

Open Door Developer Handbook

Open Door Key Variables and Cross-File References

This document summarizes key variables and concepts that are shared across the frontend and backend of the Open Door app. Keep this handy as the project grows.

1. `BASE_URL`

****Used in:**** `/frontend/app.js`

****Purpose:**** Specifies the base API endpoint used for all `fetch()` requests.

```
```js
const BASE_URL = "https://open-door-backend.onrender.com";
```
```

- Switch to localhost for local development
- Ensure it matches the backend deployment URL in production

2. `content`

****Used in:****

- `app.js` (POST request body)
- `app.py` (`data.get('content')`)

****Purpose:**** Holds the user's submitted message content.

```
```js
body: JSON.stringify({ content: message })
```
```

```
```python
message = data.get('content', "").strip()
```
```

- Must match between frontend and backend for posts to work

3. `timestamp`

****Used in:****

- SQLite DB: `DEFAULT CURRENT_TIMESTAMP`
- `app.py`: selected in `/get-messages`
- `app.js`: displayed via `formatTimestamp(msg.timestamp)`

****Purpose:**** Records when the message was posted and displays a friendly timestamp.

- Format consistently in frontend to avoid display errors

4. `flagged`

****Used in:****

- SQLite DB schema
- Backend moderation logic
- Admin panel rendering

****Purpose:**** Indicates whether a message was flagged by soft moderation.

- Stored as `INTEGER` (0 or 1)
- Cast with `int()` or `!!` as needed

5. `ADMIN_KEY`

****Used in:**** `app.py`, `/admin` and `/delete-message` routes

****Purpose:**** Controls access to protected admin functions.

```
```python
```

```
if key != ADMIN_KEY:
```

```
 return "Unauthorized", 403
```

```
```
```

- Passed via query string `?key=pass1029`
- Replace with login auth in future

Suggested Improvements

- Move `BASE_URL` and `ADMIN_KEY` into `.env` or config file
- Document payload structures (`{ content: ... }`)
- Add consistent timestamp formatting utils

Open Door API Routes & Database Schema Reference

This document outlines the core API endpoints and the structure of the SQLite database used in Open Door.

API Routes

These are the RESTful endpoints exposed by the Flask backend. Each route is responsible for specific functionality and communicates with the frontend via JSON.

`GET /get-messages`

****Purpose:****

Fetch all messages from the database, ordered by most recent.

****Returns:****

```
```json
[
 {
 "id": 1,
 "content": "Your message here",
 "timestamp": "2025-04-30 14:23:05",
 "flagged": 0
 },
 ...
]
...

```

## **`POST /post-message`**

**\*\*Purpose:\*\***

Receive a new message and insert it into the database, with optional soft moderation.

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

```
```json
{
  "content": "I'm having a hard day"
}
...

---
```

****Returns:****

```
```json
```

```
{
 "status": "Message saved successfully!",
 "flagged": true
}
```

```
```
```

```
---
```

`GET /admin?key=pass1029`

****Purpose:****

Display all messages in an admin panel with moderation visibility.

- Requires query parameter: `key`
- Returns HTML (not JSON)

```
---
```

`POST /delete-message/<id>?key=pass1029`

****Purpose:****

Delete a message by its ID. Used from the admin dashboard.

- Requires admin key
- Returns a redirect to the admin page

```
---
```

`GET /force-init`

****Purpose:****

Initialize the database manually. Useful for reboots or re-deployments.
Returns a plain text message confirming DB initialization.

Database Schema

SQLite database `database.db` contains one table:

Table: `messages`

| Column | Type | Description | |
|-------------|----------|------------------------------------|--|
| ----- | ----- | ----- | |
| `id` | INTEGER | Primary key, auto-incremented ID | |
| `content` | TEXT | Message content from the user | |
| `timestamp` | DATETIME | Defaults to current time on insert | |
| `flagged` | INTEGER | 1 if flagged by moderation, else 0 | |

Suggestions for Growth

- Add a `replied_to` or `parent_id` column for future message threads
- Add `email` or `user_id` (if authentication is implemented later)
- Move `flagged` logic to a separate moderation table (if needed)

