

Why We Use `config.js` Instead of `config.json` in Cucumber

What Are Tags in Cucumber?

- Tags are labels you can assign to your **feature files** or **scenarios** using @ symbol (e.g., `@smoke`, `@regression`).
 - They allow you to **run specific tests only**, rather than running all of them.
 - Helps in organizing and grouping test cases for different purposes like:
 - `@smoke` – for quick, basic tests
 - `@regression` – for a full suite
-

Why `config.json` Doesn't Work Well with Tags

- `config.json` is a **static JSON file** — good for simple settings like paths, timeouts, etc.
 - **Limitations:**
 - You **cannot pass tags** from the command line easily.
 - **No logic or variable usage** is possible (JSON is just data).
 - You cannot **change tags dynamically** using environment variables.
-

Why Use `config.js` Instead

- `config.js` is a **JavaScript file**, so it supports:
 - **Dynamic logic**
 - **Variables**
 - **Environment-based configurations**
 - You can control what tests to run using tags by passing them at runtime.
 - It makes your **test execution more flexible and powerful**.
-

Example: `config.js` with Tag Support

```
const commonConfig = {
  defaultTimeout: 60000,
  paths: ['./features/**/*.*.feature'],
};
```

```
const tags = process.env.TAGS || '@smoke'; // Default tag is @smoke

module.exports = {
  ...commonConfig,
  tags: tags,
};
```

Run tests with different tags like:

```
TAGS='@regression' npx cucumber-js
```

Quick Comparison: config.json VS config.js

Feature	config.json (static)	config.js (dynamic)
Supports Tags?	No	Yes
Can Use Variables?	No	Yes
Supports Logic?	No	Yes
Flexibility	Low	High
Best For	Simple settings	Tags & dynamic configs

Summary

- Use **tags** to run only selected test scenarios (like smoke or regression).
 - config.json is **simple** but **not flexible** for advanced setups like tagging.
 - config.js allows you to use **logic, variables, and dynamic config**.
 - If your project needs **tag-based execution**, switch to config.js.
-