

Documentazione API

A cura di: Giulia Kalemi, Chiara Leone

Indice

1. [Autenticazione](#)
2. [Mentee](#)
3. [Mentor](#)
4. [Messaggi](#)
5. [Pagamenti](#)
6. [Utenti](#)

Autenticazione

- Base URL: /api/auth
- 1.1 Login
- Autentica un utente esistente.
- **Endpoint:** POST /api/auth/login
- **Autenticazione richiesta:** No

Request Body:

```
{ "email": "string", "password": "string" }
```

Validazione:

- Email valida richiesta
- Password richiesta

Response (200 OK):

```
{ "success": true, "token": "jwt_token_string",
```

```
"user": {  
  "id": "number",  
  "email": "string",  
  "role": "Mentor | Mentee",  
  "nome": "string",  
  "cognome": "string"  
}
```

Registrazione Mentor

- Registra un nuovo mentor con upload CV.
- **Endpoint:** POST /api/auth/register/mentor
- **Autenticazione richiesta:** No
- **Content-Type:** multipart/form-data

Request Body:{

```
"nome": "string",  
"cognome": "string",  
"email": "string",  
"password": "string",  
"telefono": "string",  
"biografia": "string",  
"specializzazione": "string",  
"anni_esperienza": "number",  
"tariffa_oraria": "number",  
"cv_file": "file (PDF, DOC, DOCX)"
```

```
}
```

Validazione:

- Tutti i campi richiesti
- Email valida
- Password minimo 6 caratteri
- CV file richiesto (formato PDF, DOC, DOCX)
- Tariffa oraria ≥ 0

Response (201 Created):

```
{
```

```
"success": true,
```

```
"message": "Mentor registrato con successo",
```

```
"mentor": {
```

```
"id": "number",
```

```
"email": "string",
```

```
"cv_path": "string"
```

```
}
```

```
}
```

Registrazione Mentee

- Registra un nuovo mentee.
- Endpoint: POST /api/auth/register/mentee
- Autenticazione richiesta: No

Request Body:

```
{
```

```
"nome": "string",
```

```
"cognome": "string",
```

```
"email": "string",
```

```
"password": "string",
```

```
"telefono": "string",
```

```
"obiettivi": "string"
```

```
}
```

Validazione:

- Tutti i campi richiesti
- Email valida
- Password minimo 6 caratteri

Response (201 Created):

```
{
```

```
"success": true,
```

```
"message": "Mentee registrato con successo",
```

```
"mentee": {
```

```
"id": "number",
```

```
"email": "string"
```

```
}
```

```
}
```

Logout

- Effettua il logout dell'utente.
- **Endpoint:** POST /api/auth/logout
- **Autenticazione richiesta:** Sì (JWT Token)

Response (200 OK):

```
{
```

```
"success": true,
```

```
"message": "Logout effettuato con successo"
```

```
}
```

```
...
```

Mentee

- Base URL: `/api/mentee``

Ricerca Mentor (Pubblica)

- Cerca mentor disponibili con filtri.
- **Endpoint:** `GET /api/mentee/search``
- **Autenticazione richiesta:** No
- **Query Parameters:**
- `?specializzazione=string`
- `&nome=string`
- `&tariffaMax=number`
- `&rating=number`
- `&page=number`
- `&limit=number`

Response (200 OK):

```
{"success": true,
```

```
"mentors": [
```

```
{
```

```
"id": "number",
```

```
"nome": "string",
```

```
"cognome": "string",
```

```
"specializzazione": "string",
```

```
"tariffa_oraria": "number",
```

```
"rating_medio": "number",
```

```
"anni_esperienza": "number",
```

```
"biografia": "string"
```

```
}
```

```
],
```

```
"pagination": {
```

```
"page": "number",
```

```
"limit": "number",
```

```
"total": "number"
```

```
}
```

```
}
```

Profilo Mentor (Pubblico)

- Visualizza il profilo dettagliato di un mentor.
- **Endpoint:** GET /api/mentee/mentor/:id
- **Autenticazione richiesta:** No
- **URL Parameters:**
- id (number) - ID del mentor

Response (200 OK):

```
{
```

```
"success": true,
```

```
"mentor": {
```

```
"id": "number",
```

```
"nome": "string",
```

"cognome": "string",

"email": "string",

"specializzazione": "string",

"biografia": "string",

"anni_esperienza": "number",

"tariffa_oraria": "number",

"rating_medio": "number",

"numero_recensioni": "number",

"disponibilita": [

{

"giorno": "string",

"ora_inizio": "string",

"ora_fine": "string"

}

]

}

}

Area Riservata Mentee

- Accede all'area riservata del mentee.
- **Endpoint:** GET /api/mentee/area
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Response (200 OK):

{

"success": true,

```
"mentee": {
```

```
  "id": "number",
```

```
  "nome": "string",
```

```
  "cognome": "string",
```

```
  "email": "string",
```

```
  "telefono": "string",
```

```
  "obiettivi": "string",
```

```
  "prossimi_appuntamenti": "number",
```

```
  "totale_speso": "number"
```

```
}
```

```
}
```

Crea Prenotazione

- Prenota una sessione con un mentor.
- **Endpoint:** POST /api/mentee/booking
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Request Body:

```
{
```

```
  "mentor_id": "number",
```

```
  "data": "YYYY-MM-DD",
```

```
  "ora_inizio": "HH:MM",
```

```
  "ora_fine": "HH:MM",
```

```
  "note": "string (opzionale)"
```

```
}
```

Validazione:

- *mentor_id* richiesto
- *data futura* richiesta
- *ora_inizio* e *ora_fine* valide
- *durata minima* 30 minuti

Response (201 Created):

```
{
  "success": true,
  "message": "Prenotazione creata con successo",
  "appointment": {
    "id": "number",
    "mentor_id": "number",
    "data": "string",
    "ora_inizio": "string",
    "ora_fine": "string",
    "stato": "in_attesa",
    "costo": "number"
  }
}
```

Stato Appuntamenti

- *Visualizza tutti gli appuntamenti del mentee.*
- **Endpoint:** `GET /api/mentee/appointments``
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)
- **Query Parameters:**
- `?stato=in_attesa|confermato|completato|cancellato`
- `&data_da=YYYY-MM-DD`

- &data_a=YYYY-MM-DD

Response (200 OK):

```
{  
  "success": true,  
  "appointments": [  
    {  
      "id": "number",  
      "mentor": {  
        "id": "number",  
        "nome": "string",  
        "cognome": "string"  
      },  
      "data": "string",  
      "ora_inizio": "string",  
      "ora_fine": "string",  
      "stato": "string",  
      "costo": "number",  
      "note": "string"  
    }  
  ]  
}
```

Cancella Appuntamento

- Cancella un appuntamento esistente.

- **Endpoint:** *PUT /api/mentee/appointments/:id/cancel*
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)
- **URL Parameters:**
- *id (number)* - ID dell'appuntamento

Response (200 OK):

```
{
  "success": true,
  "message": "Appuntamento cancellato con successo"
}
```

Storico Pagamenti

- Visualizza lo storico dei pagamenti effettuati.
- **Endpoint:** *GET /api/mentee/payments/history*
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Response (200 OK):

```
{
  "success": true,
  "payments": [
    {
      "id": "number",
      "data": "string",
      "importo": "number",
      "metodo_pagamento": "card | paypal",
      "stato": "completato | in_attesa | fallito",
      "appuntamento_id": "number",
      "mentor_nome": "string"
    }
  ]
}
```

```
}
```

```
],
```

```
"totale": "number"
```

```
}
```

Modifica Profilo

- Modifica i dati del profilo mentee.
- **Endpoint:** PUT /api/mentee/profile
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Request Body:

```
{
```

```
"nome": "string (opzionale)",
```

```
"cognome": "string (opzionale)",
```

```
"telefono": "string (opzionale)",
```

```
"obiettivi": "string (opzionale)"
```

```
}
```

Response (200 OK):

```
{
```

```
"success": true,
```

```
"message": "Profilo aggiornato con successo",
```

```
"mentee": {
```

```
"id": "number",
```

```
"nome": "string",
```

```
"cognome": "string",
```

```
"telefono": "string",
```

```
"obiettivi": "string"
```

```
}
```

```
}
```

Visualizza Recensioni Proprie

- Visualizza tutte le recensioni scritte dal mentee.
- **Endpoint:** GET /api/mentee/reviews
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Response (200 OK):

```
{
```

```
"success": true,
```

```
"reviews": [
```

```
{
```

```
"id": "number",
```

```
"mentor": {
```

```
"id": "number",
```

```
"nome": "string",
```

```
"cognome": "string"
```

```
},
```

```
"valutazione": "number (1-5)",
```

```
"commento": "string",
```

```
"data_creazione": "string"
```

```
}
```

```
]
```

```
}
```

Crea Recensione

- Crea una recensione per un mentor.
- **Endpoint:** POST /api/mentee/reviews
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Request Body:

```
{  
  "mentor_id": "number",  
  "appuntamento_id": "number",  
  "valutazione": "number (1-5)",  
  "commento": "string"  
}
```

Validazione:

- mentor_id richiesto
- appuntamento_id richiesto
- valutazione tra 1 e 5
- commento opzionale

Response (201 Created):

```
{  
  "success": true,  
  "message": "Recensione creata con successo",  
  "review": {  
    "id": "number",  
    "mentor_id": "number",  
    "valutazione": "number",  
    "commento": "string",
```

```
"data_creazione": "string"
```

```
}
```

```
}
```

Modifica Recensione

- Modifica una recensione esistente.
- **Endpoint:** PUT /api/mentee/reviews/:id
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)
- **URL Parameters:**
 - id (number) - ID della recensione

Request Body:

```
{
```

```
"valutazione": "number (1-5, opzionale)",
```

```
"commento": "string (opzionale)"
```

```
}
```

Response (200 OK):

```
{
```

```
"success": true,
```

```
"message": "Recensione aggiornata con successo",
```

```
"review": {
```

```
"id": "number",
```

```
"valutazione": "number",
```

```
"commento": "string"
```

```
}
```

```
}
```

Cancella Recensione

- Elimina una recensione.
- **Endpoint:** DELETE /api/mentee/reviews/:id
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)
- **URL Parameters:**
- id (number) - ID della recensione

Response (200 OK):

json

```
{  
  
  "success": true,  
  
  "message": "Recensione eliminata con successo"  
  
}
```

- Tutte le Notifiche
- Visualizza tutte le notifiche del mentee.
- **Endpoint:** GET /api/mentee/notifications
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)

Response (200 OK):

json

```
{  
  
  "success": true,  
  
  "notifications": [  
  
    {  
  
      "id": "number",  
  
      "tipo": "string",  
  
      "messaggio": "string",  
  
      "letta": "boolean",
```



```
    "data_creazione": "string"
  }
]
}
```

Singola Notifica

- Visualizza una notifica specifica.
- **Endpoint:** GET /api/mentee/notifications/:id
- **Autenticazione richiesta:** Sì (JWT + Ruolo Mentee)
- **URL Parameters:**
- id (number) - ID della notifica

Response (200 OK):

json

```
{
  "success": true,
  "notification": {
    "id": "number",
    "tipo": "string",
    "messaggio": "string",
    "letta": "boolean",
    "data_creazione": "string",
    "dettagli": "object"
  }
}
```

Mentor

- Base URL: /api/mentor
- 3.1 Opzioni per Registrazione (Pubblica)
- Ottiene le opzioni disponibili per la registrazione mentor.
- **Endpoint:** GET /api/mentor/options
- **Autenticazione richiesta:** No

Response (200 OK):

```
{
  "success": true,
  "options": {
    "specializzazioni": ["string"],
    "fasce_orarie": ["string"],
    "metodi_pagamento": ["string"]
  }
}
```

Info Personali

- Ottiene le informazioni personali del mentor.
- **Endpoint:** GET /api/mentor/personal/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID del mentor

Response (200 OK):

```
json
{
  "success": true,
  "mentor": {
    "id": "number",
```

```
"nome": "string",  
  
"cognome": "string",  
  
"email": "string",  
  
"telefono": "string",  
  
"biografia": "string",  
  
"specializzazione": "string",  
  
"anni_esperienza": "number",  
  
"tariffa_oraria": "number",  
  
"cv_path": "string"  
  
}  
  
}
```

Aggiorna Info Personali

- Modifica le informazioni personali del mentor.
- **Endpoint:** PUT /api/mentor/personal/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID del mentor

Request Body:

```
{  
  
"nome": "string (opzionale)",  
  
"cognome": "string (opzionale)",  
  
"telefono": "string (opzionale)",  
  
"biografia": "string (opzionale)",  
  
"specializzazione": "string (opzionale)",  
  
}
```

"anni_esperienza": "number (opzionale)",

"tariffa_oraria": "number (opzionale)"

}

Response (200 OK):

{

"success": true,

"message": "Informazioni aggiornate con successo",

"mentor": {

"id": "number",

"nome": "string",

"cognome": "string"

}

} I Miei Appuntamenti

- Visualizza tutti gli appuntamenti del mentor autenticato.
- **Endpoint:** `GET /api/mentor/appointments`
- **Autenticazione richiesta:** Sì (JWT)
- **Query Parameters:**
- ``
- ?stato=in_attesa|confermato|completato|cancellato
- &data_da=YYYY-MM-DD
- &data_a=YYYY-MM-DD

Response (200 OK):

json

{

"success": true,

"appointments": [

```
{  
  "id": "number",  
  "mentee": {  
    "id": "number",  
    "nome": "string",  
    "cognome": "string"  
  },  
  "data": "string",  
  "ora_inizio": "string",  
  "ora_fine": "string",  
  "stato": "string",  
  "costo": "number",  
  "note": "string"  
}  
]  
}
```

3.5 Appuntamenti per Mentor

- Ottiene gli appuntamenti di un mentor specifico.
- **Endpoint:** GET /api/mentor/appointments/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID del mentor

Response (200 OK):

```
{
```

```
"success": true,  
  
"appointments": [  
  
  {  
  
    "id": "number",  
  
    "mentee_id": "number",  
  
    "data": "string",  
  
    "ora_inizio": "string",  
  
    "ora_fine": "string",  
  
    "stato": "string"  
  
  }  
  
]  
  
}
```

3.6 Dettaglio Appuntamento

- Visualizza un appuntamento specifico.
- **Endpoint:** GET /api/mentor/appointment/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
 - id (number) - ID dell'appuntamento

Response (200 OK):

```
{  
  
  "success": true,  
  
  "appointment": {  
  
    "id": "number",  
  
    "mentee": {
```

```
"id": "number",

"nome": "string",

"cognome": "string",

"email": "string"

},

"data": "string",

"ora_inizio": "string",

"ora_fine": "string",

"stato": "string",

"costo": "number",

"note": "string",

"data_creazione": "string"

}

}
```

- Rispondi ad Appuntamento
- Conferma o rifiuta una richiesta di appuntamento.
- **Endpoint:** PUT /api/mentor/appointment/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
 - id (number) - ID dell'appuntamento

Request Body:

```
{

"azione": "conferma | rifiuta",

"motivo": "string (opzionale, richiesto se rifiuta)"

}
```

Response (200 OK):

```
{  
  
  "success": true,  
  
  "message": "Appuntamento confermato/rifiutato con successo",  
  
  "appointment": {  
  
    "id": "number",  
  
    "stato": "confermato | rifiutato"  
  
  }  
  
}
```

3.8 Aggiorna Stato Appuntamento

- Modifica lo stato di un appuntamento.
- **Endpoint:** PUT /api/mentor/appointments/:id/status
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID dell'appuntamento

Request Body:

```
json  
  
{  
  
  "stato": "completato | cancellato"  
  
}
```

Response (200 OK):

```
json  
  
{  
  
  "success": true,
```


"message": "Stato aggiornato con successo"

}

3.9 Cancella Appuntamento

- Elimina un appuntamento.
- **Endpoint:** DELETE /api/mentor/appointment/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID dell'appuntamento

Response (200 OK):

json

{

"success": true,

"message": "Appuntamento eliminato con successo"

}

3.10 Tutte le Notifiche Mentor

- Visualizza tutte le notifiche del mentor.
- **Endpoint:** GET /api/mentor/notifications
- **Autenticazione richiesta:** Sì (JWT)

Response (200 OK):

json

{

"success": true,

"notifications": [

{

"id": "number",

```
"tipo": "string",  
  
"messaggio": "string",  
  
"letta": "boolean",  
  
"data_creazione": "string"  
  
}  
  
]  
  
}
```

3.11 Singola Notifica Mentor

- Visualizza una notifica specifica del mentor.
- **Endpoint:** GET /api/mentor/notification/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID della notifica

Response (200 OK):

```
"success": true,  
  
"notification": {  
  
  "id": "number",  
  
  "tipo": "string",  
  
  "messaggio": "string",  
  
  "letta": "boolean",  
  
  "data_creazione": "string",  
  
  "dettagli": "object"  
  
}  
  
}
```

3.12 Visualizza Disponibilità

- Ottiene le fasce di disponibilità del mentor.
- **Endpoint:** GET /api/mentor/availability/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID del mentor

Response (200 OK):

json

```
{  
  
  "success": true,  
  
  "availability": [  
  
    {  
  
      "id": "number",  
  
      "giorno_settimana": "Lunedì | Martedì | ... | Domenica",  
  
      "ora_inizio": "HH:MM",  
  
      "ora_fine": "HH:MM",  
  
      "attivo": "boolean"  
    }  
  ]  
}
```

3.13 Aggiungi Disponibilità

- Crea una nuova fascia di disponibilità.
- **Endpoint:** POST /api/mentor/availability/:id
- **Autenticazione richiesta:** Sì (JWT)
- **URL Parameters:**
- id (number) - ID del mentor

- **Request Body:**

json

```
{  
  
  "giorno_settimana": "Lunedì | Martedì | ... | Domenica",  
  
  "ora_inizio": "HH:MM",  
  
  "ora_fine": "HH:MM"  
  
}
```

Response (201 Created):

json

```
{  
  
  "success": true,  
  
  "message": "Disponibilità aggiunta con successo",  
  
  "availability": {  
  
    "id": "number",  
  
    "giorno_settimana": "string",  
  
    "ora_inizio": "string",  
  
    "ora_fine": "string"  
  
  }  
  
}
```

3.14 Aggiorna Disponibilità

Modifica una fascia di disponibilità esistente.

Endpoint: PUT /api/mentor/availability/:id

Autenticazione richiesta: Sì (JWT)

URL Parameters:

- *id (number)* - ID della disponibilità

Request Body:

json

```
{  
  
  "giorno_settimana": "string (opzionale)",  
  
  "ora_inizio": "HH:MM (opzionale)",  
  
  "ora_fine": "HH:MM (opzionale)",  
  
  "attivo": "boolean (opzionale)"  
}
```

Response (200 OK):

json

```
{  
  
  "success": true,  
  
  "message": "Disponibilità aggiornata con successo"  
}
```

3.15 Elimina Disponibilità

Rimuove una fascia di disponibilità.

Endpoint: `DELETE /api/mentor/availability/:id`

Autenticazione richiesta: Sì (JWT)

URL Parameters:

- *id (number)* - ID della disponibilità

Response (200 OK):

```
json
{
  "success": true,
  "message": "Disponibilità eliminata con successo"
}
```

3.16 Tutte le Recensioni

Visualizza tutte le recensioni ricevute dal mentor.

Endpoint: GET /api/mentor/reviews/:id

Autenticazione richiesta: Sì (JWT)

URL Parameters:

- id (number) - ID del mentor

Response (200 OK):

```
json
{
  "success": true,
  "reviews": [
    {
      "id": "number",
      "mentee": {
        "id": "number",
        "nome": "string",
        "cognome": "string"
      },
    },
  ],
}
```

```
"valutazione": "number",  
  
"commento": "string",  
  
"data_creazione": "string"  
  
}  
  
]  
  
}
```

3.17 Statistiche Recensioni

Ottiene le statistiche aggregate delle recensioni.

Endpoint: GET /api/mentor/reviews-stats/:id

Autenticazione richiesta: Sì (JWT)

URL Parameters:

- id (number) - ID del mentor

Response (200 OK):

```
json  
  
{  
  
  "success": true,  
  
  "stats": {  
  
    "rating_medio": "number",  
  
    "totale_recensioni": "number",  
  
    "distribuzione": {  
  
      "5_stelle": "number",  
  
      "4_stelle": "number",  
  
      "3_stelle": "number",
```

```
"2_stelle": "number",  
  
"1_stella": "number"  
  
}  
  
}  
  
}
```

4. Messaggi

Base URL: /api/messages

4.1 Invia Messaggio

Invia un messaggio a un altro utente.

Endpoint: POST /api/messages/send

Autenticazione richiesta: Sì (JWT)

Request Body:

```
json  
  
{  
  
  "destinatario_id": "number",  
  
  "testo": "string"  
  
}
```

Response (201 Created):

```
json  
  
{  
  
  "success": true,  
  
  "message": {  
  
    "id": "number",
```



```
"mittente_id": "number",  
  
"destinatario_id": "number",  
  
"testo": "string",  
  
"letto": "boolean",  
  
"data_invio": "string"  
  
}  
  
}
```

4.2 Conversazione

Visualizza tutti i messaggi tra due utenti.

Endpoint: GET /api/messages/conversation/:userId1/:userId2

Autenticazione richiesta: Sì (JWT)

URL Parameters:

- userId1 (number) - ID del primo utente
- userId2 (number) - ID del secondo utente

Response (200 OK):

```
json  
  
{  
  
  "success": true,  
  
  "messages": [  
  
    {  
  
      "id": "number",  
  
      "mittente_id": "number",  
  
      "destinatario_id": "number",  
  
      "testo": "string",
```

```
"letto": "boolean",  
  
"data_invio": "string"  
  
}  
  
]  
  
}
```

4.3 Tutte le Chat

Visualizza tutte le conversazioni di un utente.

Endpoint: GET /api/messages/chats/:userId

Autenticazione richiesta: Sì (JWT)

URL Parameters:

- *userId (number)* - ID dell'utente

Response (200 OK):

json

```
{  
  
  "success": true,  
  
  "chats": [  
  
    {  
  
      "utente": {  
  
        "id": "number",  
  
        "nome": "string",  
  
        "cognome": "string",  
  
        "ruolo": "string"  
  
      },  
  
    },  
  
  ],  
  
}
```

```
"ultimo_messaggio": {  
  "testo": "string",  
  "data_invio": "string",  
  "letto": "boolean"  
},  
"messaggi_non_letti": "number"  
}  
]  
}
```

5. Pagamenti

Base URL: /api/payments

5.1 Pagamento con Carta

Effettua un pagamento con carta di credito.

Endpoint: POST /api/payments/card

Autenticazione richiesta: Sì (JWT)

Request Body:

```
json  
  
{  
  "appuntamento_id": "number",  
  "importo": "number",  
  "carta": {  
    "numero": "string",  
    "scadenza": "MM/YY",
```

```
"cvv": "string",  
  
"intestatario": "string"  
  
}  
  
}
```

Response (200 OK):

```
json  
  
{  
  
  "success": true,  
  
  "message": "Pagamento completato con successo",  
  
  "payment": {  
  
    "id": "number",  
  
    "appuntamento_id": "number",  
  
    "importo": "number",  
  
    "metodo": "card",  
  
    "stato": "completato",  
  
    "transaction_id": "string",  
  
    "data_pagamento": "string"  
  
  }  
  
}
```

5.2 Pagamento con PayPal

Inizia un pagamento con PayPal.

Endpoint: POST /api/payments/paypal

Autenticazione richiesta: Sì (JWT)

Request Body:

```
json

{

  "appuntamento_id": "number",

  "importo": "number"

}
```

Response (200 OK):

```
json

{

  "success": true,

  "approval_url": "string",

  "order_id": "string"

}
```

5.3 Conferma Pagamento PayPal

Conferma un pagamento PayPal dopo l'approvazione.

Endpoint: POST /api/payments/paypal/confirm

Autenticazione richiesta: Sì (JWT)

Request Body:

```
json

{

  "order_id": "string",

  "app
```

DEVI METTERE LA SOCUMANTAZIONE IN UN FILE PDF SISTEMATO DA MANDARE AL PROF

22:08

```
untamento_id": "number" }
```

****Response (200 OK):****

```
```json
```

```
{

 "success": true,

 "message": "Pagamento PayPal completato con successo",

 "payment": {

 "id": "number",

 "appuntamento_id": "number",

 "importo": "number",

 "metodo": "paypal",

 "stato": "completato",

 "transaction_id": "string",

 "data_pagamento": "string"

 }

}

...

—
```

### 5.4 Storico Pagamenti Mentee

*Visualizza lo storico dei pagamenti di un mentee.*

**\*\*Endpoint:\*\*** `GET /api/payments/history/mentee/:id``

**\*\*Autenticazione richiesta:\*\*** Sì (JWT)

**\*\*URL Parameters:\*\***

- ``id` (number)` - ID del mentee

**\*\*Response (200 OK):\*\***

```json`

`{`

`"success": true,`

`"payments": [`

`{`

`"id": "number",`

`"appuntamento_id": "number",`

`"mentor_nome": "string",`

`"importo": "number",`

`"metodo": "card | paypal",`

`"stato": "completato | in_attesa | fallito",`

`"data_pagamento": "string"`

`}`

```
],  
  
  "totale_speso": "number"  
  
}  
  
...  
  
—
```

5.5 Guadagni Totali Mentor

- *Visualizza i guadagni totali di un mentor.*
- ****Endpoint:**** `GET /api/payments/earnings/mentor/:id`
- ****Autenticazione richiesta:**** Sì (JWT)
- ****URL Parameters:****
- - `id` (number) - ID del mentor

****Response (200 OK):****

```
{  
  
  "success": true,  
  
  "earnings": {  
  
    "totale": "number",  
  
    "questo_mese": "number",  
  
    "ultimo_mese": "number",  
  
    "appuntamento_completati": "number",  
  
    "dettagli": [  
  
      {
```



```
"mese": "string",  
  
"importo": "number",  
  
"numero_appuntamenti": "number"  
  
}  
  
]  
  
}  
  
}  
  
...  
  
—
```

6. Utenti

Base URL: `/api/users`

6.1 Tutti gli Utenti

Ottiene la lista di tutti gli utenti (solo admin).

Endpoint: `GET /api/users`

Autenticazione richiesta: Sì (JWT - Admin)

Response (200 OK):

```
```json
```

```
{
```

```
 "success": true,
```

```
 "users": [
```

```
 {
```

```
 "id": "number",
```

```
 "nome": "string",
```

```
 "cognome": "string",
```

```
 "email": "string",
```

```
 "ruolo": "Mentor | Mentee | Admin",
```

```
 "data_registrazione": "string"
```

```
 }
```

```
]
```

```
}
```

```
```
```

—

6.2 Utente per ID

Ottiene i dettagli di un utente specifico.

****Endpoint:**** `GET /api/users/:id`

****Autenticazione richiesta:**** Sì (JWT - Admin)

****URL Parameters:****

- `id` (number) - ID dell'utente

****Response (200 OK):****

```json

{

  "success": true,

  "user": {

    "id": "number",

    "nome": "string",

    "cognome": "string",

    "email": "string",

    "telefono": "string",

    "ruolo": "string",

    "data\_registrazione": "string"

  }

}

```

—

6.3 Crea Utente

Crea un nuovo utente (solo admin).

Endpoint: `POST /api/users``

Autenticazione richiesta: Sì (JWT - Admin)

Request Body:

```json`

`{`

`"nome": "string",`

`"cognome": "string",`

`"email": "string",`

`"password": "string",`

`"ruolo": "Mentor | Mentee | Admin",`

`"telefono": "string (opzionale)"`

`}`

````

Response (201 Created):

```json`

`{`

`"success": true,`

*"message": "Utente creato con successo",*

*"user": {*

*"id": "number",*

*"email": "string",*

*"ruolo": "string"*

*}*

*}*

*...*

*—*

### *### 6.4 Aggiorna Utente*

*Modifica i dati di un utente esistente.*

*\*\*Endpoint:\*\* `PUT /api/users/:id`*

*\*\*Autenticazione richiesta:\*\* Sì (JWT - Admin)*

*\*\*URL Parameters:\*\**

*- `id` (number) - ID dell'utente*

*\*\*Request Body:\*\**

*``json*

```
{
 "nome": "string (opzionale)",
 "cognome": "string (opzionale)",
 "email": "string (opzionale)",
 "telefono": "string (opzionale)",
 "ruolo": "string (opzionale)"
}
```

...

**\*\*Response (200 OK):\*\***

**``json**

```
{
 "success": true,
 "message": "Utente aggiornato con successo",
 "user": {
 "id": "number",
 "nome": "string",
 "cognome": "string"
 }
}
```

...

### ### 6.5 Elimina Utente

*Elimina un utente dal sistema.*

**\*\*Endpoint:\*\*** `DELETE /api/users/:id``

**\*\*Autenticazione richiesta:\*\*** Sì (JWT - Admin)

**\*\*URL Parameters:\*\***

- ``id`` (number) - ID dell'utente

**\*\*Response (200 OK):\*\***

```json`

```
{  
  
  "success": true,  
  
  "message": "Utente eliminato con successo"  
}
```

````

—

## Codici di Stato HTTP

| *Codice* | *Significato* |

|----|-----|

| 200 | *OK - Richiesta completata con successo* |

| 201 | *Created - Risorsa creata con successo* |

| 400 | *Bad Request - Dati della richiesta non validi* |

| 401 | *Unauthorized - Autenticazione richiesta o fallita* |

| 403 | *Forbidden - Permessi insufficienti* |

| 404 | *Not Found - Risorsa non trovata* |

| 409 | *Conflict - Conflitto con lo stato attuale (es. email già esistente)* |

| 500 | *Internal Server Error - Errore del server* |

—

## *## Autenticazione*

*Tutte le route protette richiedono un JWT token nell'header Authorization:*

*Authorization: Bearer <your\_jwt\_token>*

*Il token viene fornito dopo un login riuscito e deve essere incluso in tutte le richieste autenticate.*