Working with Feature Policies in client-side Javascript

> ## Note:
>
> The `Feature-Policy` header has been renamed to `Permissions-Policy` and browsers will soon start implementing the change.
>
> The `document.featurePolicy` Javascript object will also change to `document.permissionsPolicy`.

The Feature Policies Javascript API allows our client-side code to query for what features are available and what is restricted by Feature Policies.

The Javascript API **doesn't replace the Feature-Policy header or the allow attribute of an iframe**. You sill need to set the header before this will work, or it will work with the default values.

The downside is that the current version of feature/permissions policy **doesn't support setting the policies in a meta tag, only in a header or the allow attribute of an iframe**. This makes the usage more complicated because we need to convince the server administrator (if we're not it) to implement them or add the header to our application's server code if we're creating our own.

We'll use the following Apache configuration. The configuration:

- Allows geolocation from the same origin and from `example.com`
- Allows `camera` and `microphone` from the same origin
- Forbids access to `usb`, `document.domain` and `document.write`, even from the same origin

```
<Location />
  Header setifempty Feature-Policy
    geolocation 'self' 'https://example/.com/;
    camera 'self';
    microphone 'self';
    usb 'none';
    document-domain 'none';
    clipboard-write 'self';
</Location>
```

We can query the `featurePolicy` object in Javascript to conditionally run code only if the feature is supported and allowed by the policy.

The following example will run geolocation code if the browser supports the feature and it's allowed by the feature policy

```
if (("geolocation" in navigator) &&
    (document.featurePolicy.allowsFeature('geolocation'))) {
  console.log('Geolocation supported and allowed');
} else {
  console.log('Geolocation not supported or not allowed');
}
```

While this example will test if the `navigator.clipboard.writeText` is supported and is allowed by the feature policy and run the code.

```
if((navigator.clipboard.writeText) &&
    (document.featurePolicy.allowsFeature('clipboard-write'))) {
  console.log('Feature is allowed');
} else {
  console.log('Feature not allowed');
}
```

Features of the JS API

The feature policy JS API in Chrome (since Chrome 74) provides the following

methods:

List of feature policies allowed by the page

```
document.featurePolicy.allowedFeatures();
// → list of features supported
```

**Check if a specific feature is supported##

```
document.featurePolicy.allowsFeature('geolocation');
```

Check if a feature is available for a specific domain

```
document.featurePolicy.allowsFeature('geolocation', 'https://another-exam
```

List of available feature policies allowed in the browser, regardless of whether they are in force

```
document.featurePolicy.features();
```

List origins (used throughout the page) that are allowed to use a given feature

```
document.featurePolicy.getAllowlistForFeature('geolocation');
```