# **fbAuto Backend API Documentation**

This document describes all available API endpoints for the fbAuto backend.  
The server is built with Express.js and integrates automation services, Facebook Messenger, and job posting features.

## **Base URL**

* Local Development: http://localhost:5000/api
* Production: ‘BACKEND URL’

All endpoints are prefixed with /api, unless noted (like Messenger webhooks).

## **Health Check**

### **GET /health**

Returns the current timestamp.  
Useful for checking if the server is alive.

Response:

1737858327000

## **Authentication Routes**

Base: /api/auth

| **Method** | **Endpoint** | **Auth** | **Description** |
| --- | --- | --- | --- |
| POST | /register | ❌ | Register a new user |
| POST | /login | ❌ | Login with credentials, returns JWT + sets cookie |
| POST | /logout | ✅ | Logout and clear cookie |
| GET | /profile | ✅ | Get logged-in user profile |
| GET | /cookie | ❌ | Debug endpoint: returns stored cookie token |

## **Candidate Routes**

Base: /api/candidates

| **Method** | **Endpoint** | **Auth** | **Description** |
| --- | --- | --- | --- |
| GET | / | ✅ | Get all candidates |
| DELETE | /:id | ✅ | Delete candidate by ID |

## **Cookie Management Routes (use this when the login fails or cookies expires)**

Base: /api/cookies

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /import | Import cookies for a user |
| POST | /bulk-import | Bulk import cookies for multiple users |
| GET | /status/:email | Get cookie status for a specific user |
| GET | /users | Get all users with valid cookies |
| GET | /test/:email | Test cookies for a specific user |
| DELETE | /:email | Delete cookies for a user |
| PATCH | /invalidate/:email | Mark cookies as invalid |
| DELETE | /cleanup/expired | Cleanup expired cookies |

## **Job Routes (here /process-all are important)**

Base: /api/jobs  
Requires JWT authentication (verifyToken).

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | / | Create a new job |
| GET | / | Get all jobs for the logged-in user |
| GET | /pending | Get all pending jobs |
| GET | /:jobId | Get job by ID |
| PUT | /:jobId | Update job by ID |
| DELETE | /:jobId | Delete job by ID |
| POST | /:jobId/post-to-facebook | Post a job to Facebook |
| GET | /:jobId/analytics | Get job analytics |
| POST | /pending/process-all | Processes all pending Facebook job posts submitted by the user via frontend |
| POST | /pending/:jobId/process | Process a single pending job |
| ``` |  |  |

### **Special Mention: POST /api/jobs/pending/process-all**

* This endpoint scans all pending jobs created by the logged-in user.
* It then automatically posts them to Facebook (using stored cookies/credentials).
* Designed to batch-process posts created in the frontend without requiring manual execution one by one.

Example:

POST http://localhost:5000/api/jobs/pending/process-all

Response:

{

"success": true,

"processedJobs": 5,

"status": "All pending jobs have been processed and posted to Facebook"

}

## **Facebook Automation Routes (here just insert the facebook creds just for once that it it the other no need to be used)**

Base: /api/facebook  
Requires JWT authentication.

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /credentials | Save Facebook login credentials |
| GET | /credentials | Get stored Facebook credentials |
| POST | /automation/run | Run Facebook automation |
| GET | /automation/status | Get current automation status |
| POST | /automation/stop | Stop Facebook automation |
| POST | /automation/custom | Run custom automation |
| POST | /job-post/run | Run job post automation (single user) |
| POST | /job-post/run-all | Run job post automation for all users |
| GET | /job-post/status | Get job post automation status |
| POST | /job-post/stop | Stop job post automation |

## **Global Automation Routes**

Base: /api/automation

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /status | Get system automation status & resource usage |
| POST | /manual-comment-monitoring/:userId | Run manual comment monitoring for a user |
| POST | /job-post/:userId | Run job posting automation for a user (immediate engagement) |

## **Messenger & Context Routes (this the route we are using to pass context in the messenger link we are using in the POST)**

### **Messenger Redirect**

* GET /api/messenger-redirect?context=...

Redirects candidates to Messenger with job context attached.  
If the job is closed/expired, returns a Job Closed page.

### **Messenger Webhooks**

These are special routes (no /api prefix):

* GET /webhook/messenger → Used by Facebook for verification handshake.
* POST /webhook/messenger → Receives and processes Messenger events (referrals, messages, postbacks).

## **Job Context Routes**

Base: /api

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /store-job-context | Store job context |
| GET | /job-context/:sessionId | Get job context by session ID |
| POST | /link-context-to-user | Link a job context session to a user |
| GET | /active-contexts | List all active job contexts (debugging) |

## **Examples**

### **Register User**

POST /api/auth/register

Content-Type: application/json

{

"email": "test@example.com",

"password": "secret"

}

Response

{

"success": true,

"user": {

"id": "123",

"email": "test@example.com"

}

}

# **Notes**

* Authentication is via JWT stored in cookies (token).
* Most routes under /auth, /jobs, /facebook, and /candidates require valid login.
* Messenger webhooks are public because Facebook needs to call them.