



## Scrape

# Batch Scrape

Batch scrape multiple URLs

## Batch scraping multiple URLs

You can now batch scrape multiple URLs at the same time. It takes the starting URLs and optional parameters as arguments. The params argument allows you to specify additional options for the batch scrape job, such as the output formats.

## How it works

It is very similar to how the `/crawl` endpoint works. It submits a batch scrape job and returns a job ID to check the status of the batch scrape.

The sdk provides 2 methods, synchronous and asynchronous. The synchronous method will return the results of the batch scrape job, while the asynchronous method will return a job ID that you can use to check the status of the batch scrape.

## Usage

Python

Node

cURL

```
from firecrawl import FirecrawlApp

app = FirecrawlApp(api_key="fc-YOUR_API_KEY")

# Scrape multiple websites:
batch_scrape_result = app.batch_scrape_urls(['firecrawl.dev', 'mendable.ai'], {'
```

```
print(batch_scrape_result)
```

 **Firecrawl**

# Or, you can use the asynchronous method:

```
batch_scrape_job = app.async_batch_scrape_urls(['firecrawl.dev', 'mendable.ai'],
Scrape > Batch Scrape
print(batch_scrape_job))
```

# (async) You can then use the job ID to check the status of the batch scrape:

```
batch_scrape_status = app.check_batch_scrape_status(batch_scrape_job['id'])
print(batch_scrape_status)
```

## Response

If you're using the sync methods from the SDKs, it will return the results of the batch scrape job. Otherwise, it will return a job ID that you can use to check the status of the batch scrape.

## Synchronous

Completed

```
{
  "status": "completed",
  "total": 36,
  "completed": 36,
  "creditsUsed": 36,
  "expiresAt": "2024-00-00T00:00:00.000Z",
  "next": "https://api.firecrawl.dev/v1/batch/scrape/123-456-789?skip=26",
  "data": [
    {
      "markdown": "[Firecrawl Docs home page

# Scrape multiple websites:
batch_scrape_result = app.batch_scrape_urls(
    ['https://docs.firecrawl.dev', 'https://docs.firecrawl.dev/sdks/overview'],
    {
        'formats': ['extract'],
    }
)

```

**Firecrawl**

```

        'extract': {
            'prompt': 'Extract the title and description from the page.',
            'schema': {
                'type': 'object',
                'properties': {
                    'title': {'type': 'string'},
                    'description': {'type': 'string'}
                },
                'required': ['title', 'description']
            }
        }
    }
)
print(batch_scrape_result)

# Or, you can use the asynchronous method:
batch_scrape_job = app.async_batch_scrape_urls(
    ['https://docs.firecrawl.dev', 'https://docs.firecrawl.dev/sdks/overview'],
    {
        'formats': ['extract'],
        'extract': {
            'prompt': 'Extract the title and description from the page.',
            'schema': {
                'type': 'object',
                'properties': {
                    'title': {'type': 'string'},
                    'description': {'type': 'string'}
                },
                'required': ['title', 'description']
            }
        }
    }
)
print(batch_scrape_job)

# (async) You can then use the job ID to check the status of the batch scrape:
batch_scrape_status = app.check_batch_scrape_status(batch_scrape_job['id'])
print(batch_scrape_status)

```

# Response

## Firecrawl

### Synchronous

Scrape > **Batch Scrape**


Completed


```
{
  "status": "completed",
  "total": 36,
  "completed": 36,
  "creditsUsed": 36,
  "expiresAt": "2024-00-00T00:00:00.000Z",
  "next": "https://api.firecrawl.dev/v1/batch/scrape/123-456-789?skip=26",
  "data": [
    {
      "extract": {
        "title": "Build a 'Chat with website' using Groq Llama 3 | Firecrawl",
        "description": "Learn how to use Firecrawl, Groq Llama 3, and Langchain to"
      }
    },
    ...
  ]
}
```

## Asynchronous

```
{
  "success": true,
  "id": "123-456-789",
  "url": "https://api.firecrawl.dev/v1/batch/scrape/123-456-789"
}
```



 Suggest edits

 Raise issue

Scrape > **Batch Scrape**  
< Scrape

LLM Extract >

Powered by Mintlify