# 

[**Auth**](#_p7v709rgp96x) **1**

[**Transactions**](#_cr5v6w7g5bqr) **1**

[Group](#_2aeo0evzpoe) **1**

[**Stats**](#_fbomujzec7wn) **1**

# 

# **Auth**

**POST /api/login**

**POST /api/users**

**GET /api/refresh**

# **Transactions**

**POST /transactions**

Crea una nuova transazione o un nuovo trasferimento (anche verso un goal)

**PATCH /transaction/{id}**

Modifica una transazione

**DELETE /transaction/{id}**

Cancella una transazione

**GET /transactions**

Ottiene un elenco di transazioni filtrate in base al filtro specificato come parametro, il filtro deve contenere un account per poter controllare i permessi. Ritorna anche i dati degli account coinvolti.

**GET /accounts**

Ottiene tutti gli account dell’utente loggato (esclusi gli account dei goal dell’utente)

**GET /account/{id}**

Ottiene le informazioni dell’account (transazioni escluse)

**PATCH /account/{id}**

Modifica il nome di un account

**DELETE /account/{id}**

Elimina un account (lo marca come eliminato probabilmente)

**POST /accounts**

Crea un account associato all’utente loggato

# **Group (goals, budget)**

**POST /goals**

Crea un goal per l’utente attualmente loggato (con il relativo account del goal)

**PATCH /goal/{id}**

Modifica un goal esistente (controllare che sia di proprietà dell’utente), il nome è modificabile senza restrizioni, l’importo si può modificare a patto di non scendere sotto i soldi già stanziati per il goal, la data di scadenza si può modificare senza però impostarla ad un giorno precedente ad oggi, il resto non si può modificare.

**DELETE /goal/{id}**

Elimina un goal (controllare che sia di proprietà dell’utente), forse più che eliminarlo lo marca come cancellato senza rimuoverlo dal DB, al fine di preservare la storia delle transazioni.

**GET /goals**

Ritorna tutti i goal dell’utente loggato

**POST /budgets**

Aggiunge un budget per una categoria ai budget dell’utente loggato, se è già presente un budget per quella categoria da un’errore. Se viene inserito il budget per una sottocategoria si deve controllare che la somma dei budget delle sottocategorie sia minore o uguale al budget della categoria genitore. NOTA: non si può creare una sottocategoria di un’altra sottocategoria.

**PATCH /budget/{id}**

Modifica l’importo di un budget (solo l’importo), si devono fare controlli per fare in modo che la somma dei budget delle sotto categorie rimanga minore o uguale a quella delle categorie genitori.

**DELETE /budget/{id}**

Elimina il budget per una categoria. Non si può eliminare il budget di una categoria che ha delle sottocategorie per cui è definito un budget. Controllare che il budget appartenga all’utente loggato.

**GET /budgets**

Ottiene l’elenco dei budget dell’utente

NOTA: le categorie sono fisse, vengono inserite nel DB una sola volta dagli amministratori di sistema (questo nella implementazione, ma non nell’ipotetico progetto completo).

# **Stats**

usare il prefisso /api/stats/ per tutte le statistiche, per semplificare la gestione del proxy.

**NOTE GENERALI:**

Quando si crea un oggetto con la POST ritornare l’oggetto creato completo di id e altre informazioni inserite dal server. Usare http status 201.