Documentazione Auth

🔐 Processo di Autenticazione nel Sistema

Il flusso di autenticazione di **Trekkigram** è stato progettato per garantire **sicurezza**, **continuità d'accesso** e una **navigazione fluida** per l'utente. Di seguito viene descritto ogni passaggio.

🟡 1. Inizializzazione dell'Autenticazione

- L'utente visita `trekkigram.com`.
- Il sistema verifica automaticamente se è presente una **sessione attiva**.
- Se esiste una sessione valida → reindirizzamento alla **home**.

  Altrimenti → reindirizzamento alla **pagina di login**.

🔑 2. Login Utente

- L'utente inserisce **email** e **password** nel form.
- Il client invia i dati al server tramite una **richiesta POST**.
- Il server riceve la richiesta nel file `AuthGateway.php`.

🔍 3. Verifica delle Credenziali

- `AuthGateway.php` estrae le credenziali dalla richiesta.
- Il sistema controlla se l'email è presente nel database.
- La password viene **hashata** e confrontata con quella salvata.
- Se non coincidono → risposta con **errore di autenticazione**.

🧾 4. Creazione della Sessione

- In caso di successo, viene chiamata `generateSessionToken()`.
- Viene creato un **token casuale** con `random_bytes()`.
- Il token viene salvato nella sessione PHP con:
  - ID utente
  - Timestamp di creazione
  - Scadenza (default: **1 anno**)
- Il browser riceve un **cookie** con l'ID di sessione.

📦 5. Risposta al Client

- Il server restituisce:
  - Il **token di sessione**
  - I **dati utente** principali (es. ID, nome)
- Il client salva queste info nel **localStorage**.
- L'utente viene reindirizzato alla **dashboard** o alla **homepage**.

🔄 6. Richieste Autenticate

- Per ogni richiesta futura a un'API protetta:
  - Il browser invia **automaticamente il cookie** di sessione.
  - Il server chiama `verifySessionToken()` per verificare validità e scadenza.
  - Se tutto è ok → la richiesta viene autorizzata.

🛡️ 7. Controlli di Sicurezza

Il sistema verifica:

- Che la sessione **esista**
- Che l'**ID utente** sia valido
- Che la sessione **non sia scaduta**

❌ Se uno di questi controlli fallisce → viene restituito un **errore 401 – Non autorizzato**.

⏳ 8. Gestione della Scadenza

Quando la sessione è scaduta:

- Il server **distrugge la sessione**
- Viene restituito un **401**
- Il client intercetta l'errore
- L'utente viene reindirizzato al **login**

🚪 9. Logout

- L'utente clicca su **Logout**
- Il server esegue `session_unset()` e `session_destroy()`
- Il client rimuove i dati salvati in `localStorage`
- L'utente viene riportato alla **pagina di login**

✅ Questo processo assicura che **solo gli utenti autenticati** possano accedere alle risorse riservate, mantenendo un'esperienza d'uso fluida e sicura.