

1. Autentifikacija

Test Scenario 1.1: Uspešna autentifikacija sa validnim kredencijalima

- **Opis:** Proverava da li korisnik može da dobije auth token koristeći validne kredencijale.
- **Koraci:**
 1. Pošaljite POST zahtev na /auth endpoint sa ispravnim korisničkim imenom i šifrom prema podacima u API dokumentaciji.
 2. Očekujte da server vrati auth token.
- **Podaci za test:**
 - username: validan korisnički nalog (admin).
 - password: validna šifra (password123).
- **Očekivani rezultat:**
 - Status kod: 200 OK.
 - Body odgovora treba da sadrži ključ token sa validnim tokenom.
 - Na osnovu validnog tokena, korisnik bi trebalo da bude autorizovan za dalja testiranja.

Test Scenario 1.2: Neuspešna autentifikacija sa netačnim kredencijalima

- **Opis:** Proverava kako sistem reaguje kada korisnik pokuša da se autentifikuje sa netačnim kredencijalima.
- **Koraci:**
 1. Pošaljite POST zahtev na /auth endpoint sa netačnim korisničkim imenom ili šifrom.
 2. Očekujte da server odbije autentifikaciju.
- **Podaci za test:**
 - username: netačan korisnički nalog (npr. wrongUser).
 - password: netačna šifra (npr. wrongPassword).
- **Očekivani rezultat:**
 - Status kod: 401 Unauthorized.
 - Body odgovora treba da sadrži poruku o grešci, npr. "Bad credentials".
- **Zapažanja i bugovi:**
 - API ne vraća 401 Unauthorized kada pokušate autentifikaciju sa netačnim korisničkim imenom i šifrom. Umesto toga, vraća 200 OK.

2. CRUD Operacije

Test Scenario 2.1: Kreiranje novog entiteta (Rezervacija)

- **Opis:** Testira proces kreiranja nove rezervacije (npr. sobe ili automobila) pomoću POST zahteva.
- **Koraci:**
 1. Pošaljite POST zahtev na /booking endpoint sa validnim podacima za kreiranje rezervacije.
 2. Koristite validan auth token u Authorization headeru.
 3. Očekujte da server uspešno kreira rezervaciju.
- **Podaci za test:**
 - firstname: Ime korisnika koji pravi rezervaciju (npr. "Jim").
 - lastname: Prezime korisnika (npr. "Brown").
 - totalprice: Ukupna cena rezervacije (npr. 111).
 - depositpaid: Da li je depozit plaćen (true).
 - bookingdates.checkin: Datum kada korisnik planira da se prijavi (npr. "2018-01-01").
 - bookingdates.checkout: Datum kada korisnik planira da se odjavi (npr. "2019-01-01").
 - additionalneeds: Dodatne potrebe (npr. "Breakfast").
 - token: Auth token dobijen u prethodnim testovima.
- **Očekivani rezultat:**
 - Status kod: 201 Created ili 200 OK.
 - Body odgovora treba da sadrži podatke o rezervaciji, uključujući jedinstveni id rezervacije.
- **Ograničenja:**
 - API ne proverava validnost formata datuma, pa će rezervacije sa pogrešnim datumima biti uspešno kreirane, iako su ti datumi nevalidni.

Test Scenario 2.2: Čitanje entiteta (Rezervacija) putem ID-a

- **Opis:** Testira mogućnost čitanja podataka o rezervaciji koristeći njen ID.
- **Koraci:**
 1. Pošaljite GET zahtev na /booking/{id} endpoint koristeći ID rezervacije koja je prethodno kreirana.
 2. Koristite validan auth token u Authorization headeru.
 3. Očekujte da server vrati podatke o rezervaciji sa tačnim ID-em.
- **Podaci za test:**
 - id: ID rezervacije koja je prethodno kreirana.
 - token: Auth token dobijen u prethodnom testu.
- **Očekivani rezultat:**
 - Status kod: 200 OK.
 - Body odgovora treba da sadrži podatke o rezervaciji.

Test Scenario 2.3: Ažuriranje entiteta (Rezervacija)

- **Opis:** Testira mogućnost ažuriranja postojećeg entiteta (npr. promena datuma ili tipa resursa).
- **Koraci:**
 1. Pošaljite PUT ili PATCH zahtev na /booking/{id} endpoint sa izmenjenim podacima.
 2. Koristite validan auth token u Authorization headeru.
 3. Očekujte da server uspešno ažurira podatke o rezervaciji.
- **Podaci za test:**
 - firstname: Ime korisnika koji pravi rezervaciju.
 - lastname: Prezime korisnika.
- **Očekivani rezultat:**
 - Status kod: 200 OK.
 - Body odgovora treba da sadrži ažurirane podatke o rezervaciji.

- **Ograničenja:**
 - API vraća 403 Forbidden sa ispravnim Authorization header-om i validnim tokenom, što onemogućava proveru odgovora prilikom pokušaja editovanja rezervacije.
 - API ne proverava validnost formata datuma, pa će rezervacije sa pogrešnim datumima biti uspešno kreirane, iako su ti datumi nevalidni.

Test Scenario 2.4: Brisanje entiteta (Rezervacija)

- **Opis:** Testira proces brisanja entiteta i proverava da li rezervacija više nije dostupna.
- **Koraci:**
 1. Pošaljite DELETE zahtev na /booking/{id} endpoint sa validnim ID-em entiteta koji želite da obrišete.
 2. Koristite validan auth token u Authorization headeru.
 3. Očekujte da rezervacija bude uspešno obrisana.
 4. Pošaljite GET zahtev na isti ID da biste proverili da li je rezervacija obrisana.
- **Podaci za test:**
 - id: ID rezervacije koja se briše.
 - token: Auth token dobijen u prethodnim testovima.
- **Očekivani rezultat:**
 - Status kod za DELETE zahtev: 204 No Content ili 200 OK.
 - Status kod za GET zahtev nakon brisanja: 404 Not Found, jer entitet više ne postoji.
 - Body odgovora treba da sadrži poruku o grešci (npr. "Entity not found").
- **Ograničenja:**
 - API vraća 403 Forbidden sa ispravnim Authorization header-om i validnim tokenom, što onemogućava proveru odgovora prilikom pokušaja brisanja rezervacije.
- **Zapažanja i bugovi:**
 - API za verifikaciju obrisane rezervacije, zbog buga da prilikom brisanja rezervacije vraća 403 Forbidden, vraća 200 OK umesto 404 Not Found.

