

LOGIN UTENTE

URL: /auth/login

VERBO: POST

prende in input un oggetto contenente mail e password di un utente e restituisce un oggetto contenente il relativo token di sessione/autenticazione.

LOGOUT UTENTE

URL: /auth/logout

VERBO: POST

Prende in input un token di sessione/autenticazione e chiude la relativa sessione, invalidando il token.

INSERIMENTO NUOVO UTENTE NEL SISTEMA

URL: /users

VERBO: POST

Inserisce un nuovo utente nel sistema. L'oggetto passato come payload conterrà tutti i dati del profilo utente, ad esempio:

```
{
  "nome": "pinco",
  "cognome": "pallino",
  "email": "pp@gmail.it",
  "pwd": "abcdef",
  "dataDiNascita": "1995-04-27",
  "città": "Roma",
  "telefono": "3456789012",
  "titoloDiStudi": "Laurea Base in Informatica",
  "imgProfilo": "profilo.png"
}
```

ESTRAZIONE UTENTE PER CHIAVE:

URL: /users/{ID}

VERBO: GET

Restituisce il profilo dell'utente {ID} sotto forma dello stesso oggetto utilizzato come payload per l'inserimento (POST) tramite la URL /users, ad esclusione della password.

AGGIORNAMENTO PROFILO UTENTE (città,telefono,titolo di studio) PER CHIAVE:

URL: /users/{ID}

VERBO: PUT

Aggiorna il profilo dell'utente {ID}. Il payload è lo stesso oggetto utilizzato per l'inserimento (POST) tramite la URL /users eccetto: nome,cognome,email,pwd,dataDiNascita.

INSERIMENTO RIPETIZIONE:

URL: /repetitions

VERBO: POST

Inserisce una nuova ripetizione nel sistema. L'oggetto passato come payload conterrà tutti i dati della ripetizione, ad esempio:

```
{
  "tutor_key":1,
  "città":"Roma",
  "luogoIncontro":"a scelta dello studente",
  "costo":12,
  "descr":"descrizione della ripetizione",
  "categoria_key":1,
  "materie":[
    { "nome":"programmazione python"},
    { "nome":"database"}
  ]
}
```

ESTRAZIONE RIPETIZIONE PER CHIAVE:

URL: /repetitions/{ID}

VERBO: GET

Restituisce le informazioni relative alla ripetizione {ID} sotto forma dello stesso oggetto utilizzato come payload dell'inserimento (POST) tramite la url /repetitions, con in più le informazioni relative alla categoria a cui la ripetizione appartiene e quelle relative al tutor che la offre.

AGGIORNAMENTO RIPETIZIONE PER CHIAVE (prezzo/ora,descrizione,luogo incontro, lista materie):

URL: /repetitions/{ID}

VERBO: PUT

Aggiorna le informazioni relative alla ripetizione {ID}. Il payload è lo stesso oggetto utilizzato come payload dell'inserimento (POST) tramite la url /repetitions, ad esclusione di "tutor_key".

CANCELLAZIONE RIPETIZIONE PER CHIAVE:

URL: /repetitions/{ID}

VERBO: DELETE

Elimina la ripetizione {ID} dal sistema.

LISTA RIPETIZIONI PER FILTRO (varie combinazioni possibili):

URL: /repetitions?[{CITY}]&[{CATEGORY}]&[{SUBJECT}]&[{TUTOR_ID}]

VERBO: GET

Restituisce la lista di tutte le ripetizioni presenti nel sistema filtrate per:

- città
- città e categoria
- città, categoria e materia
- tutor

sottoforma di lista di oggetti della stessa forma di quelli restituiti dal recupero (GET) di una ripetizione tramite la url /repetitions/{ID}

INSERIMENTO PRENOTAZIONE RELATIVA AD UNA RIPETIZIONE:

URL: repetitions/{ID}/bookings

VERBO: POST

Inserimento nel sistema di una nuova prenotazione alla ripetizione {ID}. L'oggetto passato come payload conterrà tutti i dati della prenotazione, ad esempio:

```
{  
  "data": "2007-09-25 15:40:00",  
  "materia_key": 2,  
  "stato": 0,  
  "descr": "ripetizione descrizione",  
  "studente_key": 1  
}
```

dove studente_key è la chiave dello studente che sta richiedendo la prenotazione e materia_key è la chiave della materia (tra quelle contenute nella ripetizione) sulla quale vuole ricevere la lezione.

AGGIORNAMENTO PRENOTAZIONE RELATIVA AD UNA RIPETIZIONE:

URL: repetitions/{ID}/bookings

VERBO: PUT

Aggiornamento dello stato, della data e, eventualmente, del voto e della recensione. L'aggiornamento del voto e della recensione serve in realtà ad inserire nel sistema un feedback relativo alla ripetizione effettuata, poichè le informazioni relative al feedback sono contenute nell'oggetto prenotazione. L'oggetto passato come payload avrà lo stesso formato di quello usato per l'inserimento (POST) di una prenotazione tramite la url repetitions/{ID}/bookings, avente eventualmente in aggiunta i campi voto e recensione.

ESTRAZIONE PRENOTAZIONI PER CHIAVE STUDENTE:

URL: users/{ID}/bookings

VERBO: GET

Restituisce la lista delle prenotazioni effettuate dallo studente {ID} sotto forma dello stesso oggetto utilizzato come payload dell'inserimento (POST) tramite la url /repetitions/{ID}/bookings, con l'aggiunta delle informazioni riguardanti la ripetizione relativa alla prenotazione e il tutor che la offre. (Questa API serve a recuperare la lista delle prenotazioni effettuate da uno studente per le quali lo studente non ha ancora rilasciato un feedback.)

LISTA FEEDBACK RICEVUTI DA UN TUTOR:

URL: /users/{ID}/feedbacks

VERBO: GET

Restituisce la lista dei feedback ricevuti dal tutor {ID} (ovvero la lista degli oggetti prenotazioni relativi a ripetizioni offerte dal tutor {ID} per le quali sono stati rilasciati dei feedback).

ESTRAZIONE VOTO MEDIO TUTOR:

URL: /users/{ID}/feedbacks/vote

VERBO: GET

Restituisce il voto medio dell'utente calcolato sulla base dei voti contenuti in ogni feedback da esso ricevuto.

RECUPERO LISTA MATERIE PER CATEGORIA:

URL: /categories/{ID}/subjects

VERBO: GET

Restituisce la lista delle materie appartenenti alla categoria {ID}.