

## # Documentazione delle API del Progetto Feltrinelli Gaming

Ecco la documentazione completa delle API disponibili nel progetto Feltrinelli Gaming, con dettagli su come utilizzarle:

### ## Indice

1. Health Check
2. Autenticazione
3. Gestione Sessioni
4. Quiz Libri
5. Quiz Autori
6. Quiz Anni
7. Classifiche
8. Premi

### ## Health Check

#### ### GET /api/health-check

Verifica che il server sia in esecuzione.

Risposta :

```
```json
{
  "status": "ok",
  "message": "Server is running"
}
```
```

#### ### GET /api/health

Endpoint di health check più dettagliato.

Risposta :

```
```json
{
  "status": "ok",
  "message": "Gaming Engine API is running"
}
```
```

### ## Autenticazione

#### ### POST /api/auth/login

Effettua il login di un utente.

Richiesta :

```
```json
{
  "email": "utente@esempio.com",
  "password": "password123"
}
```
```

Risposta :

```
```json
{
  "user": {
    "id": "user-uuid",
    "username": "nomeutente",
    "email": "utente@esempio.com"
  },
}
```

```
"token": "jwt-token"
}
...
```

## Gestione Sessioni  
### POST /api/games/session  
Crea una nuova sessione di gioco.

Richiesta :

```
```json
{
  "user_id": "user-uuid",
  "game_id": "00000000-0000-0000-0000-000000000001"
}
...
```

Risposta :

```
```json
{
  "session_id": "session-uuid",
  "user_id": "user-uuid",
  "game_id": "00000000-0000-0000-0000-000000000001",
  "created_at": "2023-01-01T12:00:00Z",
  "score": 0,
  "completed": false
}
...
```

## Quiz Libri  
### GET /api/games/bookquiz/question  
Ottiene una domanda casuale per il quiz sui libri.

Parametri Query :

- difficulty (opzionale): Livello di difficoltà (default: 1)

Risposta :

```
```json
{
  "question_id": "question-uuid",
  "question_text": "Indovina il libro da questo estratto:",
  "abstract_snippet": "Testo dell'estratto...",
  "options": [
    {
      "id": "option-uuid-1",
      "title": "Titolo Libro 1",
      "author": "Autore 1",
      "image_url": "url-immagine-1"
    },
    {
      "id": "option-uuid-2",
      "title": "Titolo Libro 2",
      "author": "Autore 2",
      "image_url": "url-immagine-2"
    },
    {
      "id": "option-uuid-3",
      "title": "Titolo Libro 3",

```

```

    "author": "Autore 3",
    "image_url": "url-immagine-3"
  },
  {
    "id": "option-uuid-4",
    "title": "Titolo Libro 4",
    "author": "Autore 4",
    "image_url": "url-immagine-4"
  }
]
}
...

```

### POST /api/games/bookquiz/answer  
 Invia una risposta per una domanda del quiz sui libri.

Richiesta :

```

```json
{
  "session_id": "session-uuid",
  "question_id": "question-uuid",
  "answer_option_id": "option-uuid-2",
  "time_taken": 5.2
}
...

```

Risposta :

```

```json
{
  "is_correct": true,
  "correct_book": {
    "id": "book-uuid",
    "title": "Titolo Libro Corretto",
    "authors": "Autore Corretto",
    "abstract": "Abstract completo del libro...",
    "image_url": "url-immagine"
  },
  "message": "Risposta corretta!",
  "points": 1
}
...

```

## Quiz Autori  
 ### POST /api/games/authorquiz/session/init  
 Inizializza una nuova sessione per il quiz sugli autori.

Richiesta :

```

```json
{
  "user_id": "user-uuid"
}
...

```

Risposta :

```

```json
{

```

```

"session_id": "session-uuid",
"user_id": "user-uuid",
"game_id": "00000000-0000-0000-0000-000000000002",
"created_at": "2023-01-01T12:00:00Z",
"score": 0,
"completed": false
}
...

```

### GET /api/games/authorquiz/question  
 Ottiene una domanda casuale per il quiz sugli autori.

Parametri Query :

- difficulty (opzionale): Livello di difficoltà (default: 1)  
 Risposta :

```

```json
{
  "question_id": "question-uuid",
  "question_text": "Chi è l'autore di questi libri?",
  "books": [
    {
      "id": "book-uuid-1",
      "title": "Titolo Libro 1",
      "image_url": "url-immagine-1"
    },
    {
      "id": "book-uuid-2",
      "title": "Titolo Libro 2",
      "image_url": "url-immagine-2"
    }
  ],
  "options": [
    {
      "id": "author-uuid-1",
      "name": "Nome Autore 1"
    },
    {
      "id": "author-uuid-2",
      "name": "Nome Autore 2"
    },
    {
      "id": "author-uuid-3",
      "name": "Nome Autore 3"
    },
    {
      "id": "author-uuid-4",
      "name": "Nome Autore 4"
    }
  ],
  "correct_author": {
    "id": "author-uuid-2",
    "name": "Nome Autore 2"
  }
}
...

```

### POST /api/games/authorquiz/answer  
 Invia una risposta per una domanda del quiz sugli autori.

Richiesta :

```
```json
{
  "session_id": "session-uuid",
  "question_id": "question-uuid",
  "answer_option_id": "author-uuid-2",
  "time_taken": 4.8
}
```
```

Risposta :

```
```json
{
  "is_correct": true,
  "correct_author": {
    "id": "author-uuid-2",
    "name": "Nome Autore 2"
  },
  "message": "Risposta corretta!",
  "points": 1
}
```
```

## Quiz Anni

### POST /api/games/yearquiz/session/init

Inizializza una nuova sessione per il quiz sugli anni di pubblicazione.

Richiesta :

```
```json
{
  "user_id": "user-uuid"
}
```
```

Risposta :

```
```json
{
  "session_id": "session-uuid",
  "user_id": "user-uuid",
  "game_id": "00000000-0000-0000-0000-000000000003",
  "created_at": "2023-01-01T12:00:00Z",
  "score": 0,
  "completed": false
}
```
```

### GET /api/games/yearquiz/question

Ottiene una domanda casuale per il quiz sugli anni di pubblicazione.

Parametri Query :

- difficulty (opzionale): Livello di difficoltà (default: 1)

Risposta :

```
```json
```

```
{
  "question_id": "question-uuid",
  "question_text": "In quale anno è stato pubblicato questo libro?",
  "book": {
    "id": "book-uuid",
    "title": "Titolo Libro",
    "author": "Nome Autore",
    "image_url": "url-immagine"
  },
  "options": [1985, 1990, 1995, 2000],
  "correct_year": 1990
}
...
...
```

### POST /api/games/yearquiz/answer  
 Invia una risposta per una domanda del quiz sugli anni.

Richiesta :

```
```json
{
  "session_id": "session-uuid",
  "question_id": "question-uuid",
  "answer_year": 1990,
  "time_taken": 3.5
}
...
```
```

Risposta :

```
```json
{
  "is_correct": true,
  "correct_year": 1990,
  "message": "Risposta corretta!",
  "points": 1
}
...
```
```

## Classifiche  
 ### GET /api/leaderboard  
 Ottiene la classifica globale per tutti i giochi.

Parametri Query :

- period (opzionale): Periodo della classifica (all\_time, monthly, weekly)
- limit (opzionale): Numero massimo di risultati (default: 10)

Risposta :

```
```json
{
  "data": [
    {
      "id": "leaderboard-entry-uuid-1",
      "user_id": "user-uuid-1",
      "points": 150,
      "users": {
        "username": "nomeutente1",
        "avatar_url": "url-avatar-1"
      }
    }
  ]
}
...
```
```

```

    }
  },
  {
    "id": "leaderboard-entry-uuid-2",
    "user_id": "user-uuid-2",
    "points": 120,
    "users": {
      "username": "nomeutente2",
      "avatar_url": "url-avatar-2"
    }
  }
]
}

```

### GET /api/leaderboard/:gameId  
 Ottiene la classifica per un gioco specifico.

Parametri Path :

- gameId : ID del gioco (es. 00000000-0000-0000-0000-000000000001)

Parametri Query :

- period (opzionale): Periodo della classifica (all\_time, monthly, weekly)  
 - limit (opzionale): Numero massimo di risultati (default: 10)

Risposta :

```

```json
{
  "data": [
    {
      "id": "leaderboard-entry-uuid-1",
      "user_id": "user-uuid-1",
      "game_id": "00000000-0000-0000-0000-000000000001",
      "points": 80,
      "users": {
        "username": "nomeutente1",
        "avatar_url": "url-avatar-1"
      }
    },
    {
      "id": "leaderboard-entry-uuid-2",
      "user_id": "user-uuid-2",
      "game_id": "00000000-0000-0000-0000-000000000001",
      "points": 65,
      "users": {
        "username": "nomeutente2",
        "avatar_url": "url-avatar-2"
      }
    }
  ]
}

```

### POST /api/leaderboard/submit  
 Invia un punteggio alla classifica.

Richiesta :

```

```json

```

```
{
  "userId": "user-uuid",
  "gameId": "00000000-0000-0000-0000-000000000001",
  "correctAnswers": 4,
  "totalQuestions": 5,
  "sessionId": "session-uuid"
}
```

Risposta :

```
```json
{
  "success": true,
  "message": "Punteggio aggiornato con successo"
}
```

### POST /api/leaderboard/submit-all-periods

Invia un punteggio alla classifica per tutti i periodi (all\_time, monthly, weekly).

Richiesta :

```
```json
{
  "userId": "user-uuid",
  "gameId": "00000000-0000-0000-0000-000000000001",
  "correctAnswers": 4,
  "totalQuestions": 5,
  "sessionId": "session-uuid"
}
```

Risposta :

```
```json
{
  "success": true,
  "message": "Punteggio aggiornato per tutti i periodi"
}
```

## Premi

### GET /api/rewards/available

Ottiene i premi disponibili per un gioco e un periodo specifici.

Parametri Query :

- game\_id : ID del gioco
- period : Periodo (all\_time, monthly, weekly)

Risposta :

```
```json
{
  "success": true,
  "rewards": [
    {
      "id": "reward-uuid-1",
      "name": "Tazza Feltrinelli",
      "description": "Una tazza esclusiva Feltrinelli",
    }
  ]
}
```



```

    "image_url": "url-immagine-tazza",
    "points_required": 100,
    "rank": 1
  },
  {
    "id": "reward-uuid-2",
    "name": "Buono Sconto 10%",
    "description": "Buono sconto del 10% su tutti i libri",
    "image_url": "url-immagine-buono",
    "points_required": 50,
    "rank": 2
  },
  {
    "id": "reward-uuid-3",
    "name": "Segnalibro Feltrinelli",
    "description": "Segnalibro esclusivo Feltrinelli",
    "image_url": "url-immagine-segnalibro",
    "points_required": 25,
    "rank": 3
  }
]
}

```

### GET /api/rewards/user/:userId  
 Ottiene i premi ottenuti da un utente specifico.

Parametri Path :

- userId : ID dell'utente

Risposta :

```

```json
{
  "success": true,
  "rewards": [
    {
      "id": "user-reward-uuid-1",
      "user_id": "user-uuid",
      "reward_id": "reward-uuid-1",
      "game_id": "00000000-0000-0000-0000-000000000001",
      "period": "all_time",
      "rank": 1,
      "claimed_at": "2023-01-01T12:00:00Z",
      "rewards": {
        "name": "Tazza Feltrinelli",
        "description": "Una tazza esclusiva Feltrinelli",
        "image_url": "url-immagine-tazza"
      }
    }
  ]
}

```

## Come utilizzare le API

### Esempio di flusso di gioco per Quiz Libri:

1. Inizializzare una sessione :

```

```javascript
const response = await fetch('http://localhost:3000/api/games/session', {

```

```

method: 'POST',
headers: { 'Content-Type': 'application/json' },
body: JSON.stringify({
  user_id: 'user-uuid',
  game_id: '00000000-0000-0000-0000-000000000001'
})
});
const session = await response.json();
const sessionId = session.session_id;
'''
'''

```

2. Ottenere una domanda :

```

'''javascript
const questionResponse = await fetch('http://localhost:3000/api/games/bookquiz/question?
difficulty=1');
const question = await questionResponse.json();
'''
'''

```

3. Inviare una risposta :

```

'''javascript
const answerResponse = await fetch('http://localhost:3000/api/games/bookquiz/answer', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    session_id: sessionId,
    question_id: question.question_id,
    answer_option_id: question.options[0].id,
    time_taken: 3.5
  })
});
const result = await answerResponse.json();
'''
'''

```

4. Inviare il punteggio finale alla classifica :

```

'''javascript
const leaderboardResponse = await fetch('http://localhost:3000/api/leaderboard/submit-all-
periods', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    userId: 'user-uuid',
    gameId: '00000000-0000-0000-0000-000000000001',
    correctAnswers: 4,
    totalQuestions: 5,
    sessionId: sessionId
  })
});
const leaderboardResult = await leaderboardResponse.json();
'''
'''

```

### Note importanti:

- Tutti gli ID dei giochi sono UUID predefiniti:
- Quiz Libri: 00000000-0000-0000-0000-000000000001
- Quiz Autori: 00000000-0000-0000-0000-000000000002
- Quiz Anni: 00000000-0000-0000-0000-000000000003
- I punteggi vengono calcolati automaticamente dal backend in base al numero di risposte corrette

- Le classifiche sono disponibili per diversi periodi: generale (all\_time), mensile (monthly) e settimanale (weekly)
- I premi sono associati alle posizioni in classifica (1°, 2°, 3° posto)