

Backend API

Tutti i file riguardanti il backend si troveranno nella cartella `api` , in modo che i link siano del tipo `sito.com/api/[endpoint]` , per organizzare meglio le pagine

Per la documentazione

Se in futuro si decide di modificare l'api, si può optare di usare uno schema url del tipo `sito.com/api/[versione]/[endpoint]` , esempio:
`rollo.com/api/v2/auth/login.php`

Indice

- [Errori](#)
- [Autenticazione](#)
 - [login.php](#)
 - [register.php](#)
 - [logout.php](#)
- [Utenti](#)
 - [get.php](#)
 - [update.php](#)
- [Sedi](#)
 - [list.php](#)
 - [add.php](#)
 - [update-sedi.php](#)
- [Biciclette](#)
 - [disponibili.php](#)
 - [revisionare.php](#)
 - [stato.php](#)
- [Noleggi](#)
 - [storico.php](#)
 - [avvia.php](#)
 - [termina.php](#)

Errori

I messaggi di errore vengono comunicati tramite 3 chiavi json:

- `ok` `bool` : sempre `false` per gli errori, `true` per tutto il resto

- `msg string` : contiene il messaggio di errore da mostrare all'utente
- `error string facoltativo` : informazioni aggiuntive per lo sviluppatore

Lo status code sara' di tipo `400-499` per richieste malformate e `500-599` per errori lato server (in questo caso non serve precisare `msg` , si puo lasciare vuoto e comunicare l'errore solo con `error`)

Esempio: richieste che richiedono una sessione valida, nel caso di utente non autenticato, possono restituire status code `401` (Unauthorized), oppure per dati mancanti si puo usare `400` (Bad request)

Richieste andate a buon fine useranno lo status code `200` (gia gestito da Apache) o [quello ritenuto opportuno](#))

```
{
  "ok": false,
  "msg": "...",
  "error": "..."
}
```

Autenticazione

- URL Base: `/api/auth/`

login.php

- **Metodo:** `POST`

Si occupa di far fare il login all'utente, controlla le informazioni fornite (email, password) e le confronta con quelle nel database

Dati in entrata	Richiesto	Descrizione
email	✓	email dell'utente
password	✓	password dell'utente

esempio risposta:

```
{
  "ok": true,
  "user": {
    "id": "...",
    "nome": "John",
    "cognome": "Doe",
    "email": "johndoe@email.com",
    "nToken": 325
  }
}
```

```
}  
}
```

register.php

- **Metodo:** POST

Permette di registrare un nuovo utente nel database (prima controlla se l'utente non esiste già: email già usata)

Dati in entrata	Richiesto	Descrizione
email	✓	email dell'utente
password	✓	password dell'utente
nome	✓	nome dell'utente
cognome	✓	cognome dell'utente

esempio risposta:

```
{  
  "ok": true,  
  "user": {  
    "id": "uuid",  
    "name": "John",  
    "surname": "Doe",  
    "email": "johndoe@email.com",  
    "nToken": 325  
  }  
}
```

logout.php

- **Metodo:** POST

Invalida la sessione dell'utente

esempio risposta:

```
{  
  "ok": true,  
  "user": {  
    "id": "uuid",  
    "name": "John",  
    "surname": "Doe",  
    "email": "johndoe@email.com",  
    "nToken": 325  
  }  
}
```

```
}  
}
```

Utenti

- URL Base: /api/utenti/

get.php

- Metodo: GET

Parametro GET	Richiesto	Descrizione
id	✗	UUID dell'utente

esempio risposta:

```
{  
  "ok": true,  
  "users": [  
    {  
      "id": "uuid",  
      "nome": "Mario",  
      "cognome": "Rossi",  
      "email": "mario@rossi.com",  
      "nToken": 120,  
      "ruolo": "utente"  
    }  
  ]  
}
```

update.php

- Metodo: POST

Parametro	Richiesto	Descrizione
id	✓	UUID dell'utente
nome/cognome/email/password	✗	Campi aggiornabili

esempio risposta

```
{  
  "ok": true,
```

```
    "user": {
      "id": "uuid",
      "nome": "Mario",
      "cognome": "Rossi",
      "email": "mario@rossi.com",
      "nToken": 120,
      "ruolo": "utente"
    }
  }
```

Sedi

- URL Base: /api/sedi/

list.php

- **Metodo:** GET
Restituisce l'elenco delle sedi.

esempio risposta:

```
{
  "ok": true,
  "sedi": [
    {
      "id": "...",
      "nome": "Sede Nord",
      "comune": "Padova",
      "cap": 12345,
      "via": "popa",
      "civico": 123,
      "num_biciclette_gestite": 59,
      "telefono" : "123456789"
    }
  ]
}
```

add.php

- **Metodo:** POST
- **JSON:** false (lato admin, sito web)
Aggiunge una nuova sede.

update-sedi.php

- **Metodo:** POST

- **JSON:** `false` (lato admin, sito web)

Aggiorna una sede esistente.

Biciclette

- URL Base: `/api/biciclette/`

disponibili.php

- **Metodo:** `GET`

Restituisce le biciclette registrate nel database appartenenti ad una determinata sede

Parametro	Richiesto	Descrizione
sede	✓	ID sede numerico

esempio risposta:

```
{
  "ok": true,
  "biciclette": [
    {
      "id": "...",
      "modello": "vtp-dot1q",
      "iot": {
        "latitudine": 50.034370,
        "longitudine": 19.210370,
        "data_installazione": "2020-12-25",
        "data_revisione": "2025-03-14"
      },
      "sede_appartenenza": "Sede Nord",
      "stato": true
    }
  ]
}
```

revisionare.php

- **Metodo:** `GET`

Restituisce biciclette che necessitano revisione entro una data.

Parametro	Richiesto	Descrizione
data	✓	Data YYYY-MM-DD

esempio risposta:

```
{
  "ok": true,
  "biciclette": [
    {
      "id": "...",
      "modello": "vtp-dot1q",
      "iot": {
        "latitudine": 50.034370,
        "longitudine": 19.210370,
        "data_installazione": "2020-12-25",
        "data_revisione": "2025-03-14"
      },
      "sede_appartenenza": "Sede Nord",
      "stato": true
    }
  ]
}
```

stato.php

- **Metodo:** POST

Permette di modificare stato e posizione delle biciclette.

Parametro	Richiesto	Descrizione
id	✓	id della bicicletta di cui modificare le informazioni
latitudine	✓	nuova latitudine della bicicletta
longitudine	✓	nuova longitudine della bicicletta
stato	✓	stato della bicicletta (true=occupata, false=disponibile)

esempio risposta:

```
{
  "ok": true
}
```

Noleggi

- URL Base: /api/noleggi/

storico.php

- **Metodo:** GET

Restituisce lo storico noleggi per utente.

Parametro	Richiesto	Descrizione
id	✓	UUID dell'utente

esempio risposta:

```
{
  "ok": true,
  "noleggi": [
    {
      "id": "...",
      "data": "27-01-2025",
      "ora_inizio": "14:34",
      "ora_fine": "15:07",
      "distanza_percorsa": 15.0,
      "prezzo": 40,
      "utente": "...",
      "bicicletta": "..."
    }
  ]
}
```

avvia.php

- **Metodo:** POST

Inizia un nuovo noleggio.

Parametro	Richiesto	Descrizione
utente	✓	UUID
bicicletta	✓	ID numerico bicicletta
ora_inizio	✗	default: ora corrente

esempio risposta:

```
{
  "ok": true,
  "noleggio": {
    "id": "...",
    "data": "27-01-2025",
    "ora_inizio": "14:34"
```



```
}  
}
```

termina.php

- **Metodo:** POST

Termina un noleggio in corso.

Parametro	Richiesto	Descrizione
id_noleggio	✓	ID noleggio
ora_fine	✗	default: ora attuale
distanza	✓	distanza percorsa in km

esempio risposta:

```
{  
  "ok": true,  
  "noleggio": {  
    "id": "...",  
    "data": "27-01-2025",  
    "ora_inizio": "14:34",  
    "ora_fine": "15:07",  
    "distanza_percorsa": 15.0,  
    "prezzo": 40,  
    "utente": "...",  
    "bicicletta": "..."  
  }  
}
```