

Jokes API Documentation

Overview

The Jokes API provides a collection of humorous jokes in JSON format, perfect for displaying random jokes in web apps, widgets, or fun content sections.

API Files

- **Jokes:**

`jokes.json`

– Contains an array of objects with joke text and author name.

- **Extended Jokes:**

`jokes++.json`

– Structured similarly, but with potential for additional metadata in the future (e.g., categories, ratings).

File Structure

jokes.json contains:

- **author**

: Name of the person who told the joke.

- **part1**

: The setup of the joke.

- **part2**

: The punchline of the joke.

- **score**

: The score/rating of the joke.

- **mature**

: Boolean indicating whether the joke is suitable for mature audiences.

- **link**

: The URL link to the joke's Reddit post.

Using the API

Use JavaScript's

`fetch()`

method to retrieve jokes:

```
fetch('path/to/jokes.json')
  .then(response => response.json())
  .then(data => {
    console.log(data); // Logs array of joke objects
  });
```

Displaying Jokes (Example)

HTML container:

```
<div id="joke-box"></div>
```

JavaScript to populate a random joke:

```
document.addEventListener('DOMContentLoaded', () => {
  const jokeBox = document.getElementById('joke-box');

  fetch('path/to/jokes.json')
    .then(res => res.json())
    .then(data => {
      const random = data[Math.floor(Math.random() * data.length)];
      jokeBox.innerHTML = `
        <blockquote>${random.part1} ${random.part2}</blockquote>
        <footer>— ${random.author}</footer>
      `;
    });
});
```

Best Practices

- Randomize joke display to keep content fresh and fun.
- Preload jokes for performance-sensitive applications.
- Ensure jokes are appropriately categorized for different audience types.

Conclusion

The Jokes API is a simple and fun tool to integrate humor into digital experiences, ideal for social media widgets, entertainment apps, or websites focused on lighthearted content.