Zapier Natural Language Actions API

Contents

Full docs here: https://nla.zapier.com/api/v1/docs

Zapier Natural Language Actions gives you access to the 5k+ apps, 20k+ actions on Zapier's platform through a natural language API interface.

NLA supports apps like Gmail, Salesforce, Trello, Slack, Asana, HubSpot, Google Sheets, Microsoft Teams, and thousands more apps: https://zapier.com/apps

Zapier NLA handles ALL the underlying API auth and translation from natural language –> underlying API call –> return simplified output for LLMs. The key idea is you, or your users, expose a set of actions via an oauth-like setup window, which you can then query and execute via a REST API.

NLA offers both API Key and OAuth for signing NLA API requests.

1. Server-side (API Key): for quickly getting started, testing, and production scenarios where LangChain will only use actions exposed in the developer's Zapier account (and will use the developer's connected accounts on Zapier.com)
2. User-facing (Oauth): for production scenarios where you are deploying an end-user facing application and LangChain needs access to end-user's exposed actions and connected accounts on Zapier.com

This quick start will focus on the server-side use case for brevity. Review full docs or reach out

Skip to main content

This example goes over how to use the Zapier integration with a `SimpleSequentialChain`, then an `Agent`. In code, below:

```
%load_ext autoreload
%autoreload 2
```

```
import os

# get from https://platform.openai.com/
os.environ["OPENAI_API_KEY"] = os.environ.get("OPENAI_API_KEY", "")

# get from https://nla.zapier.com/demo/provider/debug (under User Information,
after logging in):
os.environ["ZAPIER_NLA_API_KEY"] = os.environ.get("ZAPIER_NLA_API_KEY", "")
```

Example with Agent

Zapier tools can be used with an agent. See the example below.

```
from langchain.llms import OpenAI
from langchain.agents import initialize_agent
from langchain.agents.agent_toolkits import ZapierToolkit
from langchain.utilities.zapier import ZapierNLAWrapper
```

```
## step 0. expose gmail 'find email' and slack 'send channel message' actions

# first go here, log in, expose (enable) the two actions:
https://nla.zapier.com/demo/start -- for this example, can leave all fields "Have
AI guess"
# in an oauth scenario, you'd get your own <provider> id (instead of 'demo') which
you route your users through first
```

```
llm = OpenAI(temperature=0)
zapier = ZapierNLAWrapper()
toolkit = ZapierToolkit.from_zapier_nla_wrapper(zapier)
agent = initialize_agent(toolkit.get_tools(), llm, agent="zero-shot-react-
description", verbose=True)
```

```
agent.run("Summarize the last email I received regarding Silicon Valley Bank. Send
```

Skip to main content

```
> Entering new AgentExecutor chain...
 I need to find the email and summarize it.
Action: Gmail: Find Email
Action Input: Find the latest email from Silicon Valley Bank
Observation: {"from__name": "Silicon Valley Bridge Bank, N.A.", "from__email":
"sreply@svb.com", "body_plain": "Dear Clients, After chaotic, tumultuous &
stressful days, we have clarity on path for SVB, FDIC is fully insuring all
deposits & have an ask for clients & partners as we rebuild. Tim Mayopoulos
<https://eml.svb.com/NjEwLUtBSy0yNjYAAAGKgoxUeBCLAyF_NxON97X4rKEaNBLG",
"reply_to__email": "sreply@svb.com", "subject": "Meet the new CEO Tim Mayopoulos",
"date": "Tue, 14 Mar 2023 23:42:29 -0500 (CDT)", "message_url":
"https://mail.google.com/mail/u/0/#inbox/186e393b13cfdf0a", "attachment_count":
"0", "to__emails": "ankush@langchain.dev", "message_id": "186e393b13cfdf0a",
"labels": "IMPORTANT, CATEGORY_UPDATES, INBOX"}
Thought: I need to summarize the email and send it to the #test-zapier channel in
Slack.
Action: Slack: Send Channel Message
Action Input: Send a slack message to the #test-zapier channel with the text
"Silicon Valley Bank has announced that Tim Mayopoulos is the new CEO. FDIC is
fully insuring all deposits and they have an ask for clients and partners as they
rebuild."
Observation: {"message__text": "Silicon Valley Bank has announced that Tim
Mayopoulos is the new CEO. FDIC is fully insuring all deposits and they have an
ask for clients and partners as they rebuild.", "message__permalink":
"https://langchain.slack.com/archives/C04TSGU0RA7/p1678859932375259", "channel":
"C04TSGU0RA7", "message__bot_profile__name": "Zapier", "message__team":
"T04F8K3FZB5", "message__bot_id": "B04TRV4R74K", "message__bot_profile__deleted":
"false", "message__bot_profile__app_id": "A024R9PQM", "ts_time": "2023-03-
15T05:58:52Z", "message__bot_profile__icons__image_36": "https://avatars.slack-
edge.com/2022-08-02/3888649620612_f864dc1bb794cf7d82b0_36.png",
"message__blocks[]block_id": "kdZZ", "message__blocks[]elements[]type": "
['rich_text_section']"}
Thought: I now know the final answer.
Final Answer: I have sent a summary of the last email from Silicon Valley Bank to
the #test-zapier channel in Slack.

> Finished chain.
```

```
'I have sent a summary of the last email from Silicon Valley Bank to the #test-
zapier channel in Slack.'
```

If you need more explicit control, use a chain, like below.

```
from langchain.llms import OpenAI
from langchain.chains import LLMChain, TransformChain, SimpleSequentialChain
from langchain.prompts import PromptTemplate
from langchain.tools.zapier.tool import ZapierNLARunAction
from langchain.utilities.zapier import ZapierNLAWrapper
```

Skip to main content

```
## step 0. expose gmail 'find email' and slack 'send direct message' actions

# first go here, log in, expose (enable) the two actions:
https://nla.zapier.com/demo/start -- for this example, can leave all fields "Have
AI guess"
# in an oauth scenario, you'd get your own <provider> id (instead of 'demo') which
you route your users through first

actions = ZapierNLAWrapper().list()
```

```
## step 1. gmail find email

GMAIL_SEARCH_INSTRUCTIONS = "Grab the latest email from Silicon Valley Bank"

def nla_gmail(inputs):
    action = next((a for a in actions if a["description"].startswith("Gmail: Find
Email")), None)
    return {"email_data": ZapierNLARunAction(action_id=action["id"],
zapier_description=action["description"],
params_schema=action["params"]).run(inputs["instructions"])}
gmail_chain = TransformChain(input_variables=["instructions"], output_variables=
["email_data"], transform=nla_gmail)
```

```
## step 2. generate draft reply

template = """You are an assisstant who drafts replies to an incoming email.
Output draft reply in plain text (not JSON).

Incoming email:
{email_data}

Draft email reply:"""

prompt_template = PromptTemplate(input_variables=["email_data"], template=template)
reply_chain = LLMChain(llm=OpenAI(temperature=.7), prompt=prompt_template)
```

```
## step 3. send draft reply via a slack direct message

SLACK_HANDLE = "@Ankush Gola"

def nla_slack(inputs):
    action = next((a for a in actions if a["description"].startswith("Slack: Send
Direct Message")), None)
    instructions = f'Send this to {SLACK_HANDLE} in Slack: {inputs["draft_reply"]}'
    return {"slack_data": ZapierNLARunAction(action_id=action["id"],
zapier_description=action["description"],
params_schema=action["params"]).run(instructions)}
```

Skip to main content

```
slack_chain = TransformChain(input_variables=["draft_reply"], output_variables=
["slack_data"], transform=nla_slack)
```

```
## finally, execute

overall_chain = SimpleSequentialChain(chains=[gmail_chain, reply_chain,
slack_chain], verbose=True)
overall_chain.run(GMAIL_SEARCH_INSTRUCTIONS)
```

```
> Entering new SimpleSequentialChain chain...
{"from__name": "Silicon Valley Bridge Bank, N.A.", "from__email":
"sreply@svb.com", "body_plain": "Dear Clients, After chaotic, tumultuous &
stressful days, we have clarity on path for SVB, FDIC is fully insuring all
deposits & have an ask for clients & partners as we rebuild. Tim Mayopoulos
<https://eml.svb.com/NjEwLUtBSy0yNjYAAAGKgoxUeBCLAyF_NxON97X4rKEaNBLG",
"reply_to__email": "sreply@svb.com", "subject": "Meet the new CEO Tim Mayopoulos",
"date": "Tue, 14 Mar 2023 23:42:29 -0500 (CDT)", "message_url":
"https://mail.google.com/mail/u/0/#inbox/186e393b13cfdf0a", "attachment_count":
"0", "to__emails": "ankush@langchain.dev", "message_id": "186e393b13cfdf0a",
"labels": "IMPORTANT, CATEGORY_UPDATES, INBOX"}

Dear Silicon Valley Bridge Bank,

Thank you for your email and the update regarding your new CEO Tim Mayopoulos. We
appreciate your dedication to keeping your clients and partners informed and we
look forward to continuing our relationship with you.

Best regards,
[Your Name]
{"message__text": "Dear Silicon Valley Bridge Bank, \n\nThank you for your email
and the update regarding your new CEO Tim Mayopoulos. We appreciate your
dedication to keeping your clients and partners informed and we look forward to
continuing our relationship with you. \n\nBest regards, \n[Your Name]",
"message__permalink":
"https://langchain.slack.com/archives/D04TKF5BBHU/p1678859968241629", "channel":
"D04TKF5BBHU", "message__bot_profile__name": "Zapier", "message__team":
"T04F8K3FZB5", "message__bot_id": "B04TRV4R74K", "message__bot_profile__deleted":
"false", "message__bot_profile__app_id": "A024R9PQM", "ts_time": "2023-03-
15T05:59:28Z", "message__blocks[]block_id": "p7i",
"message__blocks[]elements[]elements[]type": "[['text']]",
"message__blocks[]elements[]type": "['rich_text_section']"}

> Finished chain.
```

```
'{"message__text": "Dear Silicon Valley Bridge Bank, \\n\\nThank you for your
email and the update regarding your new CEO Tim Mayopoulos. We appreciate your
dedication to keeping your clients and partners informed and we look forward to
continuing our relationship with you. \\n\\nBest regards, \\n[Your Name]",
```

Skip to main content

```
"D04TKF5BBHU", "message__bot_profile__name": "Zapier", "message__team":
"T04F8K3FZB5", "message__bot_id": "B04TRV4R74K", "message__bot_profile__deleted":
"false", "message__bot_profile__app_id": "A024R9PQM", "ts_time": "2023-03-
15T05:59:28Z", "message__blocks[]block_id": "p7i",
"message__blocks[]elements[]elements[]type": "[[\'text\']]",
"message__blocks[]elements[]type": "[\'rich_text_section\']"}'
```