Claude Docs

Tools > Web search tool

Web search tool

Copy page

The web search tool gives Claude direct access to real-time web content, allowing it to answer questions with up-to-date information beyond its knowledge cutoff. Claude automatically cites sources from search results as part of its answer.

Please reach out through our feedback form to share your experience with the web search tool.

Supported models

Web search is available on:

- Claude Sonnet 4.5 (`claude-sonnet-4-5-20250929`)
- Claude Sonnet 4 (`claude-sonnet-4-20250514`)
- Claude Sonnet 3.7 (deprecated) (`claude-3-7-sonnet-20250219`)
- Claude Haiku 4.5 (`claude-haiku-4-5-20251001`)
- Claude Haiku 3.5 (deprecated) (`claude-3-5-haiku-latest`)
- Claude Opus 4.5 (`claude-opus-4-5-20251101`)
- Claude Opus 4.1 (`claude-opus-4-1-20250805`)
- Claude Opus 4 (`claude-opus-4-20250514`)

How web search works

When you add the web search tool to your API request:

1. Claude decides when to search based on the prompt.
2. The API executes the searches and provides Claude with the results. This process may repeat multiple times throughout a single request.
3. At the end of its turn, Claude provides a final response with cited so

Ask Docs

ⓘ Your organization's administrator must enable web search in Console.

Provide the web search tool in your API request:

Shell

```
curl https://api.anthropic.com/v1/messages \
    --header "x-api-key: $ANTHROPIC_API_KEY" \
    --header "anthropic-version: 2023-06-01" \
    --header "content-type: application/json" \
    --data '{
        "model": "claude-sonnet-4-5",
        "max_tokens": 1024,
        "messages": [
            {
                "role": "user",
                "content": "What is the weather in NYC?"
            }
        ],
        "tools": [{
            "type": "web_search_20250305",
            "name": "web_search",
            "max_uses": 5
        }]
    }'
```

Tool definition

The web search tool supports the following parameters:

JSON

Ask Docs

```
  "name": "web_search",

  // Optional: Limit the number of searches per request
  "max_uses": 5,

  // Optional: Only include results from these domains
  "allowed_domains": ["example.com", "trusteddomain.org"],

  // Optional: Never include results from these domains
  "blocked_domains": ["untrustedsource.com"],

  // Optional: Localize search results
  "user_location": {
    "type": "approximate",
    "city": "San Francisco",
    "region": "California",
    "country": "US",
    "timezone": "America/Los_Angeles"
  }
}
```

Max uses

The `max_uses` parameter limits the number of searches performed. If Claude attempts more searches than allowed, the `web_search_tool_result` will be an error with the `max_uses_exceeded` error code.

Domain filtering

When using domain filters:

- Domains should not include the HTTP/HTTPS scheme (use `example.com` instead of `https://example.com`)
- Subdomains are automatically included (`example.com` covers `docs.example.com`)
- Specific subdomains restrict results to only that subdomain (`docs.example.com` returns only results from that subdomain, not from `example.com` or `api.example.com`)
- Subpaths are supported and match anything after the path (`example.com/blog` matches `example.com/blog/post-1`)
- You can use either `allowed_domains` or `blocked_domains` , but not both in the same request.

Ask Docs

Only one wildcard (*) is allowed per domain entry, and it must appear after the domain part (in the path)

- Valid: `example.com/*` , `example.com/*/articles`
- Invalid: `*.example.com` , `ex*.com` , `example.com/*/news/*`

Invalid domain formats will return an `invalid_tool_input` tool error.

> ⓘ Request-level domain restrictions must be compatible with organization-level domain restrictions configured in the Console. Request-level domains can only further restrict domains, not override or expand beyond the organization-level list. If your request includes domains that conflict with organization settings, the API will return a validation error.

Localization

The `user_location` parameter allows you to localize search results based on a user's location.

- `type` : The type of location (must be `approximate`)
- `city` : The city name
- `region` : The region or state
- `country` : The country
- `timezone` : The IANA timezone ID.

Response

Here's an example response structure:

Ask Docs

```
  "content": [
    // 1. Claude's decision to search
    {
      "type": "text",
      "text": "I'll search for when Claude Shannon was born."
    },
    // 2. The search query used
    {
      "type": "server_tool_use",
      "id": "srvtoolu_01WYG3ziw53XMcoyKL4XcZmE",
      "name": "web_search",
      "input": {
        "query": "claude shannon birth date"
      }
    },
    // 3. Search results
    {
      "type": "web_search_tool_result",
      "tool_use_id": "srvtoolu_01WYG3ziw53XMcoyKL4XcZmE",
      "content": [
        {
          "type": "web_search_result",
          "url": "https://en.wikipedia.org/wiki/Claude_Shannon",
          "title": "Claude Shannon - Wikipedia",
          "encrypted_content": "EqgfCioIARgBIiQ3YTAwMjY1Mi1mZjM5LTQ1NGUtODgxNC
          "page_age": "April 30, 2025"
        }
      ]
    },
    {
      "text": "Based on the search results, ",
      "type": "text"
    },
    // 4. Claude's response with citations
    {
      "text": "Claude Shannon was born on April 30, 1916, in Petoskey, Michiga
      "type": "text",
      "citations": [
        {
          "type": "web_search_result_location",
          "url": "https://en.wikipedia.org/wiki/Claude_Shannon",
          "title": "Claude Shannon - Wikipedia",
          "encrypted_index": "Eo8BCioIAhgBIiQyYjQ0OWJmZi1lNm..",
          "cited_text": "Claude Elwood Shannon (April 30, 1916 – February 24,
        }
      ]
    }
  ],
```

Ask Docs

```
      "input_tokens": 6039,
      "output_tokens": 931,
      "server_tool_use": {
        "web_search_requests": 1
      }
    },
    "stop_reason": "end_turn"
  }
```

Search results

Search results include:

- `url` : The URL of the source page
- `title` : The title of the source page
- `page_age` : When the site was last updated
- `encrypted_content` : Encrypted content that must be passed back in multi-turn conversations for citations

Citations

Citations are always enabled for web search, and each `web_search_result_location` includes:

- `url` : The URL of the cited source
- `title` : The title of the cited source
- `encrypted_index` : A reference that must be passed back for multi-turn conversations.
- `cited_text` : Up to 150 characters of the cited content

The web search citation fields `cited_text` , `title` , and `url` do not count towards input or output token usage.

> ⓘ When displaying API outputs directly to end users, citations must be included to the original source. If you are making modifications to API outputs, including by reprocessing and/or combining them with your own material before displaying them to end users, display citations as appropriate based on consultation with your legal team.

Ask Docs

When the web search tool encounters an error (such as hitting rate limits), the Claude API still returns a 200 (success) response. The error is represented within the response body using the following structure:

```
{
  "type": "web_search_tool_result",
  "tool_use_id": "servertoolu_a93jad",
  "content": {
    "type": "web_search_tool_result_error",
    "error_code": "max_uses_exceeded"
  }
}
```

These are the possible error codes:

- `too_many_requests` : Rate limit exceeded
- `invalid_input` : Invalid search query parameter
- `max_uses_exceeded` : Maximum web search tool uses exceeded
- `query_too_long` : Query exceeds maximum length
- `unavailable` : An internal error occurred

`pause_turn` stop reason

The response may include a `pause_turn` stop reason, which indicates that the API paused a long-running turn. You may provide the response back as-is in a subsequent request to let Claude continue its turn, or modify the content if you wish to interrupt the conversation.

Prompt caching

Web search works with prompt caching. To enable prompt caching, add at least one `cache_control` breakpoint in your request. The system will automatically cache up until the last `web_search_tool_result` block when executing the tool.

For multi-turn conversations, set a `cache_control` breakpoint on or after the last `web_search_tool_result` block to reuse cached content.

For example, to use prompt caching with web search for a multi-turn conversation:

Ask Docs

Ask Docs

```
client = anthropic.Anthropic()

# First request with web search and cache breakpoint
messages = [
    {
        "role": "user",
        "content": "What's the current weather in San Francisco today?"
    }
]

response1 = client.messages.create(
    model="claude-sonnet-4-5",
    max_tokens=1024,
    messages=messages,
    tools=[{
        "type": "web_search_20250305",
        "name": "web_search",
        "user_location": {
            "type": "approximate",
            "city": "San Francisco",
            "region": "California",
            "country": "US",
            "timezone": "America/Los_Angeles"
        }
    }]
)

# Add Claude's response to the conversation
messages.append({
    "role": "assistant",
    "content": response1.content
})

# Second request with cache breakpoint after the search results
messages.append({
    "role": "user",
    "content": "Should I expect rain later this week?",
    "cache_control": {"type": "ephemeral"}  # Cache up to this point
})

response2 = client.messages.create(
    model="claude-sonnet-4-5",
    max_tokens=1024,
    messages=messages,
    tools=[{
        "type": "web_search_20250305",
        "name": "web_search",
        "user_location": {
            "region": "California",
            "country": "US",
            "timezone": "America/Los_Angeles"
        }
    }]
)
# The second response will benefit from cached search results
# while still being able to perform new searches if needed
print(f"Cache read tokens: {response2.usage.get('cache_read_input_tokens', 0)}
```

Streaming

With streaming enabled, you'll receive search events as part of the stream. There will be a pause while the search executes:

```
event: message_start
data: {"type": "message_start", "message": {"id": "msg_abc123", "type": "messa

event: content_block_start
data: {"type": "content_block_start", "index": 0, "content_block": {"type": "t

// Claude's decision to search

event: content_block_start
data: {"type": "content_block_start", "index": 1, "content_block": {"type": "s

// Search query streamed
event: content_block_delta
data: {"type": "content_block_delta", "index": 1, "delta": {"type": "input_jso

// Pause while search executes

// Search results streamed
event: content_block_start
data: {"type": "content_block_start", "index": 2, "content_block": {"type": "w

// Claude's response with citations (omitted in this example)
```

Batch requests

Ask Docs

requests.

Usage and pricing

Web search usage is charged in addition to token usage:

```
"usage": {
  "input_tokens": 105,
  "output_tokens": 6039,
  "cache_read_input_tokens": 7123,
  "cache_creation_input_tokens": 7345,
  "server_tool_use": {
    "web_search_requests": 1
  }
}
```

Web search is available on the Claude API for **$10 per 1,000 searches**, plus standard token costs for search-generated content. Web search results retrieved throughout a conversation are counted as input tokens, in search iterations executed during a single turn and in subsequent conversation turns.

Each web search counts as one use, regardless of the number of results returned. If an error occurs during web search, the web search will not be billed.

Claude Docs

Solutions

AI agents

Code modernization

Coding

Customer support

Education

Financial services

Government

Learn

Blog

Catalog

Courses

Use cases

Connectors

Customer stories

Engineering at Anthropic

Ask Docs

Claude Docs

Partners

Amazon Bedrock

Google Cloud's Vertex AI

Service partners

Startups program

Company

Anthropic

Careers

Economic Futures

Research

News

Responsible Scaling Policy

Security and compliance

Transparency

Help and security

Availability

Status

Support

Discord

Terms and policies

Privacy policy

Responsible disclosure policy

Terms of service: Commercial

Terms of service: Consumer

Usage policy

Ask Docs