

Cilj igre Spojnice je spojiti nazive iz leve tabele sa odgovarajućim nazivima iz desne tabele. Igra traje 60 sekundi. Igrač ima deset pokušaja da spoji odgovarajuće nazive. Za svaki uspešno spojeni par igrač dobija poen.

## Pokretanje aplikacije igra Spojnice

Aplikacija ima sledeće foldere: gameMicroservice, scoreMicroservice, userMicroservice i View

Za mikroservise potrebno je imati instalirano sledeće na računaru:

- <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>
- <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>
- <https://dotnet.microsoft.com/download/dotnet/5.0>

### Pokrenuti SQL Server

U svakom od mikroservisa zameniti string za konekciju - **DatabaseConnection**:

- \gameMicroservice\Game\appsettings.json
- \scoreMicroservice\Score\appsettings.json
- \userMicroservice\User\appsettings.json

Najjednostavniji način da se dodje do "connection string-a" jeste putem visual studija, pratite sledeća uputstva:

<https://social.msdn.microsoft.com/Forums/sqlserver/en-US/ba6b1757-301c-4545-98e4-69a81b33e876/find-connection-string-used-by-sql-server-management-studio>

Ući u svaki folder pojedinačno putem command prompta u koren foldera Game, Score i User i izvršiti komandu **dotnet ef database update** ona će pokupiti sta se nalazi u Migrations folderu i napraviti strukturu baze. Ukoliko Migration folder ne postoji ili je prazan ili pri promenom modela baze izvršiti komandu pre update : **dotnet ef migrations add {nazivMigracije}**. Ako nemate instaliran dotnet-ef, koristeći komandu : **dotnet tool install --global dotnet-ef** instalirajte ga.

Kako ne bi bilo potrebe ručno da se unose komande, napravljeni su skriptni fajlovi:

- **databaseUpdate.bat** koji će izvršiti *dotnet ef database update* komandu u svakom mikroservisu.
- **databaseMigrationsAdd.bat** koji će izvršiti *dotnet ef migrations add databaseMigration* komandu u svakom mikroservisu.

**Mikroservisi se pokreću** svaki od njih nezavisno od drugog. Otvoriti Command Promt i zatim preko command prompta ići do foldera koji sadrži *Program.cs* fajl koji je i startni fajl svakog mikroservisa.

- \gameMicroservice\Game
- \scoreMicroservice\Score
- \userMicroservice\User

Tako da otvoriti 3 command prompta preko njih ući u svaki folder od gore navedenih. Zatim u sva 3 command prompta kada se nadjemo u gore pomenutim direktorijuma radimo **dotnet run** koja će pokrenuti svaki mikroservis.

Dodatno postoji i **START.bat** fajl koji će ujedno pokrenuti svaki mikroservis kao i angular aplikaciju.

Sada nakon njihovog pokretanja ne bi bilo loše da se proba da se kontaktira svaki preko običnog browsera koji će biti korišćen za testiranje kako bi odobrili “nesigurnu” konekciju. Mikroservisi su postavljeni da budu **https** a kako sertifikat nije postavljen od strane reputabilne kompanije browser nas pita da to potvrdimo, svakako testiranje je lokalno tako da je prihvatljivo. Otvoriti browser i otići na sledeće URL adrese i potvrditi https konekciju:

- <https://localhost:5101/swagger/index.html> - User mikroservis
- <https://localhost:5201/swagger/index.html> - Score mikroservis
- <https://localhost:5301/swagger/index.html> - Game mikroservis

Swagger je aktivan tako da je moguće i da se testiraju API putanje nad mikroservisima bez potrebe za klijentskom aplikacijom.

### **Pokretanje frontend aplikacije - Angular**

Otvoriti još jedan command prompt i ući u folder: **\\View\\Angular\\game-app** zatim uraditi **ng run**

Pored ng run komande isto je moguće uraditi i sam build i nakon toga može aplikacija sama preko generisanih html, css i js fajlova da se pokrene bez potrebe za pokretanjem angular “servera”.

Ukoliko je angular pokrenut preko ng serve komande onda će angular klijentska aplikacija biti dostupna na *localhost:4200* adresi.

## Dodatno upustvo

Ako ste ispratili navedena uputstva verovatno je da nemate kreiran ni jedan nalog. Potrebno je ručno da se napravi nalog tipa admin kako bi nakon logovanja mogli da pravite nove igre.

Admin nalog se pravi na jedan od sledećih načina:

- Ručnim unosom u bazu, npr. Preko microsoft sql server management studija
- Registrovanjem putem REGISTER opcije preko angulara, zatim ručnom zamenom *user* u *admin* kao tip naloga
- Putem REST API zahteva. Ovo može da se uradi na jedan od sledećih načina:
  - pisanjem neke skripte koja to radi
  - putem aplikacija poput postman-a
  - ili čak putem swagger-a koji je aktiviran na svakom mikroservisu.
    - <https://localhost:5101/swagger/index.html> - adresa ka User mikroservisu sa swagger servisom, ovde samo ići na /create POST metod i uneti parametre za username, password, email i role-u. Role parametar staviti kao admin kako bi aplikacija prepoznala i dala privilegije da imate mogućnost da pravite nove igre, da ih menjate i brišete putem interfejsa.

Slanje emailova je uvezano putem SMTP protokola. Tako da možete povezati da radi i sa drugim mail nalogom. Ukoliko želite da promenite sa kog maila se šalju poruke, potrebno je postaviti varijable u klasama navedenih foldera:

- \gameMicroservice\Game\Data\UserEmailSender.cs
- \scoreMicroservice\Score\Data\UserEmailSender.cs
- \userMicroservice\User\Data\UserEmailSender.cs