

# Documentazione Backend

## Tecnologie Utilizzate

Per il backend sono state utilizzate le seguenti tecnologie:

- **Express:** per implementare un servizio REST API.
- **Prisma:** per la gestione dei modelli e l'esecuzione di query sul database.
- **PostgreSQL:** database relazionale per la memorizzazione dei dati ricevuti dall'app mobile.

## Struttura del Backend

Il backend è strutturato nei seguenti componenti principali:

- **Router:** entry point per le chiamate API, instrada le richieste ai rispettivi controller.
- **Auth Middleware:** verifica l'autorizzazione delle richieste in entrata.
- **Controller:** valida i dati delle richieste e delega l'elaborazione ai servizi.
- **Service:** contiene la business logic per la generazione delle risposte.

Inoltre troviamo un ErrorHandler per centralizzare la gestione degli errori.

## Altre Tecnologie Utilizzate

- **Zod:** utilizzato per validare i parametri e i dati ricevuti nelle richieste.
- **API Key Authorization:** metodo di autenticazione per le chiamate API.

## Struttura del Database

Le principali entità del database sono:

- **User:** memorizza le informazioni dell'utente, inclusi email e l'hash della password per il login, ed ha i riferimenti al **Wallet** e al **Budget**.
- **Wallet:** rappresenta il portafoglio dell'utente e contiene i riferimenti alle spese effettuate.
- **Expense:** registra le spese, suddivise in **Entrate** e **Uscite** e categorizzate per tipologia (es. Lavoro, Shopping, ecc.).
- **Budget:** permette all'utente di impostare un budget per ciascuna categoria di spesa.

Per semplificare l'implementazione, il **Budget** è associato direttamente all'utente anziché ai singoli Wallet.

---

# Documentazione Frontend

## Tecnologie Utilizzate

Per il frontend sono state utilizzate le seguenti tecnologie:

- **React Query**: utilizzato per la gestione delle chiamate API. Questa libreria ci aiuta a implementare l'**optimistic caching pattern** per migliorare l'esperienza utente.
- **Redux Toolkit**: impiegato per la gestione dello stato globale, utile per salvare dati non strettamente legati alle risposte delle API.
- **MMKV**: uno store sincrono con supporto alla crittografia dei dati, nel nostro caso usato per salvare il token di sessione.
- **Expo**: per gestire tutta infrastruttura mobile

## Struttura del Frontend

Gli attori principali del frontend sono:

- **Navigation**: gestisce la navigazione tra le schermate e la validazione dei parametri passati.
- **Components**: contiene tutti i componenti UI/UX, organizzati secondo il paradigma **Atomic Design**.
- **Provider**: utilizzato per implementare una **Bottom Sheet**. Il provider è montato sopra la navigazione questo permettendo al drawer di aprirsi sopra la bottom navigation. In oltre e richiamabile da tutti I componenti.
- **Models**: raccoglie tutti i modelli / interfacce utilizzate per le chiamate API.
- **Hooks**: contiene la business logic dell'applicazione.
- **Container**: funge da intermediario tra le **Screen** e i **Component**.
- **Api**: definisce tutte le chiamate verso il backend.
- **Service**: usato per **wrappare** alcune librerie, come MMKV, per renderne l'utilizzo indipendente dall'implementazione.

Per il LOGIN usare le seguenti credenziali:

- USER: [test.test@gmail.com](mailto:test.test@gmail.com)

- PASSWORD: 123456

---

## Tempi di Sviluppo

- **Stima**: 7d
-