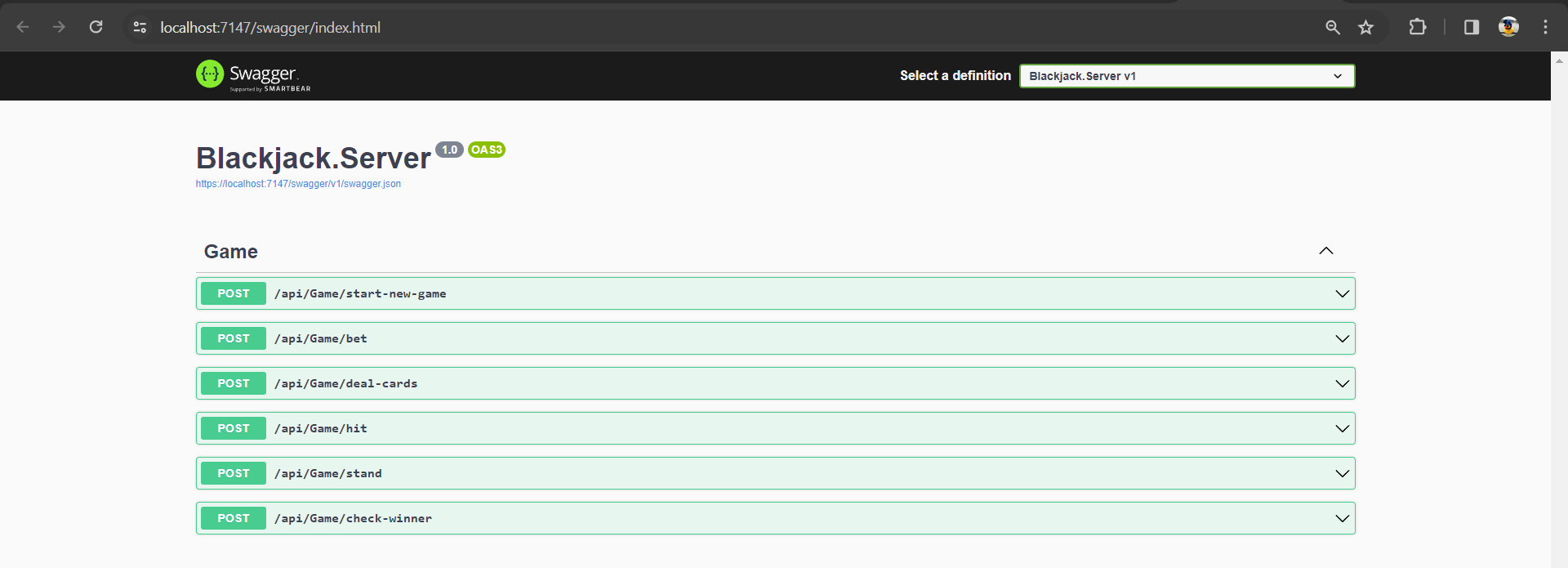
Blackjack

**Prezentare generală**

1. **Front-end**
2. **Back-end**



Backend-ul, realizat cu ASP.NET Core și .NET 8.0, constituie nucleul logic al aplicației și furnizează mai multe endpoint-uri esențiale pentru crearea și desfășurarea unui joc de Blackjack:

O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat

**start-new-game** - Acest endpoint inițiază un joc nou de blackjack. Este responsabil pentru începerea unei noi runde, resetarea stării jocului și pregătirea pentru o nouă rundă de pariere.

**Raspuns**: Stadiul jocului. (mana jucatorului, a dealer-ului si datele jucatorului)

**bet** - Aici se plasează pariurile pentru runda curentă de blackjack. Jucătorul poate specifica suma pe care dorește să o parieze pentru a participa la mâna respectivă.

**Raspuns**: Mesaj text care precizeaza dacă pariul a fost plasat cu succes.

**deal-cards** - Acest endpoint împarte cărțile către jucător și dealer în cadrul unei runde. Jucătorul primește inițial două cărți, iar dealerul primește o carte vizibilă și una ascunsă.

**Raspuns**: Stadiul jocului. (mana jucatorului, a dealer-ului si datele jucatorului)

**hit** - Atunci când jucătorul dorește să primească o carte suplimentară din pachet, apelează acest endpoint. Aceasta îi oferă o nouă carte jucătorului în încercarea de a se apropia cât mai mult de valoarea 21 fără a o depăși.

**Raspuns**: Stadiul jocului. (mana jucatorului, a dealer-ului si datele jucatorului)

**stand** - Jucătorul folosește acest endpoint pentru a rămâne cu cărțile pe care le are în mână, indicând că nu mai dorește cărți suplimentare. Acesta este pasul în care jucătorul decide să rămână cu mâna sa actuală și să lase dealerul să-și continue jocul.

**Raspuns**: Stadiul jocului. (mana jucatorului, a dealer-ului si datele jucatorului)

O imagine care conține text, software, Font, număr

Descriere generată automat

**check-winner** - După ce jucatorul și dealerul și-au ales acțiunile finale (hit sau stand), acest endpoint determină câștigătorul rundei. Verifică mâinile jucătorului și ale dealerului pentru a decide cine câștigă mâna conform regulilor blackjack-ului și distribuie câștigurile sau pierderile corespunzătoare.

**Raspuns**: Mesaj text. (“Player wins”, “Dealer wins” sau “Tie”)

**Structura fișierelor**

**O imagine care conține text, captură de ecran, software

Descriere generată automat** **O imagine care conține text, captură de ecran, Font

Descriere generată automat**

**Blackjack.Server** reprezinta logica aplicației și are o structură bine definită:

**Controllers** - reprezintă modulul care definește și gestionează endpoint-urile utilizate în interacțiunea cu aplicația.

**Models** - cuprinde clasele, enum-urile și interfețele folosite în logica jocului, precum și stările specifice ale jocului (faze precum Betting, Player turn, Dealer turn, Game over). Aici sunt și fabricile de cărți și pachete de cărți, detalii despre acestea puteți găsi în secțiunea de Design Patterns.

**Services** - acționează ca puntea de legătură între logica jocului și endpoint-urile definite în Controllers, facilitând interacțiunea dintre aceste componente.

Directorul **Blackjack.Tests** conține seturile de fișiere de test pentru modelele și alte componente din serverul de Blackjack. Aici sunt incluse testele unitare și de integrare pentru a evalua corectitudinea și funcționalitatea modelelor din aplicație.

Aceste teste sunt esențiale pentru asigurarea calității codului și pentru identificarea eventualelor probleme sau erori în implementare. Ele sunt proiectate pentru a verifica comportamentul și funcționarea corectă a claselor, enum-urilor și interfețelor din cadrul logicii jocului de Blackjack.

O imagine care conține text, captură de ecran, Font

Descriere generată automat

În plus față de fișierele de test, directorul poate conține și un folder separat numit TestResults, care este folosit pentru a stoca rezultatele testelor rulate în timpul procesului de testare. Acest folder poate conține rapoarte, fișiere de tip log sau alte informații relevante generate în urma rulării și evaluării testelor pentru a facilita analiza și remedierea eventualelor probleme identificate.

**Teste:**

**CardTests** – Verifica compotamentul si funcționalitatea corectă a clasei ‘Card’ din perspectiva valorii cărților.

**DealerHandTests** – Testele asociate clasei ‘DealerHand’ evalueaza modul în care dealer-ul gestionează și își formează mâna în timpul jocului.

**DealingPackTests** – Testele pentru clasa ‘Deck’ evaluează logica de creare, gestionare și utilizare a unui întreg set de pachete de cărți în cadrul jocului de Blackjack.

**DeckTests** – Acest set de teste verifică funcționalitatea corectă a clasei ‘Deck’, care include manipularea pachetului de cărți și amestecarea lor.

**PlayerHandTests** – Verifică comportamentul clasei ‘PlayerHand’, focalizându-se pe modul în care mâinile jucătorilor sunt gestionate, inclusiv adăugarea, evaluarea valorii mâinii și alte acțiuni relevante pentru mâna jucătorului.

**PlayerTests** - Testele asociate clasei ‘Player’ evalueaza setarea corectă a atributelor acestuia.

O imagine care conține text, captură de ecran, software, Software multimedia

Descriere generată automat