

Documentazione delle API del Sistema di Prenotazione CAB

Questa documentazione fornisce una descrizione dettagliata delle API utilizzate nel sistema di Prenotazione CAB. Le API sono state create utilizzando Flask, un micro framework web in Python, e consentono al front-end (HTML, CSS, JavaScript) di interagire con il back-end per effettuare operazioni di creazione, lettura, aggiornamento e cancellazione (CRUD) sulle prenotazioni memorizzate in un database SQLite.

Introduzione

Le API del sistema di prenotazioni permettono di

- **Creare** una nuova prenotazione.
- **Recuperare** tutte le prenotazioni esistenti.
- **Aggiornare** una prenotazione esistente.
- **Eliminare** una prenotazione.

Il client (front-end) comunica con il server (back-end) attraverso richieste HTTP. Le risposte sono fornite in formato JSON.

Panoramica delle API

Metodo HTTP	Endpoint	Descrizione
POST	/prenotazioni	Aggiunge una nuova prenotazione
GET	/prenotazioni	Recupera tutte le prenotazioni
PUT	/prenotazioni/<int:id>	Modifica una prenotazione
DELETE	/prenotazioni/<int:id>	Cancella una prenotazione

Dettaglio degli Endpoint

1. Aggiungi Prenotazione

- **Endpoint:** `/prenotazioni`
- **Metodo HTTP:** `POST`
- **Descrizione:** Consente di aggiungere una nuova prenotazione al sistema.

Richiesta

- **Headers:**

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

- **Body (JSON):**

```
{  
  "nome": "string",  
  "data": "string (formato YYYY-MM-DD)",  
  "ora": "string (formato HH:MM)"  
}
```

- **Parametri:**

- `nome` (string, obbligatorio): Nome e cognome del cliente.
- `data` (string, obbligatorio): Data della prenotazione nel formato `YYYY-MM-DD`
- `ora` (string, obbligatorio): Ora della prenotazione nel formato `HH:MM`

Risposta

- **Successo (201 Created):**

- **Body (JSON):**

```
{  
  "nome": "string",  
  "data": "string",
```

```
"ora": "string"
}
```

Errori:

- **400 Bad Request:** Dati mancanti o non validi.

```
{
  "errore": "Descrizione dell'errore"
}
```

2. Ottieni Prenotazioni

- **Endpoint:** `/prenotazioni`
- **Metodo HTTP:** `GET`
- **Descrizione:** Recupera l'elenco di tutte le prenotazioni esistenti.

Richiesta

- **Headers:** Nessuno specifico.
- **Body:** Vuoto.

Risposta

- **Successo (200 OK):**
 - **Body (JSON):** Array di prenotazioni.

```
[  
  [id, "nome", "data", "ora"],  
  ...  
]
```

Ogni elemento dell'array rappresenta una prenotazione con:

- `id` (integer): Identificativo univoco della prenotazione.
 - `nome` (string): Nome del cliente.
 - `data` (string): Data della prenotazione.
 - `ora` (string): Ora della prenotazione.
- **Errori:** Nessuno specifico.

3. Modifica Prenotazione

- **Endpoint:** `/prenotazioni/<int:id>`
- **Metodo HTTP:** `PUT`
- **Descrizione:** Modifica le informazioni di una prenotazione esistente.

Richiesta

- **Headers:**

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

- **URL Parameters:**
 - `id` (integer, obbligatorio): Identificativo univoco della prenotazione da modificare.
- **Body (JSON):**

```
{
  "nome": "string",
  "data": "string (formato YYYY-MM-DD)",
  "ora": "string (formato HH:MM)"
}
```

Risposta

- **Successo (200 OK):**
 - **Body (JSON):**

```
{
  "id": integer,
  "nome": "string",
  "data": "string",
  "ora": "string"
}
```

- **Errori: 400 Bad Request:** Dati mancanti o non validi.
404 Not Found: Prenotazione con l'ID specificato non trovata.

4. Cancella Prenotazione

- **Endpoint:** `/prenotazioni/<int:id>`
- **Metodo HTTP:** `DELETE`
- **Descrizione:** Elimina una prenotazione esistente dal sistema.

Richiesta

- **Headers:** Nessuno specifico.
- **URL Parameters:**
 - `id` (integer, obbligatorio): Identificativo univoco della prenotazione da eliminare.
- **Body:** Vuoto.

Risposta

- **Successo (204 No Content):** La prenotazione è stata eliminata con successo. Nessun contenuto restituito.
- **Errori:**
 - **404 Not Found:** Prenotazione con l'ID specificato non trovata.

Error Handling

Il sistema gestisce gli errori restituendo codici di stato HTTP appropriati e messaggi di errore nel formato JSON.

Codici di Stato e Significati

- **200 OK:** La richiesta è andata a buon fine.
- **201 Created:** Una nuova risorsa è stata creata con successo.
- **204 No Content:** La richiesta è andata a buon fine, ma non c'è contenuto da restituire.
- **400 Bad Request:** La richiesta non può essere elaborata a causa di un errore del client (es. dati non validi).
- **404 Not Found:** La risorsa richiesta non esiste.
- **500 Internal Server Error:** Errore generico del server.

Formato dei Messaggi di Errore

In caso di errore, la risposta conterrà un corpo JSON con una chiave `"errore"` descrive il problema:

```
{  
  "errore": "Descrizione dettagliata dell'errore."  
}
```