



Importing JSON and CSS on Javascript

Two new proposals to import non-Javascript resources in Javascript modules are reaching stage 3 in the TC39 process, meaning that implementations are already available or will be available soon.

This post will take a first look at these features and how they can be used in a project.

JSON Modules

Json modules allow you to import JSON files and use them in your javascript modules.

Witht the following configuration file (config.json)

```
{
  "name": "My Applic" "My Application" "version": "v1.2"
}
```

We can import and use it from a sample JSON module as shown below.

```
import http from 'http';
import config from './config.json';
http
  './config.json'(req, res) => {
    res.write(`App name: ${config.name}\n`);
    res.write(`App version: ${config.version}`);
    res.end();
  })
  .listen(8080);
```

JSON modules are available in Node.js 17.1 with the experimental flag `--experimental-json-modules` and in Chrome 91 and above.

CSS Modules

CSS modules allow you to import CSS files from your Javascript modules.

When used together with constructable stylesheets, they can be used to import and install CSS stylesheets from CSS modules both in the main document and in web components' shadow roots.

```
import sheet from './styles.css' assert {  
  type: 'css'  
};  
  
'css'// Add the stylesheet to the host document  
document.adoptedStyleSheets = [sheet];  
  
// Add the stylesheet to a shadow root  
shadowRoot.adoptedStyleSheets = [sheet];
```

CSS modules are available in Chrome 93

Import assertions

Both JSON and CSS imports use [import assertions](#) to verify that the imported resource is of the expected type.

The import assertion looks like this for CSS modules:

```
assert {type: 'css'}
```

or like this for JSON modules:

```
assert {type: 'json'}
```

Without these assertions the import will fail with a type error.

Why these modules matter

CSS and JSON modules have some of the same benefits as JavaScript modules.

- **Deduplication:** The file will only be fetched, instantiated, and parsed once, regardless of how many times it's requested
- **Consistent order of evaluation:** when the importing JavaScript is running, it can rely on the content it imports having already been fetched and parsed
- **Security:** modules are fetched with CORS and use strict MIME-type checking