This guide details how to create an API Gateway endpoint to securely expose your Lambda function, enabling resume uploads from your frontend.

I. Creating the API Gateway Endpoint:

1. **Open API Gateway:** Log in to the AWS Management Console and navigate to the API Gateway service.
2. **Create New API:** Click "Create API." Choose "REST API" and give it a name (e.g., ResumeParserAPI). Click "Create API."
3. **Create Resource:** Under "Resources," click "Create Resource." Give it a name (e.g., resume) and ensure the path is /resume. Click "Create Resource."
4. **Create POST Method:** Select the resume resource and choose "Create Method." Select POST. Click the "check" icon.
5. **Configure Integration Request:** This is where you connect to your Lambda function:
    - **Integration type:** Select "AWS Service."
    - **AWS Region:** Choose the region where your Lambda function is deployed.
    - **AWS Service:** Select "Lambda function proxy."
    - **Lambda function:** Choose your Lambda function (e.g., resumeProcessor).
    - **Authorization:** For this stage, you may leave authorization blank, but for a production environment, you would want to add proper authentication and authorization to secure your API.
    - **Content-Handling:** Make sure the API Gateway handles multipart/form-data correctly. This is crucial for receiving the resume file uploads from your frontend. You'll likely need to modify the integration request settings to add support for multipart/form-data.
6.
7. **Deploy API:** Click "Deploy API."
    - **Deployment Stage:** Choose a stage name (e.g., prod).
    - **Deploy:** Click "Deploy."
8.
9. **Note the Invoke URL:** After deployment, API Gateway provides an Invoke URL. This URL is the endpoint that your frontend will use to send resume uploads. Copy this URL; you will need it for your frontend code.

II. Important Security Considerations:

- **IAM Role:** Ensure that the IAM role you chose during the integration step has the necessary permissions to invoke your Lambda function. This is absolutely vital for security. You must give it only the permission to invoke your specific Lambda function, not all Lambda functions.
- **Authentication and Authorization (Production):** For a production environment, it's absolutely critical to add authentication and authorization to secure your API Gateway endpoint. Avoid making it publicly accessible without proper authorization. You could use

API Keys, AWS_IAM, OAuth 2.0, or other methods to secure your API. This step has been left out here for simplicity, but it is crucial for secure deployment.

- **Error Handling:** Implement proper error handling in your API Gateway configuration. You should configure appropriate error responses to communicate errors to the client (your frontend).

This detailed guide helps you set up a secure API Gateway endpoint. Carefully test the integration with your Lambda function after deployment, ensuring it correctly receives and processes resume uploads. Always check and double-check your IAM roles and API Gateway settings for better security. Remember that this API Gateway setup is a simplified guide; in a production environment, add proper authentication and authorization.