

# Screaming Frog and DeepSeek Integration: Supercharge Your SEO

**Metehan Yesilyurt**

*AI Search & Information Retrieval Researcher · metehan.ai*

Published: January 28, 2025 · Canonical: <https://metehan.ai/blog/screaming-frog-deepseek-integration/>

I've created a simple way to integrate **DeepSeek** with Screaming Frog. And this setup connects over 300+ LLMs, including **Qwen-2.5**, the newest and most advanced model available today. With this integration, you can enhance your SEO analysis during website crawls.

You can find the full code and setup instructions in my [GitHub repository](#)<sup>[1]</sup>.

## Why Use DeepSeek R1 with Screaming Frog?

Screaming Frog is great for crawling and gathering data. Adding Deepseek R1 via OpenRouter lets you process that data using advanced AI models. You can analyze:

- **Page Titles:** Check if they align with user search intent.
- **Meta Descriptions:** Ensure they encourage clicks.
- **Headings or Body Text:** Review for relevance and structure.

This integration works seamlessly with OpenRouter.ai, giving you access to a wide range of AI models in a single tool.

## Features of the Integration

### Analyze Any Page Element

You can choose which parts of a page to analyze. Examples include:

- Page titles.
- Meta descriptions.
- Headings (H1, H2, etc.).
- Entire body text.

### Easy Customization

Change the AI prompt, models, or temperature settings directly in the code. You can also switch between OpenRouter's supported models.

## Simple Setup

No advanced programming skills are needed. Just copy and paste the JavaScript snippet into Screaming Frog.

## How to Set It Up

---

Follow these steps to integrate OpenRouter with Screaming Frog:

1. **Clone the Repository** Download the code from [GitHub](#)<sup>[1]</sup>.
2. **Add the Script in Screaming Frog**

- Open Screaming Frog. - Go to `Configuration > Crawl Config > Custom > Custom JavaScript`. - Paste the script from the repository.

1. **Add Your OpenRouter API Key** Replace `your_openrouter_api_key` in the script with your API key.
2. **Enable JavaScript Rendering** Go to `Configuration > Spider > Rendering` and select JavaScript rendering.
3. **Start Crawling** Run the crawl. The AI will process the selected page elements and return insights.

## Code Example

---

Here's the JavaScript snippet you'll use in Screaming Frog:

## TEXT

```
// Ask DeepSeek anything about the page - edited by https://metehan.ai
//
// Adjust the value of 'question' on line 26.
// Adjust the value of 'userContentList' on line 27, currently set to body text.
// Other examples such as page title, meta description, heading h1 or h2 are
// shown on line 32 onwards.
//
//
// This script demonstrates how JavaScript Snippets can communicate with
// APIs, in this case DeepSeek.
//
// This script also shows how the Spider will wait for JavaScript Promises to
// be fulfilled i.e. the fetch request to the ChatGPT API when fulfilled
// will return the data to the Spider.
//
// IMPORTANT:
// You will need to supply your API key below on line 25 which will be stored
// as part of your SEO Spider configuration in plain text. Also be mindful if
// sharing this script that you will be sharing your API key also unless you
// delete it before sharing.
//
// Also be aware of API limits when crawling large web sites with this snippet.
//

const OPENROUTER_API_KEY = 'sk-or-v1-xxx';
// Define the API endpoint
const apiUrl = 'https://openrouter.ai/api/v1/chat/completions';

// Define the question or prompt
const question = 'Analyze the following page content and rewrite title ';

// Select the page content to analyze
// Example: Page Title
const userContentList = [document.title];

// Function to send a request to the OpenRouter API
function chatGptRequest(userContent) {
  return fetch(apiUrl, {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${OPENROUTER_API_KEY}`,
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      model: 'deepseek/deepseek-rl-distill-llama-70b',
      messages: [
        {
          role: 'user',
          content: `${question} ${userContent}`
        }
      ],
      temperature: 0.7
    })
  })
  .then(response => {
    if (!response.ok) {
      return response.text().then(text => { throw new Error(text) });
    }
    return response.json();
  })
  .then(data => {
    return data.choices[0].message.content.trim();
  });
};
```

```
}

// Execute the function for each selected content and return the results to Screaming Frog
return Promise.all(userContentList.map(userContent => {
  return chatGptRequest(userContent);
})))
.then(data => seoSpider.data(data))
.catch(error => seoSpider.error(error));
```

## Benefits of This Setup

---

- **AI Variety:** Access over 300+ models, including **Qwen-2.5**.
- **Fast Analysis:** Process multiple page elements in real time.
- **Customizable:** Easily update prompts or choose different models.

## Final Thoughts

---

This integration helps you get better insights during your website crawls. It's easy to set up and works with Screaming Frog's Custom JavaScript feature. Try it out and explore the power of OpenRouter.ai.

[Check the GitHub repository for details<sup>\[1\]</sup>](#).

Screaming Frog plus DeepSeek is an AEO audit stack. AI citations in ChatGPT, Perplexity and Gemini follow patterns a local LLM can flag in crawl data. Answer engine optimization audits now run locally.

---

## REFERENCES & SOURCES

[1] [GitHub repository](https://github.com/metehan777/screaming-frog-deepseek) — <https://github.com/metehan777/screaming-frog-deepseek>