

PNRS1 projektni zadatak - Igra memorije

Zadatak 4 - HTTP

Ovogodišnji projektni zadatak biće izrada Android aplikacije pod nazivom **Memory Game**. Projekat se sastoji iz šest celina koje se nastavljaju jedna na drugu i zajedno čine kompletnu aplikaciju. Razvoj aplikacije se odvija u okruženju Android Studio, a rešenje je potrebno testirati na emulatoru.

Pred vama je tekst zadatka koji predstavlja četvrtu celinu. Ukupan broj bodova koje možete ostvariti na ovom zadatku je 10 bodova.

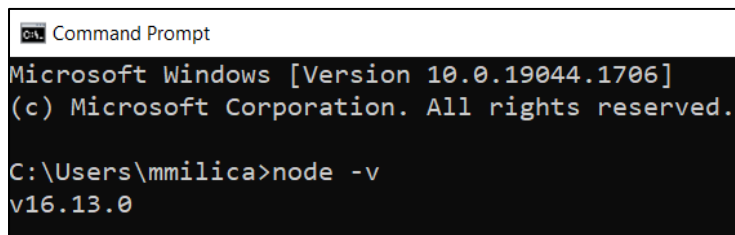
Implementaciju zadatka nastaviti na prethodno urađenom projektu.

HTTP Server

Uz zadatak je priložen server implementiran u Node.js programskom jeziku kao i API dokumentacija za isti. Kako bi se server pokrenuo potrebno je instalirati `node`.

Windows instalacija

Node se skida sa linka: <https://nodejs.org/en/> (skinuti poslednju (LTS) verziju). Pokrenuti instalaciju. Nakon instalacije, uveriti se da je node uspešno instaliran pokretanjem komande `node -v` u terminalu. Ukoliko je node uspešno instaliran trebalo bi da dobijete poruku o verziji:



```
Command Prompt
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mmilica>node -v
v16.13.0
```

Slika 1 - Provera verzije node-a

Napomena: moguće je da će se verzija razlikovati od verzije na slici.

Linux instalacija

U terminalu pokrenuti komandu za instalaciju:

`sudo apt install nodejs`

Uveriti se da je instalacija uspešna pokretanjem komande `node -v`. Ispis treba da bude sličan ispisu na slici iznad (odnosno, treba da se prikaže verzija instaliranog node-a).

Pokretanje servera

Kada je node uspešno instaliran, potrebno je raspakovati kod servera, a zatim instalirati potrebne biblioteke:

1. U terminalu, pozicionirati se na putanju na kojoj se nalazi server (ući u raspakovani direktorijum)

2. Pokrenuti komandu `npm install`.

Kada su biblioteke uspešno instalirane, server se pokreće komandom `npm run start`. Ukoliko je server uspešno pokrenut trebalo bi da se dobije ispis sličan ispisu na slici ispod:

```
D:\nastava\_1etnji\PNRS1\2022\projekat\PNRS1-MemoryGame-Server-main>npm run start

> pnrs-server@0.0.1 start
> nest start

[Nest] 23688 - 05/18/2022, 12:34:12 PM LOG [NestFactory] Starting Nest application...
[Nest] 23688 - 05/18/2022, 12:34:12 PM LOG [InstanceLoader] AppModule dependencies initialized +202ms
[Nest] 23688 - 05/18/2022, 12:34:12 PM LOG [InstanceLoader] MongooseModule dependencies initialized +1ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [InstanceLoader] MongooseCoreModule dependencies initialized +1258ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [InstanceLoader] MongooseModule dependencies initialized +30ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [InstanceLoader] MongooseModule dependencies initialized +1ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [InstanceLoader] AuthModule dependencies initialized +3ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [InstanceLoader] ScoreModule dependencies initialized +2ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RoutesResolver] AuthController {/auth}: +19ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RouterExplorer] Mapped {/auth/signup, POST} route +10ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RouterExplorer] Mapped {/auth/signin, POST} route +2ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RoutesResolver] ScoreController {/score}: +2ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RouterExplorer] Mapped {/score, GET} route +3ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RouterExplorer] Mapped {/score, POST} route +2ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [RouterExplorer] Mapped {/score, DELETE} route +2ms
[Nest] 23688 - 05/18/2022, 12:34:14 PM LOG [NestApplication] Nest application successfully started +9ms
```

Slika 2 - Pokretanje servera

Server je pokrenut na portu **3000**. Zahtevi ka server se, dakle, šalju na URL: <http://ipaddress:3000/>. Radi testiranja servera preporučujemo korišćenje aplikacije Postman (<https://www.postman.com/downloads/>).

Podaci se čuvaju u bazi koja je **deljena** među svim korisnicima. Svi ćete videti svačije unose u bazi, vodite računa da ne pretrpavate bazu (povremeno brišite igre koje upisujete dok testirate zadatak) ili da ne obrišete sve rezultate ostalima :).

Napomena: u dokumentaciji ćete videti da se na slanje HTTP POST zahteva u nekim slučajevima kao odgovor vraća objekat, međutim HTTP Helper klasa sa kojom smo radili ne čita ovaj odgovor. Klasu ne morate proširivati da čita odgovor, dovoljno je da proverom statusnog koda odgovora shvatite da li je došlo do greške ili ne. Naravno, ko želi, može i da proširi metodu za slanje POST zahteva tako da pročita i telo odgovore :) Takođe, voditi računa HTTP POST poruke vraćaju status **201** ukoliko je zahtev uspešno obrađen.

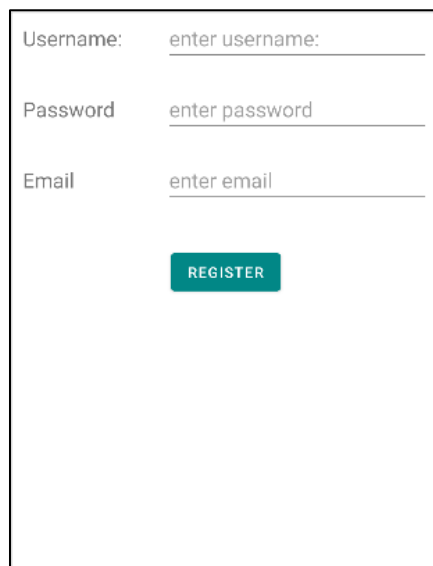
Nasumično raspoređivanje slika

Prilikom pokretanja nove igre slike treba nasumično rasporediti. Svaki restart igre (zatvaranje svih sličica) podrazumeva da će one biti nasumično raspoređene.

Registracija

Dodati novo dugme na početnoj strani. Klikom na ovo dugme otvara se nova aktivnost - *RegisterActivity*. Izgled aktivnosti kreirati po ugledu na sliku 3.

Klikom na dugme *Register* šalje se HTTP zahtev za registraciju korisnika (User - Register). Ukoliko je registracija uspešna (server vraća odgovor *201, Created*) vratiti korisnika na Login stranicu. U suprotnom prikazati Toast poruku greške.

The image shows a registration form within a rectangular frame. It contains three text input fields stacked vertically. The first field is labeled 'Username:' and has the placeholder text 'enter username:'. The second field is labeled 'Password' and has the placeholder text 'enter password'. The third field is labeled 'Email' and has the placeholder text 'enter email'. Below these fields is a teal-colored button with the text 'REGISTER' in white capital letters.

Slika 3 - Izgled aktivnosti RegisterActivity

Prijava

Klikom na dugme *Login* poslati HTTP zahtev za prijavu na server (User - Login). Ukoliko je prijava uspešna (server vraća odgovor *201, Created*) korisnik se preusmerava na aktivnost *GameActivity*. U suprotnom prikazuje se Toast poruka greške.

Čuvanje nove igre

Kada se igra završi, pored čuvanja u bazu dodati i logiku čuvanja rezultata na serveru (Game - Save game). Pokriti slučajeve kada server vrati grešku (prikazati Toast poruku greške). Logika čuvanja u bazi i dalje može da ostane.

StatisticActivity

Kada se pokrene aktivnost u listi se prikazuju trenutni podaci iz baze (ostaje ista logika kao u Zadatku 3).

Dodati dugme *Refresh* za osvežavanje podataka u listi. Klikom na ovo dugme šalje se zahtev serveru za dobavljanje SVIH rezultata (Game - Get all games). Rezultate dobavljene sa servera je potrebno sačuvati u SQLite bazi, a zatim nove rezultate iz baze upisati u listu (logika upisivanja iz baze ostaje ista kao u Zadatku 3).

Vodite računa da **ne čuvate duplikate** u bazi. Kada se klikne na dugme Refresh potrebno je obrisati postojeće podatke u lokalnoj (SQLite) bazi, a zatim ponovo upisati sve podatke dobijene od servera.

Dobavljanje svih rezultata jednog korisnika

Klikom na korisnika otvara se aktivnost *DetailsActivity* unutar koje se prikazuje lista svih rezultata jednog korisnika. Ovi rezultati se ne moraju čuvati u bazi. Dovoljno ih je prikazati direktno sa servera.

Brisanje rezultata

Korisnik koji je ulogovan sme da obriše **samo svoj rezultat**! Sakriti dugme za brisanje rezultata pored drugih korisnika (podesiti ili visibility ili enabled atribut). Klikom na dugme briše se unos u bazi (implementirano u Zadatku 3) kao i unos na serveru (šalje se HTTP zahtev za brisanje rezultata - Game - Delete all games for user). Naravno, prilikom brisanja rezultata igrača, isti se briše i iz liste.

VAŽNE NAPOMENE

- Koristiti android.developers kao vid dokumentacije, pored materijala sa vežbi i predavanja.
- Kako biste ostvarili maksimalan broj bodova, morate ispoštovati sve zahteve. Prostor za individualnost svakako ostaje na izboru boja, dimenzija teksta i izgleda aktivnosti.
- Sve boje i stringovi su resursi aplikacije, te je neophodno da budu definisani u strings.xml i colors.xml datotekama.
- Voditi računa o formatiranju koda.
- Rok za izradu je **sreda, 25.05. u 07:00h za grupe 33 i 34, odnosno četvrtak, 26.05 u 07:00h za grupe 31 i 32**. Projekat sačuvati kao .zip arhivu sa nazivom Ime_Prezime_RA_XXX_YYYY, gde je XXX_YYYY broj indeksa. Rešenje poslati do kraja zadatog roka na SOVA portal pod zadatak **Projektni zadatak 4**. Link za upis na predmet: <https://sova.uns.ac.rs/course/view.php?id=3488>.
- U sredu i četvrtak u terminima vežbi će biti organizovane odbrane domaćeg zadatka. Studenti koji ne pošalju prethodno svoje rešenje, neće imati prava odrane zadatka.

SREĆNO!