

# **Attraction Backend System – Documentazione Tecnica Completa**

1.	Introduzione .....	2
2.	Panoramica del Sistema .....	2
2.1.	Gestione Utenti .....	2
2.2.	Pianificazione Viaggi e Tracciamento Attività .....	3
2.3.	Gestione Dati Trasporto .....	3
2.4.	Funzionalità Amministrative .....	3
3.	Funzionalità Principali .....	3
3.1.	Gestione Utenti .....	3
3.2.	Pianificazione Viaggi e Ricerca .....	4
3.3.	Fermate e Orari Stazioni .....	4
3.4.	Prenotazioni e Risparmio CO <sub>2</sub> .....	4
3.5.	Gestione Feedback .....	5
4.	Funzionalità Amministrative .....	5
5.	Sicurezza e Gestione Errori .....	5
5.1	Autenticazione .....	5
5.2	Gestione Errori .....	6
6.	API Endpoints .....	9
7.	Conclusione .....	10

## 1. Introduzione

Il sistema ATTRACTION BACKEND è il componente backend centrale di una moderna piattaforma Mobility-as-a-Service (MaaS) dedicata al miglioramento dei servizi di trasporto urbano attraverso l'innovazione digitale. Sviluppato utilizzando il framework Django, questo backend costituisce il livello principale che consente una pianificazione dei viaggi fluida, mobilità multimodale, gestione degli account utente, prenotazioni e calcolo del risparmio di CO<sub>2</sub>. Fornisce API REST sicure, scalabili e standardizzate per applicazioni web e mobile.

La piattaforma supporta sia utenti autenticati che anonimi, permettendo a chiunque di pianificare viaggi ed effettuare ricerche senza registrazione obbligatoria. Gli utenti registrati beneficiano di funzionalità avanzate, tra cui gestione dell'account, persistenza della cronologia delle ricerche, funzionalità di prenotazione e raccomandazioni personalizzate. Il sistema utilizza l'autenticazione basata su JWT per una gestione sicura delle sessioni e integra flussi di verifica via email e recupero password per mantenere l'integrità e la privacy degli account.

Una caratteristica fondamentale del backend ATTRACTION è la sua integrazione con il servizio OpenTripPlanner (OTP) tramite un'interfaccia GraphQL. Questa integrazione abilita la pianificazione dei trasporti in tempo reale attraverso molteplici modalità di viaggio, come camminata, autobus, ciclismo e soluzioni di micromobilità. Il backend elabora le richieste di viaggio degli utenti, comunica con i server OTP, struttura gli itinerari basati sulle modalità di viaggio e restituisce risposte comprensibili.

Inoltre, il backend include strumenti amministrativi avanzati accessibili tramite l'interfaccia Django Admin, consentendo la gestione di utenti, feedback, prenotazioni, attività di ricerca e luoghi preferiti. Il sistema aderisce a elevati standard di gestione degli errori, sicurezza e affidabilità, garantendo stabilità operativa e un ambiente di sviluppo robusto.

## 2. Panoramica del Sistema

Il backend ATTRACTION è composto da diversi sottosistemi chiave che lavorano insieme per formare una soluzione MaaS completa. Essi includono:

### 2.1. Gestione Utenti

La gestione utenti è implementata nel modulo **accounts/** e supporta registrazione, autenticazione, aggiornamento del profilo, operazioni sulla password e gestione della sessione. Questo modulo gestisce l'identità dell'utente e garantisce l'accesso sicuro alle risorse del sistema.

## 2.2. Pianificazione Viaggi e Tracciamento Attività

La pianificazione dei viaggi è implementata nel modulo **activity/**, che elabora le richieste dei viaggiatori, comunica con OTP, recupera dati di trasporto in tempo reale e registra le attività per analisi e cronologia.

## 2.3. Gestione Dati Trasporto

Il backend recupera fermate, orari e tempi di arrivo per le stazioni di transito direttamente dal server OTP, permettendo alla piattaforma di mostrare informazioni aggiornate in tempo reale.

## 2.4. Funzionalità Amministrative

La dashboard Django Admin include strutture tabelle avanzate, campi condizionali, ricerca e filtri e restrizioni di sola lettura per fornire un ambiente controllato agli amministratori di sistema.

# 3. Funzionalità Principali

## 3.1. Gestione Utenti

Il sistema abilita la gestione completa dell'account, compresi:

- Registrazione utente tramite **/api/auth/register/**
- Login basato su JWT tramite **/api/auth/login/**
- Attivazione account tramite email

- Reimpostazione e recupero password
- Modifica profilo e gestione luoghi preferiti
- Logout sicuro con invalidazione del refresh-token

Questo modulo garantisce privacy e sicurezza prevenendo il riutilizzo dei token e proteggendo le informazioni degli utenti.

### 3.2. Pianificazione Viaggi e Ricerca

L'endpoint **/api/auth/plan-trip/** permette agli utenti di inviare origine, destinazione, orario e preferenze di viaggio. Il backend comunica con l'API GraphQL di OTP per generare opzioni multimodali ottimizzate, raggruppate per modalità di trasporto come camminata o autobus.

La cronologia ricerche è registrata per tutti gli utenti, inclusi gli anonimi tramite session key. Quando un utente anonimo si registra o accede, le precedenti ricerche vengono automaticamente collegate al suo nuovo account.

### 3.3. Fermate e Orari Stazioni

Il backend espone endpoint per informazioni sul trasporto pubblico:

- **/api/auth/stops/** per recuperare tutte le fermate
- **/api/auth/station/<stop\_id>/** per arrivi e partenze di una fermata specifica

Il template **stop\_schedule.html** visualizza i risultati, spesso accessibili tramite QR code posizionati nelle stazioni.

### 3.4. Prenotazioni e Risparmio CO<sub>2</sub>

Gli utenti possono prenotare viaggi e salvare lo storico. Ogni prenotazione include:

- Distanza (km)
- CO<sub>2</sub> emessa (kg)
- CO<sub>2</sub> risparmiata (kg), basata sulla modalità di trasporto

Questa funzionalità promuove la sostenibilità e la mobilità verde.

### 3.5. Gestione Feedback

Gli utenti possono inviare feedback tramite `/api/auth/feedback/`, supportando il miglioramento del servizio.

## 4. Funzionalità Amministrative

Il pannello Django Admin fornisce strumenti per analizzare le performance e supportare la gestione operativa. Tabelle principali includono:

- **Utenti:** tipo (lavoratore/studente/altro) e codice fiscale
- **Ricerche:** modalità, coordinate, sessione/utente
- **Prenotazioni:** origine, destinazione, modalità, metriche CO<sub>2</sub>
- **Luoghi preferiti, Feedback e altri**

Logica condizionale assicura che:

- Gli ID siano mostrati solo per utenti autenticati
- Gli utenti anonimi siano mappati tramite session key
- I campi critici siano protetti in sola lettura

## 5. Sicurezza e Gestione Errori

### 5.1 Autenticazione

- Token JWT tramite `rest_framework_simplejwt`
- Blacklist token dopo il logout
- Rotazione refresh token e scadenza

## 5.2 Gestione Errori

Tutte le risposte API seguono un formato JSON unificato con codici HTTP significativi. I fallimenti da API esterne restituiscono:

- **503 Service Unavailable**

Categorie supportate:

- **2xx**: Successo
- **4xx**: Errori di input, accesso non autorizzato, conflitti
- **5xx**: Errori server o servizi esterni

in dettaglio:

### A. /api/auth/register/ (POST)

- **201**: Utente registrato
- **400**: Input non valido, campi mancanti, password debole
- **409**: Email/username già esistente

### B. /api/auth/login/ (POST)

- **200**: Login effettuato con successo
- **400**: Credenziali non valide, richiesta malformata
- **401**: Autenticazione fallita

### C. /api/auth/token/refresh/ (POST)

- **200**: Token aggiornato
- **400**: Token di refresh non valido/scaduto
- **401**: Non autenticato

### D. /api/auth/activate/<uidb64>/<token>/ (GET)

- **200**: Account attivato
- **400**: Token o ID utente non valido
- **403**: Utente bloccato
- **404**: Utente non trovato

#### **E. /api/auth/password-reset/ (POST)**

- **200:** Link di reset inviato
- **400:** Formato email non valido
- **404:** Email non trovata (può comunque restituire 200 per motivi di sicurezza)

#### **F. /api/auth/password-reset-confirmed/<uidb64>/<token>/ (POST)**

- **200:** Password reimpostata con successo
- **400:** Token non valido/scaduto, password debole
- **403:** Utente bloccato

#### **G. /api/auth/search (GET/POST)**

- **200:** Successo
- **400:** Input non valido
- **401:** Non autenticato
- **404:** Ricerca non trovata

#### **H. /api/auth/places (GET/POST)**

- **200:** Successo
- **201:** Luogo aggiunto
- **400:** Input non valido
- **401:** Non autenticato
- **404:** Luogo non trovato

#### **I. /api/auth/track (GET)**

- **200:** Successo
- **400:** Parametro id mancante
- **404:** Nessuna attività di ricerca trovata

#### **J. /api/auth/booking (GET/POST)**

- **200:** Successo
- **201:** Prenotazione creata
- **400:** Input non valido

- **401:** Non autenticato
- **409:** Conflitto di prenotazione

#### **K. /api/auth/feedback (POST)**

- **201:** Feedback inviato
- **400:** Input non valido

#### **L. /api/auth/profile (GET/PUT)**

- **200:** Successo
- **400:** Input non valido
- **401:** Non autenticato
- **404:** Utente non trovato
- **415:** Tipo di media non supportato

#### **M. /api/auth/places/ (PUT)**

- **200:** Successo
- **400:** Input non valido
- **401:** Non autenticato
- **404:** Luogo non trovato

#### **N. /api/auth/logout (POST)**

- **205:** Logout avvenuto con successo
- **400:** Token di refresh mancante/non valido
- **401:** Non autenticato

#### **O. /api/auth/plan-trip/ (POST/GET)**

- **200:** Successo
- **400:** Input non valido, errore di validazione
- **401:** Non autenticato (se richiesto)
- **503:** Errore API esterna/rete

#### **P. /api/auth/stops/ (GET/POST)**

- **200:** Successo

- **400:** Input non valido
- **503:** Errore API esterna/rete

#### **Q. /api/auth/station/<stop\_id>/ (GET/POST)**

- **200:** Successo
- **400:** Input non valido
- **404:** Fermata non trovata
- **503:** Errore API esterna/rete

#### **R. /api/auth/email/change/confirm/<uidb64>/<new\_email\_b64>/<token>/ (GET)**

- **200:** Email modificata con successo (link valido, token valido, email aggiornata)
- **400:**
  - Link non valido o malformato (uidb64 / new\_email\_b64 non validi o utente inesistente)
  - Link di conferma email non valido o scaduto (verifica token fallita)

## 6. API Endpoints

È fornita una lista completa di endpoint che coprono autenticazione, ricerca, prenotazioni, feedback, pianificazione viaggi e dati trasporto. Il sistema definisce chiari codici errore e comportamenti attesi.

- **/api/auth/register/** — Registrazione nuovo utente
- **/api/auth/login/** — Login e ottenimento token JWT
- **/api/auth/token/refresh/** — Aggiornamento del token di accesso JWT
- **/api/auth/activate/<uidb64>/<token>/** — Attivazione dell'account
- **/api/auth/password-reset/** — Richiesta reset password
- **/api/auth/password-reset-confirm/<uidb64>/<token>/** — Conferma reset password
- **/api/auth/search/** — Cronologia ricerche
- **/api/auth/places/** — Gestione luoghi preferiti
- **/api/auth/track/** — Tracciamento attività utente

- **/api/auth/booking/** — Prenotazione viaggi
- **/api/auth/feedback/** — Invio feedback
- **/api/auth/profile/** — Recupero/aggiornamento profilo utente
- **/api/auth/logout/** — Logout
- **/api/auth/confirm-email-change/<uidb64>/<new\_email\_b64>/<token/>** — Conferma cambio email (gestione profilo)
- **/api/auth/plan-trip/** — Pianificazione viaggio (integrazione con OTP)
- **/api/auth/stops/** — Recupero di tutte le fermate
- **/api/auth/station/<stop\_id/>** — Orari e informazioni della stazione

## 7. Conclusione

Il sistema ATTRACTION-BACKEND offre un'architettura backend completa e pronta alla produzione per applicazioni di trasporto moderne. Attraverso pianificazione integrata, accesso sicuro, prenotazioni e analisi, intelligenza delle stazioni e controllo amministrativo, il sistema abilita un ambiente MaaS scalabile progettato per ottimizzare l'accessibilità alla mobilità e ridurre l'impatto ambientale.

La sua architettura estensibile permette futuri miglioramenti come raccomandazioni basate su machine learning, integrazione con sistemi di pagamento e reti di trasporto multimodali più ampie.