Nexus
Theme Development Guide

The complete reference for building Nexus themes

nexusprism.org · MIT License

Covers Nexus v1.0.0 and later

About this guide

This is the complete reference for building Nexus themes. A Nexus theme changes the structural and typographic appearance of the forum through CSS — layout, fonts, spacing, component shape, animations. It can also include an optional JavaScript file for behavioural customisation. Colors are set by the forum admin and always apply on top of any theme; themes and colors are intentionally independent.

No Elixir compilation required. A theme is a GitHub release tarball containing a `theme.json` manifest, a `theme.css` stylesheet, and an optional `theme.js` script.

1. What a theme is and is not

What a theme is

A Nexus theme is a GitHub repository containing:

- A `theme.json` manifest declaring identity and optional admin-configurable settings.
- A `theme.css` stylesheet — the primary mechanism by which a theme changes the forum's appearance.
- An optional `theme.js` script for behavioural customisation.
- Optional `logo.webp` (200×200) and `banner.webp` (800×400) for the theme store.
- A README.

When a theme is activated for a mode (dark or light), Nexus injects `theme.css` as a `<link>` stylesheet and, if present, `theme.js` as a `<script>` tag into every page.

What a theme controls

- Layout — sidebar width, topbar height, feed column width, right panel position, grid structure
- Typography — custom fonts via `@import`, font sizes, line heights, letter spacing, font weights
- Spacing and shape — padding, margins, border radius, card dimensions
- Component styling — post cards, reply threads, composer, navigation, modals, dropdowns
- Animations and transitions — hover effects, focus states, page transitions
- Mode-specific rules — different styles for dark and light mode within a single stylesheet
- CSS variable overrides — override any of Nexus's design tokens for the duration the theme is active
- Behavioural customisation — optional JavaScript for dynamic effects, font injection, DOM manipulation

What a theme does not control

Colors are outside the scope of themes by design. The forum's accent color, surface tints, text colors, and border colors are set by the admin in Admin → Appearance and always apply after `theme.css` is loaded. This ensures that every theme looks correct regardless of which color scheme the admin has chosen.

■ Do not attempt to hardcode color values in `theme.css` for properties that should respond to the admin's color choices. Use CSS variables (`--ac`, `--bg`, `--s1`, etc.) so your theme remains correct under any admin color scheme.

2. Quick start

Create a repository with a `theme.json` and a `theme.css` at the root:

theme.json

```
{
  "manifest_version": 1,
  "name": "My Theme",
  "slug": "my-theme",
  "version": "1.0.0",
  "author": "your-github-handle"
}
```

theme.css

```
/* theme.css */
.sidebar-left {
  width: 260px;
  border-right: 2px solid var(--ac);
}
.post-card {
  border-radius: 4px;
}
```

Tag a release (`v1.0.0`), create a GitHub Release. In Admin → Themes → Install from URL, paste your repo URL and click Install. Activate it for dark or light mode. Done.

3. Project layout

```
my-theme/
■■■ theme.json        Required — the manifest
■■■ theme.css         Required — the stylesheet
■■■ theme.js          Optional — behavioural script
■■■ logo.webp         Optional — 200×200 theme store icon
■■■ banner.webp       Optional — 800×400 theme store banner
■■■ README.md
```

Where files end up after install

Source file	Where it goes
theme.json	Stored in the themes.manifest DB column
theme.css	/app/uploads/themes//theme.css — served at /uploads/themes//theme.css
theme.js	/app/uploads/themes//theme.js — served at /uploads/themes//theme.js
logo.webp	Referenced via logo_url in the manifest — must be a URL pointing to a hosted file
banner.webp	Referenced via banner_url in the manifest — must be a URL pointing to a hosted file

■ `logo.webp` and `banner.webp` are not automatically served from the tarball. Host them at a stable URL (e.g. GitHub Pages or a CDN) and reference them in `theme.json` via `logo_url` and `banner_url`.

4. The manifest — theme.json

4.1 Required fields

```
{
  "manifest_version": 1,
  "name": "My Theme",
  "slug": "my-theme",
  "version": "1.0.0"
}
```

Field	Constraint	Notes
manifest_version	Conventionally 1	Not validated by the loader — included by convention only.
name	Non-empty string	Shown in the theme store and admin panel.
slug	^[a-z0-9-]+$, unique	URL-safe identifier. Cannot be changed after install.
version	Non-empty string	Compared against GitHub release tag to detect updates. Semver recommended.

4.2 Metadata fields (all optional)

Field	Notes
description	One-sentence summary shown in the theme store card.
author	GitHub username or display name.
homepage	URL to the theme's GitHub repo or website.

Field	Notes
logo_url	Full URL to a 200×200 WebP or PNG icon.
banner_url	Full URL to an 800×400 WebP/PNG/JPEG hero image.
categories	Array of strings for category filter pills in the store.

4.3 CSS variable overrides — `variables` and `modes`

Themes can override Nexus's CSS design tokens directly from `theme.json` — no CSS required for variable-only changes. The loader validates all declared variable names against a whitelist at install time; unknown keys cause the install to fail with a specific error message.

Three keys control when overrides apply:

theme.json — variable override structure

```
{
  "variables": {
    "--av-radius": "4px",
    "--fs-body": "14px"
  },
  "modes": {
    "dark": {
      "variables": {
        "--bg": "#0d0d0d"
      }
    },
    "light": {
      "variables": {
        "--bg": "#fafafa"
      }
    }
  }
}
```

Key	When applied
variables	Applied in both dark and light mode
modes.dark.variables	Applied in dark mode only — merged on top of variables
modes.light.variables	Applied in light mode only — merged on top of variables

Valid CSS variables

Only the following variable names are accepted. Any other key causes the install to fail:

```
--bg  --s1  --s2  --s3
--b1  --b2  --b3
--t1  --t2  --t3  --t4  --t5
--ac  --ac-on  --ac-bg  --ac-border  --ac-text
--green  --red  --amber  --blue  --pink
--fs-ui  --fs-body  --fs-title  --fs-feed-title  --fs-content  --fs-code
--av-radius
```

■ Variable overrides run **after** all admin appearance settings, so they take precedence. However, since colors are the admin's domain, avoid overriding color variables (`--ac`, `--bg`, surface, text, and border vars) unless your theme explicitly requires a locked palette for a specific mode.

4.4 `settings`

Admin-configurable fields. Displayed on the theme card in the admin panel. Settings allow an admin to choose between layout variants, font options, spacing levels, or anything else the theme supports.

```
"settings": [
  {
    "key": "layout",
    "type": "select",
    "label": "Layout",
    "default": "standard",
    "options": [
      {"value": "standard", "label": "Standard"},
      {"value": "wide",     "label": "Wide sidebar"},
      {"value": "compact",  "label": "Compact"}
    ]
  },
  {
    "key": "font_size",
    "type": "number",
    "label": "Base font size (px)",
    "default": 15
  }
]
```

Setting types: `string`, `text`, `boolean`, `number`, `select`, `color`. Settings values are stored in the DB. They do not automatically inject as CSS variables.

5. theme.css — the complete reference

5.1 How it is injected

`theme.css` is injected as a `<link rel="stylesheet">` into every page, after Nexus's own stylesheets. It is added and removed as the user switches modes. Because it loads after Nexus's

stylesheets, your rules have full cascade priority. Use `!important` sparingly — prefer specificity.

5.2 Targeting modes

Nexus sets `data-theme="dark"` or `data-theme="light"` on `<html>`. Use this to write mode-specific rules within a single stylesheet:

```
/* Applies in both modes */
.post-card {
  border-radius: 4px;
}

/* Dark mode only */
[data-theme="dark"] .sidebar-left {
  border-right: 1px solid var(--b2);
}

/* Light mode only */
[data-theme="light"] .sidebar-left {
  border-right: 1px solid var(--b1);
  background: var(--s2);
}
```

5.3 Targeting a specific theme

Nexus sets `data-theme-slug` on `<html>` to the active theme's slug. Use this to scope rules to your theme only, preventing conflicts when multiple themes are installed:

```
/* Only applies when my-theme is active */
[data-theme-slug="my-theme"] .post-card {
  border-left: 3px solid var(--ac);
}

/* Combine with mode targeting */
[data-theme-slug="my-theme"][data-theme="light"] .sidebar-left {
  background: var(--s2);
}
```

5.4 Using CSS variables

Nexus exposes its full design token set as CSS variables on `:root`. Always use these rather than hardcoded values for anything that should respond to the admin's color scheme.

Variable	Purpose
--bg	Page background — outermost layer
--s1	First surface — cards, panels, post bodies
--s2	Second surface — inputs, dropdowns, nested cards

Variable	Purpose
--s3	Third surface — tooltips, popovers, overlays
--b1	Subtle border — dividers, list separators
--b2	Standard border — inputs, cards
--b3	Emphasis border — focused inputs, active states
--t1 – --t5	Text colors from primary (--t1) to faintest (--t5)
--ac	Accent color — buttons, active states, links
--ac-on	Text on accent backgrounds
--ac-bg	Faint accent-tinted background
--ac-border	Accent-tinted border
--ac-text	Accent-tinted text on neutral backgrounds
--green / --red / --amber / --blue / --pink	Semantic and decorative colors
--fs-ui / --fs-body / --fs-title / etc.	Typography scale
--av-radius	Avatar border-radius (e.g. 50% = circle, 0% = square)

■ The typography variables (`--fs-*`) and `--av-radius` are normally set by the admin's Appearance panel sliders. Overriding them in `theme.css` or via the `variables` manifest key will supersede the admin's values.

5.5 Importing fonts

```
@import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600&display=swap')
;

body,
.post-card,
.reply-card,
.composer-input {
  font-family: 'Inter', system-ui, sans-serif;
}
```

■ Font imports add a network request on every page load. Use `display=swap` to avoid render-blocking. Self-hosting fonts in the tarball is not supported — fonts bundled with the theme are not served.

5.6 Layout changes

Nexus uses a three-column layout: left sidebar, center feed, right panel. You can adjust widths, remove columns, or restructure the entire layout:

```
/* Wider sidebar */
.sidebar-left {
  width: 280px !important;
}

/* Narrower right panel */
.right-panel {
  width: 280px !important;
}

/* Remove the right panel entirely */
.right-panel {
  display: none !important;
}

/* Full-width feed when right panel is hidden */
.feed-col {
  max-width: 100% !important;
}
```

5.7 Component styling

```
/* Sharp-edged cards */
.post-card,
.reply-card {
  border-radius: 0 !important;
  border-left: 3px solid var(--ac);
  border-top: none;
  border-right: none;
  border-bottom: none;
}

/* Larger post titles in the feed */
.feed-post-title {
  font-size: calc(var(--fs-feed-title) + 2px) !important;
  font-weight: 700;
}
```

5.8 Complete example stylesheet

```
/* ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
   Wide Layout Theme
   All color values use CSS variables so admin
   accent/tint choices are fully respected.
   ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */

@import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap');

body {
  font-family: 'Inter', system-ui, sans-serif;
}

.sidebar-left { width: 270px !important; }

.post-card, .reply-card, .widget-card {
  border-radius: 6px !important;
}

.post-card { border-left: 3px solid var(--ac) !important; }

.feed-post-title { font-weight: 700 !important; }

/* Dark mode only */
[data-theme="dark"] .sidebar-left {
  border-right: 0.5px solid var(--b2);
}

/* Light mode only */
[data-theme="light"] .post-card {
  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.06);
}
[data-theme="light"] .sidebar-left {
  background: var(--s2);
}
```

6. theme.js — optional JavaScript

A theme can include a `theme.js` file for behavioural customisation — dynamic font loading, DOM manipulation, custom animations, or anything else that requires JavaScript. When present, Nexus injects it as a `<script>` tag alongside `theme.css`.

6.1 Cleanup convention

When the admin changes the active theme or the user switches mode, the old theme's script is removed and the new one is injected. To clean up any side effects your script creates (event listeners, DOM nodes, timers), assign a cleanup function to `window.__nexusThemeCleanup`:

theme.js — cleanup pattern

```
(function() {
  // Your theme setup
  const el = document.createElement('div');
  document.body.appendChild(el);

  const handler = () => { /* ... */ };
  window.addEventListener('scroll', handler);

  // Register cleanup — called before this script is removed
  window.__nexusThemeCleanup = function() {
    window.removeEventListener('scroll', handler);
    el.remove();
  };
})();
```

■ Nexus calls `window.__nexusThemeCleanup()` (if it exists) before removing the theme script. Always register a cleanup function if your script modifies the DOM or attaches global event listeners.

6.2 Script injection timing

`theme.js` is injected during `applyBranding` — once when the page loads and whenever the admin saves appearance settings. It is **not** re-injected on user-initiated mode switches (dark/light toggle); mode switches only re-apply CSS variables and swap `theme.css`. Design your script accordingly.

7. theme.json manifest validation

The theme loader validates `theme.json` at install time. Common errors:

Error message	Cause and fix
theme.json not found in theme package	No theme.json at the root of the tarball. Check that your files are at the root inside the release tarball's top-level directory.
name is required	name field is missing or empty.
slug must be lowercase alphanumeric with hyphens	slug contains uppercase, spaces, or special characters. Use only a-z, 0-9, and -.
version is required	version field is missing or empty.
Unknown CSS variables: --foo, --bar	variables or modes.*.variables contains keys not in the valid variable whitelist. Check spelling and refer to Section 4.3.
variables must be an object	The variables key exists but is not a JSON object.
modes.dark.variables must be an object	The modes.dark.variables key exists but is not a JSON object.
No release found	No GitHub Release exists for the repo. Create a release tagged v1.0.0 matching your version field.

8. Publishing a theme

8.1 GitHub releases

- Create a public GitHub repository.
- Place `theme.json` and `theme.css` at the root. Add `theme.js` if needed.
- Tag a release: `git tag v1.0.0 && git push origin v1.0.0`
- Create a GitHub Release from that tag. Nexus uses GitHub's auto-generated source tarball — no custom asset needed.

To update: bump version in `theme.json`, push, create a new tag and release. Installed themes show an Update button when the latest release tag differs from the installed version.

■ Nexus strips a leading 'v' from the release tag before comparing. Tag `v1.0.1` matches `version: "1.0.1"`.

8.2 Tarball caching

Every successful theme download is cached to the server's uploads directory. On subsequent Nexus restarts, the theme loads from the local cache — no GitHub request needed. This means:

- Boot time is not affected by GitHub availability after the first install.
- GitHub is only contacted at install time and update time.
- The cache is cleared when the theme is uninstalled.

8.3 Listing in the theme store

Themes are listed in the same registry as extensions at `github.com/ResofireV2/nexus-extensions`. Add an entry with `"type": "theme"` to `registry.json` via pull request:

```
{
  "type": "theme",
  "name": "My Theme",
  "slug": "my-theme",
  "description": "A clean wide-layout theme.",
  "author": "yourhandle",
  "homepage": "https://github.com/yourhandle/nexus-my-theme",
  "github_url": "https://github.com/yourhandle/nexus-my-theme",
  "logo_url": "https://yourhandle.github.io/nexus-my-theme/logo.webp",
  "banner_url": "https://yourhandle.github.io/nexus-my-theme/banner.webp",
  "categories": ["minimal", "wide"],
  "version": "1.0.0"
}
```

9. Checklist before publishing

Check	Why
All color values use CSS variables, not hardcoded hex	Hardcoded colors will look wrong under any admin color scheme except the one you tested with.
theme.css tested in both dark and light mode	Even if your theme targets one mode, admins may assign it to both.
Font imports use display=swap	Without it, text is invisible until the font loads.
theme.json slug is globally unique and lowercase	Slug collisions with other installed themes cause install failure.
version in theme.json matches the GitHub release tag	Mismatch means the Update button never appears.
variables keys are all in the valid whitelist	Unknown keys cause install to fail with a validation error.
theme.js registers window.__nexusThemeCleanup if it modifies the DOM	Without cleanup, effects persist after the theme is deactivated.
A GitHub Release exists before sharing the install URL	Install fails with 'No release found' if no release has been published.
Tested by installing from URL before listing in the store	Catches manifest validation errors before other users encounter them.

Appendix — Manifest quick reference

Field	Req	Purpose
manifest_version	—	Conventionally 1; not validated by the loader
name	✓	Display name
slug	✓	Machine ID, URL prefix — immutable after install
version	✓	Theme version (semver recommended)
description		One-sentence summary
author		Author name or handle
homepage		Canonical URL
logo_url		200×200 icon (hosted URL)
banner_url		800×400 hero image (hosted URL)

Field	Req	Purpose
categories		Array of category strings for store filter pills
variables		CSS var overrides — applied in both modes
modes.dark.variables		CSS var overrides — dark mode only
modes.light.variables		CSS var overrides — light mode only
settings		Admin-configurable settings schema

Appendix — Rules to follow

Always

- Use CSS variables for all colors, backgrounds, borders, and text — never hardcode hex values for anything that should respond to admin color settings.
- Use `var(--fs-body)` or `var(--fs-ui)` for font sizes in UI chrome.
- Use `border: 0.5px solid var(--b1)` (or `--b2`, `--b3`) for borders — not 1px and not an arbitrary color.
- Test in both dark and light mode before publishing your theme.
- Register `window.__nexusThemeCleanup` in `theme.js` if your script modifies the DOM or attaches global listeners.
- Use `[data-theme-slug="your-slug"]` to scope rules to your theme when specificity conflicts are possible.

Never

- Never hardcode accent colors (e.g. `#7c3aed`) — use `var(--ac)` and derived accent variables.
- Never use 1px solid borders — Nexus uses 0.5px solid throughout.
- Never use `font-family: sans-serif` without the Inter fallback chain: `'Inter', system-ui, sans-serif`.
- Never set `position: fixed` in a right sidebar widget — the sidebar scrolls independently.
- Never use `z-index` values above 9000 — Nexus's overlays, modals, and toasts use this range.
- Never declare CSS variable overrides outside the valid whitelist — unknown keys cause install failure.