API – Progetto Base

STRUTTURE DATI

Frequency: enumerable

* “once”
* “daily”
* “weekly”
* “monthly”

User:

* id: ObjectId
* username: String,
* password: String

(per ora solo questi per l’utente)

Event:

* id: ObjectId (mongoDB)
* title: String,
* owner: User
* participants: [User]
* frequency: Frequency
* location: String
* startTime: Date
* endTime: Date

# EVENTI (/events)

Tutti i path devono essere autenticati. Tutti gli input devono essere validati (query, params, body).

Res.status = 402 se non autorizzato

Res.message = “Non autorizzato”

## GET /events/?from=:from&to=:to

Ritorna gli eventi dove l’utente autenticato è l’owner o uno dei partecipanti, filtrati per il range di data (nessun range di date -> ritorna tutti).

### Query

query.from = data iniziale dove fare la ricerca degli eventi

query.to = data finale di filtro degli eventi

### Req

{}

### Res 200

{status: “ok”, value: [Eventi]}

## GET /events/:id

Ritorna l’evento con id = params.id, se l’evento è dell’utente

### Params

id = id dell’evento da ritrovare

### Req

req.body = {}

### Res

res.body = [Eventi]

## POST /events/

Inserisce un nuovo evento per l’utente loggato.

Req

body: Event

Res

{status: ok, value: 1}

## PUT /events/:id

Modifica l’evento con id = params.id, se è dell’utente autenticato.

Req

body: Event

Res

{status: ok, value: 1}

## DELETE /events/:id

Elimina l’evento con id = params.id, se è l’utente autenticato è l’owner

Req

body: Event

Res

{status: ok, value: 1}