul de like-uri. Numărul de like-uri ale unui album este considerat ca fiind suma tuturor like-urilor melodiilor din acel album. În cazul în care sunt mai puțin de 5 albume, se vor afișa toate. În cazul în care sunt două albume cu număr egal de like-uri, se va afișa în ordine lexicografică.

```
{
  "command": "getTop5Albums",
  "timestamp": 540
}
```

```
{
  "command": "getTop5Albums",
  "timestamp": 540,
  "result": [ "Divide", "Led Zeppelin IV", "Hot Space", "Aerosmith", "Vol. 3... Life and Times of S. Carter" ]
}
```

getTop5Artists

Această comandă va afișa numele celor mai apreciați 5 artiști din aplicație în funcție de numărul de like-uri. Numărul de like-uri ale unui artist este considerat ca fiind suma tuturor like-urilor melodiilor din toate albumele. În cazul în care sunt mai puțin de 5 artiști, se vor afișa toți.

```
{
  "command": "getTop5Artists",
  "timestamp": 560
}
```

```
{
  "command": "getTop5Artists",
  "timestamp": 560,
```

```
{
  "result": [ "Jay Z", "Meek Mill", "The Rolling Stones", "Led Zeppelin", "Maroon 5" ]
}
```

getAllUsers

În cadrul acestei comenzi se vor afișa numele tuturor utilizatorilor din aplicație. Ordinea afișării este următoarea: utilizatori normali, artiști, hosti.

```
{
  "command": "getAllUsers",
  "timestamp": 580
}
```

```
{
  "command": "getAllUsers",
  "timestamp": 580,
  "result": [ "alice22", "bob35", "Ed Sheeran", "Mike" ]
}
```

getOnlineUsers

În cadrul acestei comenzi se vor afișa numele tuturor utilizatorilor normali din aplicație care sunt online.

```
{
  "command": "getOnlineUsers",
  "timestamp": 600
}
```

```
{
  "command": "getOnlineUsers",
  "timestamp": 600,
  "result": [ "alice22", "bob35" ]
}
```

Scheletul de cod

În rezolvarea etapei 2 veți putea folosi ca punct de pornire soluția oficială a etapei 1, dar vă sfătuim să lucrați pe codul scris de voi pentru a putea înțelege mai repede cum să implementați noile funcționalități fără nevoia de a înțelege principiile și interacțiunile din soluția oficială.

Pentru a înțelege mai bine cum funcționează citirea/scrie în fișierele JSON vă recomandăm să citiți [Json & Jackson](https://ocw.cs.pub.ro/courses/poo-ca-cd/laboratoare/tutorial-json-jackson) [https://ocw.cs.pub.ro/courses/poo-ca-cd/laboratoare/tutorial-json-jackson].

În cadrul soluției oficiale este folosit Lombok [https://projectlombok.org/features/], un tool care simplifică codul prin eliminarea anumitor segmente explicite, cum ar fi getteri și setteri. Puteți folosi și voi aceste adnotări.

Output-ul nu trebuie formatat ca în ref-uri, fiindcă se verifică conținutul obiectelor și array-urilor JSON, nu textul efectiv. Cu toate acestea, dacă folosiți Jackson, vă recomandăm să utilizați **PrettyPrinter** Documentație PrettyPrinter [https://fasterxml.github.io/jackson-databind/javadoc/2.7/com/fasterxml/jackson/databind/ObjectMapper.html#writerWithDefaultPrettyPrinter()]. Totodată, pentru a înțelege cum se poate realiza **scrierea în fișierele JSON de output**, vă sugerăm să consultați [JSON Array](http://makeseleniumeasy.com/2020/05/16/rest-assured-tutorial-27-how-to-create-json-array-using-jackson-api-objectmapper-createarraynode/) [http://makeseleniumeasy.com/2020/05/16/rest-assured-tutorial-27-how-to-create-json-array-using-jackson-api-objectmapper-createarraynode/].

Aveți în folder-ul **"lib"** toate dependențele necesare pentru rularea temei, mai exact bibliotecile Jackson.

Execuția temei

1. Se citesc listele cu useri, melodii si podcast-uri, in format JSON - e facuta deja citirea in schelet.
2. Se citesc comenzile si sunt puse in obiecte.

3. Se primesc secvențial comenzi și se execută pe măsură ce sunt primite.
 4. După executarea unei comenzi, se afișează rezultatul ei în fișierul JSON de ieșire.
 5. La terminarea tuturor comenzilor se termină și execuția programului.
- După ce clonați repo-ul de pe GitHub, vă rugăm să vă faceți un repository propriu privat în care să vă puneți doar conținutul folder-ului **"etapa1"** de pe repo-ul echipei de POO. Dacă nu puneți folder-ul cu tema la alta cale, **nu o să puteți** să faceți schimbări în Git, deoarece vă aflați în rădăcina repository-ului echipei de POO.
 - Pentru ca checker-ul să funcționeze trebuie să deschideți tema din IntelliJ la calea unde se află folderele **"src"**, **"lib"**, **"ref"**, **"input"**. Aveți folder-ul **.idea** pregenerat ca să vă ajute în acest sens. De asemenea, fișierul **.iml** conține calea către bibliotecile Jackson. Dacă aveți probleme stergeți folder-ul **.idea** și fișierul **.iml** și generați-le voi din nou din IntelliJ.
 - **Citirea comenzilor și afișarea rezultatelor trebuie făcută de voi!!!**

Recomandări

- Tema a fost concepută pentru a vă testa cunoștințele dobândite în urma parcurgerii laboratoarelor 5-8, aceasta putând fi rezolvată doar cu noțiunile învățate din acele laboratoare.
- Pentru depanarea diferențelor dintre output-ul vostru și fișierele ref, vă recomandăm acest site [<https://www.jsondiff.com/>].
- Verificați periodic această pagină, deoarece scheletul/cerința pot suferi modificări în urma unor erori din partea noastră.

Evaluare

Punctajul constă din:

- 80p implementare - trecerea testelor
- 10p coding style (vezi checkstyle)
- 5p README clar, concis, explicații axate pe design (flow, interacțiuni)
- 5p folosire git pentru versionarea temei

Este obligatoriu să folosiți cel puțin un design pattern din cele învățate la laborator, depunerea este de 10 puncte în caz contrar!

Dacă decideți să folosiți Generative AI (ex. ChatGPT) pentru implementare sau alte aspecte ale codului, treceți în README exact unde ați folosit această metodă.

De asemenea, scrieți în README și dacă ați folosit ca schelet rezolvarea oficială a etapei I.

Pentru folosirea tool-ului **Git** vă punem la dispoziție un tutorial actualizat și amplu despre el la acest [link](#) și aveți de asemenea și un tutorial despre comenzile pe care puteți să le dați din IntelliJ la acest [link](#).

Pe pagina [Indicații pentru teme](#) găsiți indicații despre scrierea readme-ului și depunțările generale pentru teme

Depunțările pentru **designul și organizarea codului** se vor scădea din punctajul testelor. Dacă vor apărea depunțări specifice temei în momentul evaluării, nementionate pe pagina cu depunțări generale, ele se vor încadra în limitele de maxim 15 pentru design, 5p pentru readme. Dacă tema nu respecta cerințele, sau are zero design OOP atunci se pot face depunțări suplimentare.

Folosirea git pentru versionare va fi verificată din folderul **.git** pe care trebuie să îl includeți în arhiva temei. Punctajul se va acorda dacă ați făcut **minim 3 commit-uri relevante și cu mesaj sugestiv**. **NU** este permis să aveți repository-urile de git publice până la deadline-ul hard.

Bonusuri: La evaluare, putem oferi bonusuri pentru design foarte bun, cod bine documentat, dar și pentru diverse elemente suplimentare alese de voi.

- Temele vor fi testate împotriva plagiatului. Orice tentativă de copiere va duce la **anularea punctajului** de pe parcursul semestrului și **repetarea materiei** atât pentru sursă(e) cât și pentru destinație(ii), fără excepție.
- **Aveți grijă să nu puneți pe Vmchecker fișiere .idea sau .iml.**

Checkstyle

Unul din obiectivele temei este învățarea respectării code-style-ului limbajului pe care îl folosiți. Aceste convenții (de exemplu cum numiți fișierele, clasele, variabilele, cum indentați) sunt verificate pentru temă de către tool-ul checkstyle [<https://checkstyle.sourceforge.io/>].

Pe pagina de [Recomandări cod](#) găsiți câteva exemple de coding style.

Dacă numărul de erori depistate de checkstyle depășește 30, atunci punctele pentru coding-style nu vor fi acordate. Dacă punctajul este negativ, *acesta se trunchiază la 0*.

Exemple:

- `punctaj_total = 100` și `nr_erori = 200` \Rightarrow `nota_finala = 90`
- `punctaj_total = 100` și `nr_erori = 29` \Rightarrow `nota_finala = 100`
- `punctaj_total = 80` și `nr_erori = 30` \Rightarrow `nota_finala = 80`
- `punctaj_total = 80` și `nr_erori = 31` \Rightarrow `nota_finala = 70`

Upload temă

Arhiva pe care o veți urca pe VMChecker [<https://vmchecker.cs.pub.ro/ui/#POO>] va trebui să conțină în directorul rădăcină:

- fișierul README
- folder-ul src cu pachetele și cu fișierele .java
- folderul .git

Resurse și linkuri utile

- Schelet de cod [<https://github.com/oop-pub/oop-project-2023/tree/main/etapa2>]
- [Indicații pentru teme](#)
- [Recomandări coding style & javadoc](#)

poo-ca-cd/teme/proiect/etapa2.txt · Last modified: 2023/12/08 17:11 by rares.constantin02