



Linkup Beta Release - /research

Date: 2026-01-30

Owner: @Philippe Mizrahi

CONFIDENTIAL

Getting started with /research (beta)

The /research endpoint is designed for **deep, multi-step reasoning and search**, while keeping outputs concise and usable in agentic workflows.

Unlike standard search, /research can take more time to run (from a few seconds up to a few minutes) as it explores, validates, and synthesizes information before returning a final answer.

Because of this, /research is **asynchronous**.

Tutorial

1. Start a research task

To begin, send a POST request to /v1/research with:

- you `api key`, and
- your `query`
- the `output type` (only `sourcedAnswer` and `structured` are currently available)

The parameters are the same as /search, **except depth has been removed**.

```
curl --request POST --url "https://api.linkup.so/v1/research" \  
  --header "Authorization: Bearer <api-key>" \  
  --header "Content-Type: application/json" \  
  --data '{  
    "q": "<query>",  
    "outputType": "sourcedAnswer"  
  }'
```

This will immediately return a task ID:

```
{ "id": "<task-id>" }
```

2. Fetch the results

Use the `task ID` to poll the result endpoint:

Example (curl):

```
curl --request GET --url "https://api.linkup.so/v1/research/<task-id>" \  
  --header "Authorization: Bearer <api-key>"
```

While the task is running, the response status will be:

- pending
- processing

Once complete, the final response will be returned.

Full Python script

```
import time  
import httpx  
  
API_KEY = "<your-api-key>"
```

```
query = "Who are Linkup, the French AI startup, and who are the cofounders?"
```

```
def start_research(query: str) → str:
    response = httpx.post(
        "https://api.linkup.so/v1/research",
        headers={
            "Authorization": f"Bearer {API_KEY}",
            "Content-Type": "application/json",
        },
        json={
            "q": query,
            "outputType": "sourcedAnswer",
        },
    )
    response.raise_for_status()
    return response.json()["id"]

def get_results(task_id: str) → dict:
    response = httpx.get(
        f"https://api.linkup.so/v1/research/{task_id}",
        headers={"Authorization": f"Bearer {API_KEY}"},
    )
    response.raise_for_status()
    return response.json()

task_id = start_research(query)
print(f"Task started: {task_id}")

while True:
    results = get_results(task_id)
    if results.get("status") not in ["pending", "processing"]:
        break
    time.sleep(2)
```

```
print(results)
```

Design notes

- The endpoints is optimized for deep reasoning and evaluation, stopping once sufficient context is found
- Outputs can be concise or detailed based on prompt instructions.
 - Since the reasoning is LLM-based, detailed search instructions will be understood by the system
- Parameters are the same as with `/search` except for depth, that has been removed
- Output types can be `sourcedAnswer` or `structured`
- Latency can range from ~10 seconds to a few minutes, depending on the query
- Currently, the endpoint accepts a maximum of 10 requests in parallel per organization

Beta limitations

This is an early beta release:

- The endpoint is not yet publicly documented
- Specs, pricing, and behavior may evolve
- We're actively using real-world usage to guide improvements

Your feedback is extremely valuable — especially around:

- Result quality
- Latency vs usefulness

- Where it performs better or worse than deep /search
- Gaps for your specific use cases

Please let us know at : support@linkup.so