AI Agent

Talent Acquisition

powered by aws

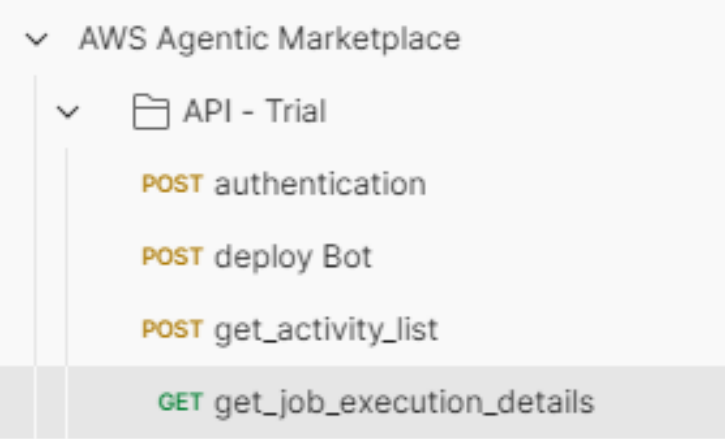
AUTOMATION ANYWHERE

1. Introduction:

This AI agent analyzes resumes against job descriptions to calculate role-fit scores and provide structured candidate evaluations covering experience, education, and skills. It helps recruiters and hiring managers make faster, more objective hiring decisions by eliminating bias and streamlining the candidate screening process.

2. Instructions:

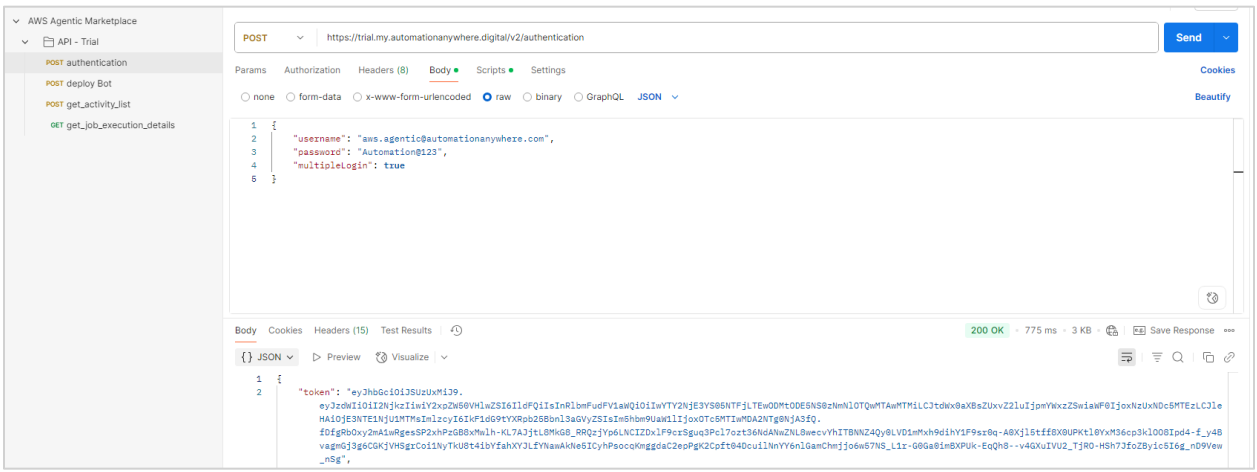
A) You will need Postman installed on your system
B) Download the "AWS_API_Based_Talent_Acquisition_AIAgent.json" API Collection
C) Import the .json file into Postman as an API Collection
D) You will see 4 API Calls



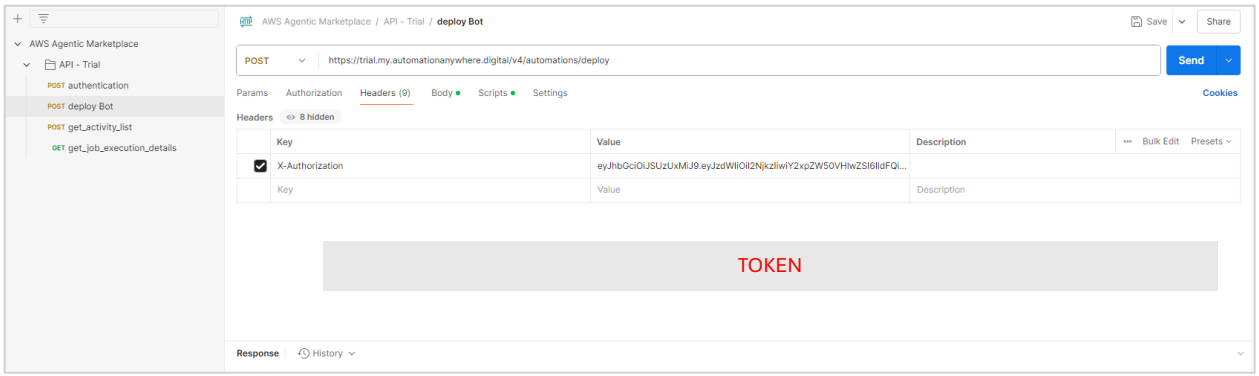
3. Using the APIs

A) **Authentication:** Update the Body with the provided username and password and hit SEND. Once the call is successful it will give you a 200 OK response and generate a TOKEN.

AI Agent – Talent Acquisition



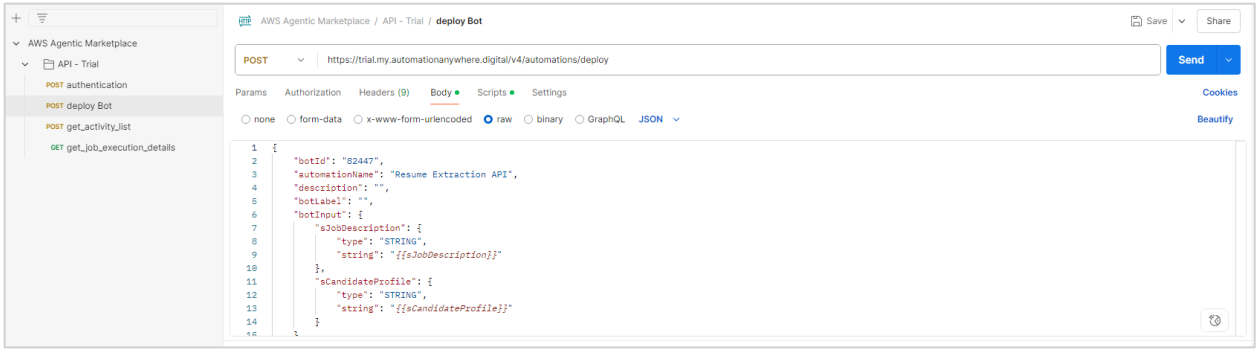
B) **Deploy:** Select the Deploy Bot API and update the HEADER section. Replace the X-Authorization value with the recently provided TOKEN.



C) **Deploy:** Update the Body to pass the input parameters. This AI Agent accepts 2 variables (Both these should be passed in the form of text):

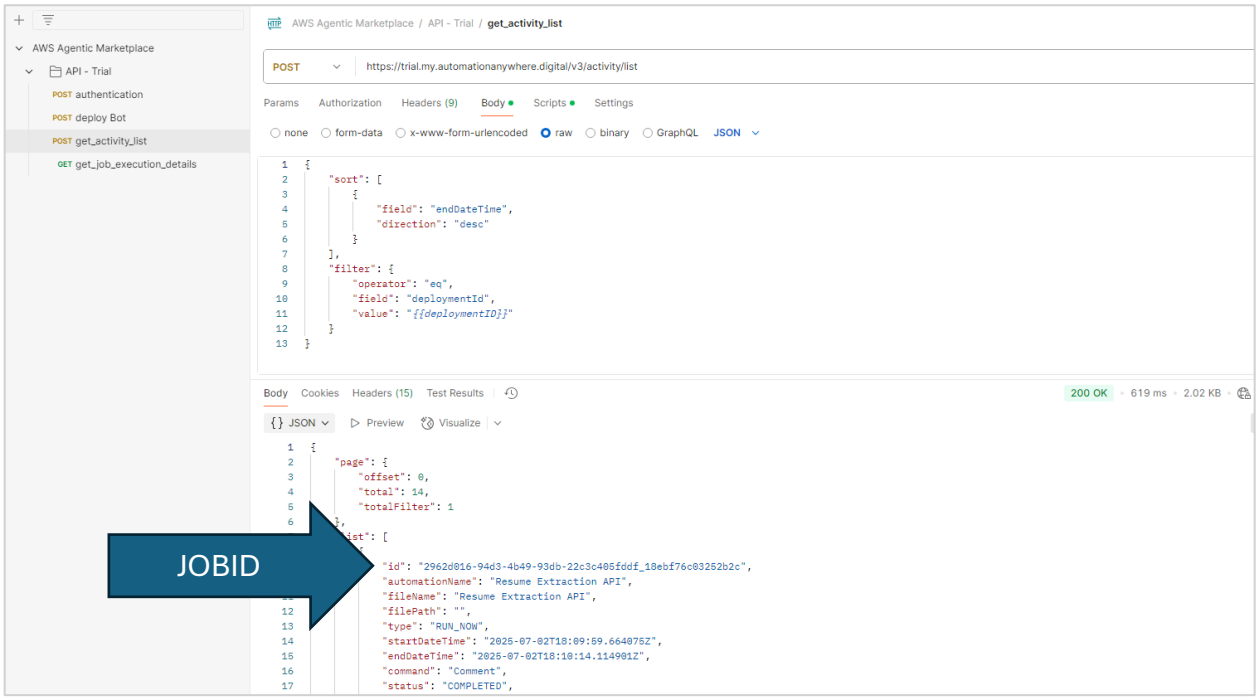
i) *Job Description*: Role and job details, requirements and experience needed.

ii) *Candidate Profile*: The candidate's resume and prior education/experience.

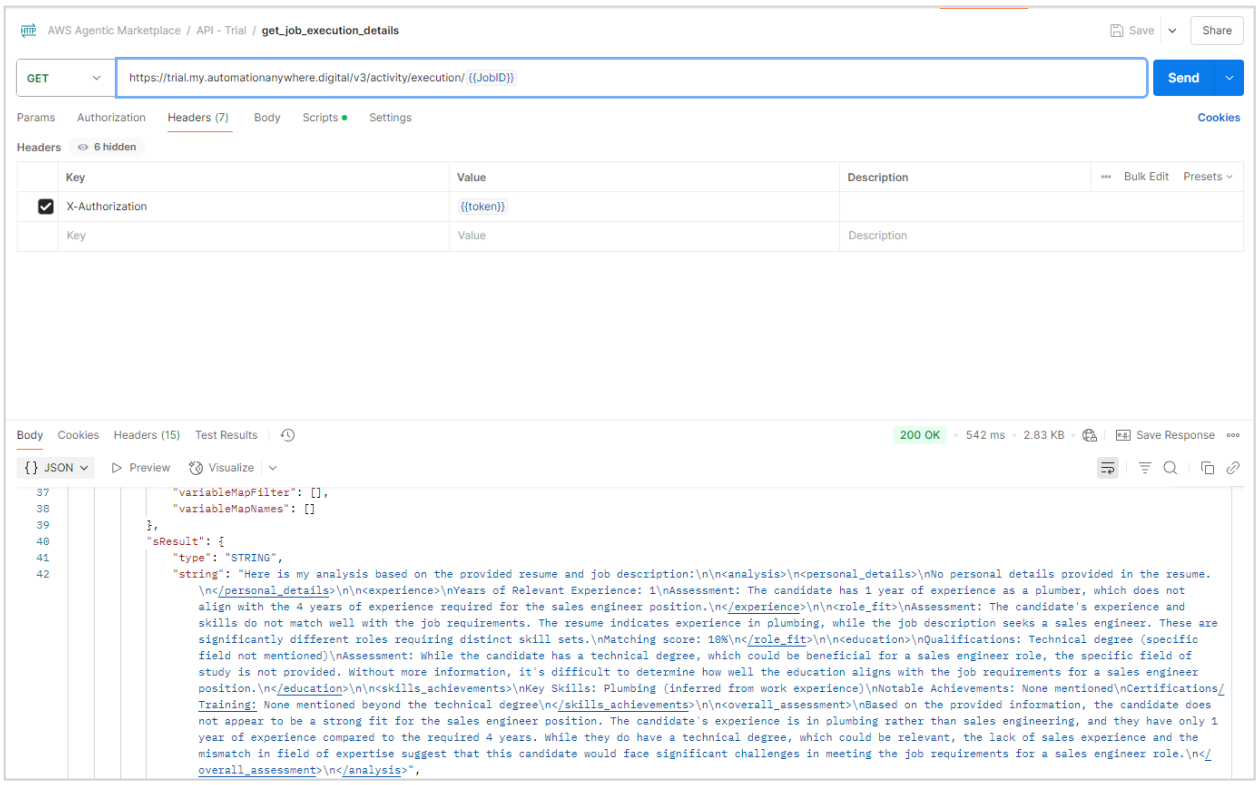


D) **Deploy:** Once the API is executed the response will be 200 OK and will deliver a *DeploymentID* and the *AutomationName*

E) **Get Activity:** This API will show you the progress and current state of the Agent. When you have long-running processes, this API will allow you to capture the completion %. You must update the HEADER with the TOKEN and pass the DEPLOYMENTID in the Body of the API.



F) **Get Job Execution Details:** This API will fetch the API response from the AI Agent. You need to pass the TOKEN and the JOBID parameters in the API Call. A string containing the full analysis and response will be provided as part of the API response.



4. API-Based Agents Deployment in AWS:

There are many ways to use Automation Anywhere AWS API-Based AI Agents which allow customers to call an API-based AI Agent via a URL endpoint. These assets will help accelerate the build and deployment of AI-powered applications. Here are some simple ways to implement this, including using AWS Lambda functions and other AWS services:

A) AWS Lambda Functions

- **Trigger via API Gateway**: Set up an API Gateway to expose a REST endpoint. When a customer calls this endpoint, it triggers a Lambda function that interacts with your API-Based AI Agent.
- **Event-Driven**: Use AWS Lambda to handle events from other AWS services (e.g., S3 uploads, DynamoDB updates) and call your API-Based AI Agent based on these events.

B) Amazon API Gateway

- **REST API**: Create a REST API using API Gateway. This API can have various endpoints that map to different functionalities of your API-Based AI Agent.

- **WebSocket API**: For real-time communication, use WebSocket APIs to maintain a persistent connection between the client and your API-Based AI Agent.

C) AWS Step Functions 

- **Orchestration**: Use Step Functions to orchestrate multiple AWS services. For example, a customer calls an API Gateway endpoint, which triggers a Step Function that coordinates calls to Lambda functions, DynamoDB, and your API-Based AI Agent.

D) Amazon S3 

- **Static Website Hosting**: Host a static website on S3 that provides a user interface for interacting with your API-Based AI Agent. The website can call the API Gateway endpoints to communicate with the API-Based AI Agent.
- **Event Notifications**: Use S3 event notifications to trigger Lambda functions when new objects are created, which can then call your API-Based AI Agent.

E) Amazon SNS & SQS 

- **Message Queues**: Use Amazon Simple Notification Service (SNS) or Simple Queue Service (SQS) to handle messages between your customers and the API-Based AI Agent. Customers can send messages to an SNS topic or SQS queue, which triggers a Lambda function to process the message and call the API-Based AI Agent.

F) Amazon Bedrock 

- **AI Agent Platform**: Utilize Amazon Bedrock to deploy and manage your AI API-Based AI Agent. Bedrock supports various foundation models and provides a scalable platform for AI-driven interactions
- **Agent Builder or Workflow builder**: Leverage agentic and workflow builders inside Amazon Bedrock by configuring direct calls to extend solutions with the API-based AI Agents as tools.

G) Amazon Q 

Is a valuable addition to your solution. Amazon Q enables its own functionality, pulling multiple AWS features to work together, and further leveraging Amazon Q for API-Based AI Agent utilization.

- **Action/Flow Execution**: Use Amazon Q's API to define and execute specific actions or flows. For example, you can create a flow that handles customer inquiries, processes data, or triggers other AWS services.

- **API Calls**: The Lambda function can make API calls to Amazon Q to initiate these actions. This ensures that the interaction is seamless and efficient.
- **Step Functions:** These calls can include calling the API Gateway endpoint, triggering the Lambda function, and interacting with Amazon Q.
- **Enhance existing Automation Anywhere solutions**
    - **Automation Co-Pilot**: Use Automation Anywhere's Automation Co-Pilot to embed automation capabilities within your application. This can be configured to call the Amazon Q API for specific tasks.
    - **AI Agent Studio**: Leverage Automation Anywhere's AI Agent Studio to create and manage AI agents that can interact with Amazon Q. These agents can be programmed to handle various workflows and processes.