

# Documentazione API - Sistema Ticketing

## Introduzione

Questa documentazione descrive le API del sistema di gestione ticket, incluse le chiamate per la registrazione, il login, la creazione e gestione dei ticket. Le API usano il metodo POST e accettano dati in formato JSON.

## Base URL

`http://localhost/ecoctrl-back/api.php`

## 1. Registrazione Utente

Registra un nuovo utente nel sistema.

Metodo: POST

Endpoint: `/api.php`

Headers:

`Content-Type: application/json`

Body JSON:

```
{
  "action": "register",
  "email": "test@example.com",
  "password": "password123"
}
```

Risposta attesa:

```
{
  "success": true,
  "message": "Registrazione completata!"
}
```

Possibili errori:

```
{
  "success": false,
```

# Documentazione API - Sistema Ticketing

```
{
  "message": "Email già registrata"
}
{
  "success": false,
  "message": "Compila tutti i campi"
}
```

## 2. Login Utente

Accede con un utente registrato e restituisce i suoi dati.

Metodo: POST

Endpoint: /api.php

Headers:

Content-Type: application/json

Body JSON:

```
{
  "action": "login",
  "email": "test@example.com",
  "password": "password123"
}
```

Risposta attesa:

```
{
  "success": true,
  "user": {
    "id": 1,
    "email": "test@example.com",
    "role": "Client"
  }
}
```

Possibili errori:

# Documentazione API - Sistema Ticketing

```
{
  "success": false,
  "message": "Utente non trovato"
}
{
  "success": false,
  "message": "Password errata"
}
{
  "success": false,
  "message": "Compila tutti i campi"
}
```

## 3. Ottenere i Ticket dell'Utente

Recupera tutti i ticket associati a un utente specifico.

Metodo: POST

Endpoint: /api.php

Headers:

Content-Type: application/json

Body JSON:

```
{
  "action": "getTickets",
  "user_id": 1
}
```

Risposta attesa:

```
{
  "success": true,
  "tickets": [
    {
      "id": 13,
      "ticketDate": "2025-03-13 15:38:40",

```

# Documentazione API - Sistema Ticketing

```
{
  "description": "Problema con il sistema",
  "client_id": 1,
  "ticketCat_id": 1,
  "status": "Aperto",
  "admin_id": 1,
  "admin_name": "Michele"
}
```

Possibili errori:

```
{
  "success": false,
  "message": "Utente non autenticato"
}
```

## 4. Creazione Nuovo Ticket

Crea un nuovo ticket per l'utente loggato.

Metodo: POST

Endpoint: /api.php

Headers:

Content-Type: application/json

Body JSON:

```
{
  "action": "createTicket",
  "user_id": 1,
  "ticketCat_id": 1,
  "description": "Problema con il sistema"
}
```

Risposta attesa:

# Documentazione API - Sistema Ticketing

```
{  
  "success": true,  
  "message": "Ticket creato con successo!"  
}
```

Possibili errori:

```
{  
  "success": false,  
  "message": "Dati mancanti"  
}  
  
{  
  "success": false,  
  "message": "Categoria ticket non valida"  
}
```

## 5. Assegnare un Admin a un Ticket

Permette a un admin di prendere in carico un ticket.

Metodo: POST

Endpoint: /api.php

Headers:

```
Content-Type: application/json
```

Body JSON:

```
{  
  "action": "assignAdminToTicket",  
  "ticket_id": 13,  
  "admin_id": 1  
}
```

Risposta attesa:

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

# Documentazione API - Sistema Ticketing

```
{
  "message": "Admin assegnato con successo al ticket"
}
```

Possibili errori:

```
{
  "success": false,
  "message": "Dati mancanti"
}
{
  "success": false,
  "message": "L'utente selezionato non è un admin"
}
```

## Considerazioni Finali

Le API del sistema di ticketing permettono di gestire utenti, ticket e assegnazione degli admin.

Se si riceve "Azione non valida", verificare che 'action' sia inviato nel body JSON.

Tutte le richieste devono essere di tipo POST e avere il Content-Type: application/json.