

List of microservices and rest endpoints

Guideline: HTTP methods for REST - GET: get a single resource or a collection - POST: create a new resource in a collection - PUT: update/replace a single resource - DELETE: delete a single resource from a collection

NOTE:

tutti gli endpoint, ad eccezione di `/session` e `/users` richiedono token per la validazione, inviato tramite header HTTP.

Le risposte degli endpoint devono restituire codici HTTP più adatti possibile, cioè che diano informazioni rispetto allo stato della richiesta.

User

Endpoints

Endpoint per la **gestione della sessione**:

- `/sessions`
 - POST
Receives: username, password in HTTP request body
Returns: `session_id` (something like `authorization: bearer <token>`)
- `/session/<session_id>`
 - GET: used by other services for checking for authentication
Returns: HTTP codes 204 o 404
 - DELETE: execute logout by deleting a session

Endpoint per gestire gli **utenti** - `/users` - POST: crea nuovo utente (popolato)

- `/user/<id>`
 - GET: returns data of the specified user
 - PUT: aggiornare dati utente (forse rimarrà inutilizzato)

Resource fields

- id
- email
- password
- nome
- cognome
- organizer: boolean

Organization

Endpoints

- `/organizations[?category]`

- GET: restituisce tutte le organizzazioni
 - * **?category**: (forse non da implementare) restituisce tutte le organizzazioni per una categoria specificata
- POST: crea nuova organizzazione (popolata) restituisce **org_id**
- **/organizations/categories**
 - GET: restituisce la lista di categorie
- **/organization/<id>**
 - GET: restituisce dati dell'organizzazione specificata

Resource fields

- id
- owner_id
- org_name
- phone
- address
- desc
- category

Event

Endpoints

- **/events[?organization_id]**
 - GET: restituisce tutti gli eventi
 - * **?organization_id**: restituisce tutti gli eventi dell'organizzazione specificata
 - POST: crea nuovo evento (popolato)
Restituisce: **event_id** of the newly created event
- **/event/<id>**
 - GET: get the specified event
- **/event/<id>/reservations**
 - GET: restituisce la lista delle prenotazioni effettuate per lo specifico evento

Fields - id - organization_id - title - description - date - start_time - end_time
- price - capacity

Reservations

Endpoints

- **/reservations[?user_id | ?event_id]**
 - GET: return all reservations
 - * **?user_id**: returns all reservations done by the specified user
 - * **?event_id**: returns all the reservations done for the specified event

- POST: new reservation (popolata).
Returns: **reservation_id** of newly created reservation
- /reservation/<reservation_id>
 - GET
 - PUT: update the reservation
 - DELETE

Resource fields

- id
- transaction_id (se pagato è popolato)
- user_id
- event_id

Payments

Endpoints

- /transactions[?user_id]
 - GET
 - * ?user_id: get all transactions done by the specified user
 - POST: crea nuova transazione (popolata)
Returns: **transaction_id** or HTTP code 402 if fails
- trasaction/<id>
 - GET
 - PUT

Resource fields

- id
- user_id
- price
- card_number
- card_holder
- cvv
- exp_date
- timestamp (data, ora)