



Plugins

Getting started

- Build a plugin
- Anatomy of a plugin
- Development workflow
- Mobile development
- Use React in your plugin
- Use Svelte in your plugin

Guides

User interface

Editor

Releasing

- Events
- Vault

Themes

Reference

- Developer policies
- Home

# Anatomy of a plugin

The [Plugin](#) class defines the lifecycle of a plugin and exposes the operations available to all plugins:

```
import { Plugin } from 'obsidian';

export default class ExamplePlugin extends Plugin {
  async onload() {
    // Configure resources needed by the plugin.
  }
  async onunload() {
    // Release any resources configured by the plugin.
  }
}
```

## Plugin lifecycle

[onload\(\)](#) runs whenever the user starts using the plugin in Obsidian. This is where you'll configure most of the plugin's capabilities.

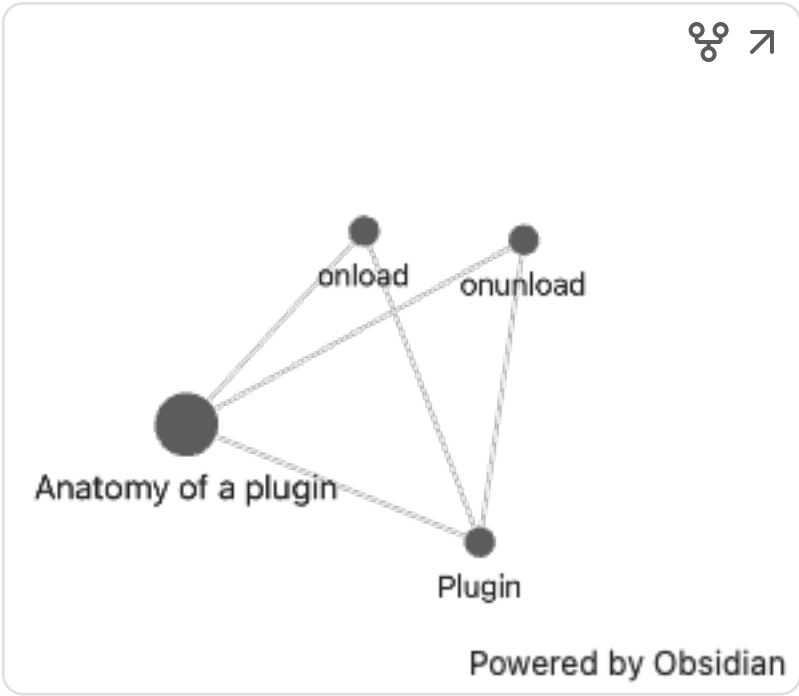
[onunload\(\)](#) runs when the plugin is disabled. Any resources that your plugin is using must be released here to avoid affecting the performance of Obsidian after your plugin has been disabled.

To better understand when these methods are called, you can print a message to the console whenever the plugin loads and unloads. The console is a valuable tool that lets developers monitor the status of their code.

To view the console:

1. Toggle the Developer Tools by pressing Ctrl+Shift+I in Windows and Linux, or Cmd-Option-I on macOS.
2. Click on the Console tab in the Developer Tools window.

INTERACTIVE GRAPH



ON THIS PAGE

Plugin lifecycle