

@design.estate/dees-element

a custom element class extending lit element class

- [readme.md for @design.estate/dees-element](#)
- [changelog.md for @design.estate/dees-element](#)

readme.md for @design.estate/dees- element

A powerful custom element base class that extends Lit's `LitElement` with integrated theming, responsive CSS utilities, RxJS-powered directives, and DOM tooling — so you can build web components that look great and stay reactive out of the box.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Install

```
npm install @design.estate/dees-element  
# or  
pnpm install @design.estate/dees-element
```

This package ships as ESM and is written in TypeScript. Make sure your project targets ES2022+ with a modern module resolution strategy (e.g. `NodeNext`).

Usage

Everything you need is exported from the main entry point:

```
import {
  DeesElement,
  customElement,
  property,
  state,
  html,
  css,
  cssManager,
  directives,
} from '@design.estate/dees-element';
```

📄 Creating a Custom Element

Extend `DeesElement` and apply the `@customElement` decorator:

```
import { DeesElement, customElement, html, css, cssManager } from '@design.estate/dees-element';

@customElement('my-button')
class MyButton extends DeesElement {
  static styles = [
    cssManager.defaultStyles,
    css`
      .btn {
        padding: 8px 16px;
        border-radius: 4px;
        background: ${cssManager.bdTheme('#0060df', '#3a8fff')};
        color: ${cssManager.bdTheme('#fff', '#fff')};
        border: none;
        cursor: pointer;
      }
    `,
  ];

  render() {
    return html`<button class="btn"><slot></slot></button>`;
  }
}
```

That single `bdTheme()` call generates a CSS variable that automatically flips between the bright and dark values when the user's theme changes — no manual toggling needed.

☐☐ Theme Management with `cssManager`

The singleton `cssManager` is the central hub for theming and responsive layout:

Method	Purpose
<code>cssManager.defaultStyles</code>	Base styles for consistent element rendering
<code>cssManager.bdTheme(bright, dark)</code>	Returns a <code>CSSResult</code> that auto-switches between bright/dark values
<code>cssManager.cssForDesktop(css, this?)</code>	Breakpoint for desktop; pass <code>this</code> for component-scoped
<code>cssManager.cssForNotebook(css, this?)</code>	Breakpoint for notebook; pass <code>this</code> for component-scoped
<code>cssManager.cssForTablet(css, this?)</code>	Breakpoint for tablet; pass <code>this</code> for component-scoped
<code>cssManager.cssForPhablet(css, this?)</code>	Breakpoint for phablet; pass <code>this</code> for component-scoped
<code>cssManager.cssForPhone(css, this?)</code>	Breakpoint for phone; pass <code>this</code> for component-scoped
<code>cssManager.cssForConstraint({ maxWidth, minWidth })</code>	Custom viewport-level constraint (curried)
<code>cssManager.cssGridColumnColumns(cols, gap)</code>	Generates CSS grid column widths

Example — responsive + themed styles:

```
@customElement('my-card')
class MyCard extends DeesElement {
  static styles = [
    cssManager.defaultStyles,
    css`
      :host {
        display: block;
        padding: 16px;
        background: ${cssManager.bdTheme('#ffffff', '#1e1e1e')};
        color: ${cssManager.bdTheme('#111', '#eee')};
        border-radius: 8px;
      }
    `,
    cssManager.cssForPhone(css`
      :host { padding: 8px; }
    `),
  ];
};
```

```
render() {  
  return html`<slot></slot>`;  
}  
}
```

☐☐ Container-Responsive Components

For components that need to respond to their **own width** (not the viewport), use the `@containerResponsive()` decorator and pass `this` as the second argument to `cssManager.cssFor*`:

```
import {  
  DeesElement, customElement, html, css, cssManager,  
  containerResponsive,  
} from '@design.estate/dees-element';  
  
@containerResponsive()  
@customElement('my-stats-grid')  
class MyStatsGrid extends DeesElement {  
  static styles = [  
    cssManager.defaultStyles,  
    css`.grid { display: grid; grid-template-columns: repeat(3, 1fr); gap: 16px; }`,  
  
    // Component-level: when THIS element is narrower than tablet width  
    cssManager.cssForTablet(css`  
      .grid { grid-template-columns: repeat(2, 1fr); }  
    `, this),  
  
    // Viewport-level: when the browser window is phone-sized  
    cssManager.cssForPhone(css`  
      .grid { grid-template-columns: 1fr; }  
    `),  
  
    // Component-level with custom width constraint  
    this.cssForConstraint({ maxWidth: 500 })(css`  
      .grid { gap: 8px; }  
    `),  
  ];  
}
```

```

render() {
  return html`<div class="grid"><slot></slot></div>`;
}
}

```

How it works:

API	Scope	Generated CSS
<code>cssManager.cssForPhablet(css)</code>	Viewport	<code>@media + @container</code> <code>wccToolsViewport</code>
<code>cssManager.cssForPhablet(css, this)</code>	Component	<code>@container <tag-name> only</code>
<code>cssManager.cssForConstraint({maxWidth: 800})(css)</code>	Viewport	<code>@media + @container</code> <code>wccToolsViewport</code>
<code>this.cssForConstraint({maxWidth: 500})(css)</code>	Component	<code>@container <tag-name> only</code>
<code>@containerResponsive()</code>	Decorator	Sets <code>container-type: inline-size +</code> <code>container-name on :host</code>

The `@containerResponsive()` decorator is required for component-scoped queries — it establishes the CSS containment context on `:host`.

⚡ Reactive Properties & State

Use the standard Lit decorators, re-exported for convenience:

```

import { DeesElement, customElement, property, state, html } from '@design.estate/dees-element';

@customElement('my-counter')
class MyCounter extends DeesElement {
  @property({ type: String })
  accessor label = 'Count';

  @state()
  accessor count = 0;

  render() {
    return html`
      <button @click=${() => this.count++}>
        ${this.label}: ${this.count}
      </button>
    `;
  }
}

```

```
`;  
}  
}
```

“ **Note:** This library uses the TC39 standard decorators with the `accessor` keyword for decorated class properties.

☐☐ Theme Change Callbacks

`DeesElement` tracks the current theme via the `goBright` property and exposes an optional `themeChanged` callback:

```
@customElement('theme-aware')  
class ThemeAware extends DeesElement {  
  protected themeChanged(goBright: boolean) {  
    console.log(goBright ? 'Switched to bright' : 'Switched to dark');  
  }  
  
  render() {  
    return html`<p>Current theme: ${this.goBright ? 'bright' : 'dark'}</p>`;  
  }  
}
```

☐☐ Lifecycle Helpers

`DeesElement` adds lifecycle utilities on top of `LitElement`:

```
@customElement('my-widget')  
class MyWidget extends DeesElement {  
  constructor() {  
    super();  
  
    // Runs once after the element is connected to the DOM  
    this.registerStartupFunction(async () => {  
      console.log('Widget connected!');  
    });  
  }  
}
```

```
// Runs when the element is disconnected – perfect for cleanup
this.registerGarbageFunction(() => {
  console.log('Widget removed');
});
}

render() {
  return html`<p>Hello World</p>`;
}
}
```

Additionally, `this.elementDomReady` is a promise that resolves after `firstUpdated`, which is handy when you need to wait for the initial render:

```
await this.elementDomReady;
// The element's shadow DOM is now fully rendered
```

☐ Directives

The `directives` namespace includes powerful template helpers, accessible via `directives.*`:

`resolve` — Render a Promise

```
import { html, directives } from '@design.estate/dees-element';

render() {
  return html`${directives.resolve(this.fetchData())}`;
}
```

`resolveExec` — Resolve a lazy async function

```
render() {
  return html`${directives.resolveExec(() => this.loadContent())}`;
}
```

`subscribe` — Render an RxJS Observable

```
import { html, directives } from '@design.estate/dees-element';
```

```
render() {
  return html`<span>${directives.subscribe(this.count)}</span>`;
}
```

subscribeWithTemplate — Observable + template transform

```
render() {
  return html`
    ${directives.subscribeWithTemplate(
      this.items$,
      (items) => html`<ul>${items.map(i => html`<li>${i}</li>`)}`</ul>`
    )}
  `;
}
```

Re-exported Lit directives

The directives namespace also re-exports these commonly used Lit directives:

- `until` — render a placeholder while a promise resolves
- `asyncAppend` — append values from an async iterable
- `keyed` — force re-creation of a template when a key changes
- `repeat` — efficiently render lists with identity tracking

📖 Full Export Reference

Export	Description
<code>DeesElement</code>	Base class for custom elements
<code>CssManager</code>	CSS/theme management class
<code>cssManager</code>	Singleton <code>CssManager</code> instance
<code>customElement</code>	Class decorator to register elements
<code>property</code>	Reactive property decorator
<code>state</code>	Internal state decorator
<code>query</code> , <code>queryAll</code> , <code>queryAsync</code>	Shadow DOM query decorators
<code>html</code>	Lit html template tag
<code>css</code>	Lit css template tag
<code>unsafeCSS</code>	Create <code>CSSResult</code> from a string

Export	Description
<code>unsafeHTML</code>	Render raw HTML in templates
<code>render</code>	Lit render function
<code>static</code> / <code>unsafeStatic</code>	Static html template helpers
<code>containerResponsive</code>	Decorator that adds CSS containment to <code>:host</code>
<code>domtools</code>	DOM tooling utilities
<code>directives</code>	All directives (resolve, subscribe, etc.)
<code>rxjs</code> (type)	RxJS type re-export

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [LICENSE](#) file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture

Capital GmbH of any derivative works.

changelog.md for @design.estate/dees- element

2026-03-27 - 2.2.4 - fix(build)

migrate package config to .smartconfig and align build tooling

- replace npmextra.json with .smartconfig.json and include tsbundle bundle configuration
- update build script and package files list to use the new smartconfig-based setup
- bump build-related dependencies and add Node types to tsconfig
- add definite assignment assertions for element properties to satisfy TypeScript checks

2026-03-12 - 2.2.3 - fix(repo)

no changes to commit

2026-03-11 - 2.2.2 - fix(decorators)

patch container-responsive styles to fix stale container queries and add customElement wrapper that sets .is and warns on class/tag mismatch

- containerResponsive now derives the kebab name and replaces stale '@container ' occurrences inside CSSResult cssText using unsafeCSS to correct container queries produced before decorators ran
- containerResponsive appends containerContextStyles to the component styles while preserving/processing existing static styles
- Added a customElement decorator wrapper that assigns .is on the class, logs a warning if the class name kebab-cases to a different tag name, and delegates to Lit's customElement

- `ts/index.ts` now re-exports `customElement` from the local decorator so the new behavior is used project-wide

2026-03-11 - 2.2.1 - fix(dees-element)

rename `cssForCustom` to `cssForConstraint`, remove `DeesElement` static `cssFor*` helpers, and add optional `elementClass` parameter to `cssManager` breakpoint helpers

- Rename: `cssManager.cssForCustom` -> `cssManager.cssForConstraint` and `DeesElement.cssForCustom` -> `DeesElement.cssForConstraint`; underlying `domtools` calls updated to `cssForConstraint/cssForConstraintContainer`.
- Breaking: Removed static helpers `DeesElement.cssForDesktop`, `cssForNotebook`, `cssForTablet`, `cssForPhablet`, `cssForPhone` — consumers must now use `cssManager.cssFor*(css, this)` or pass an element class to get component-scoped container rules.
- API change: `cssManager.cssForDesktop|cssForNotebook|cssForTablet|cssForPhablet|cssForPhone` now accept an optional `elementClass` (or pass `this`) and a helper `getContainerNameFromClass` was added to generate container names.
- Docs: `readme.md` updated to show new signatures, guidance to pass `this` for component-scoped constraints, and renaming of "custom" to "constraint".
- Dependency: bumped `@design.estate/dees-domtools` from `^2.4.0` to `^2.5.1`.

2026-03-11 - 2.2.0 - feat(dees-element)

add container-responsive APIs (`containerResponsive` decorator, `DeesElement` static `cssFor*` container helpers, and `cssManager` `cssForCustom`) and update docs

- Exported new `containerResponsive` decorator (`ts/decorators.containerresponsive.ts` and exported from `ts/index.ts`)
- Added component-level static helpers on `DeesElement`: `cssForDesktop`, `cssForNotebook`, `cssForTablet`, `cssForPhablet`, `cssForPhone`, `cssForCustom`
- Added `cssManager.cssForCustom` to provide curried custom breakpoint helper
- Updated `readme` with `Container-Responsive Components` section and usage examples
- Bumped dependency `@design.estate/dees-domtools` from `^2.3.8` to `^2.4.0`

- Non-breaking new functionality — recommend minor version bump

2026-01-27 - 2.1.6 - fix(docs, deps, tests)

update README with expanded usage docs and examples; bump devDependencies and dees-domtools; fix test import path

- Bumped devDependencies: @git.zone/tsbuild -> ^4.1.2, @git.zone/tsbundle -> ^2.8.3, @git.zone/tstest -> ^3.1.8, @types/node -> ^25.0.10
- Bumped dependency @design.estate/dees-domtools -> ^2.3.8
- README rewritten/expanded: more complete usage examples, theme management (cssManager) docs, directives reference, lifecycle helpers, install and issue reporting guidance
- Test fix: updated import to use @git.zone/tstest/tapbundle instead of @push.rocks/tapbundle

2026-01-04 - 2.1.5 - fix(build)

bump @design.estate/dees-domtools to ^2.3.7 and remove experimentalDecorators and useDefineForClassFields from tsconfig.json

- Bumped @design.estate/dees-domtools from ^2.3.6 to ^2.3.7.
- Removed "experimentalDecorators" and "useDefineForClassFields" compiler options from tsconfig.json.

2026-01-04 - 2.1.4 - fix(build)

update dev and runtime dependencies, export additional lit directives, switch test export to default, convert class properties to accessors, and update npmextra configuration

- Bump devDependencies: @git.zone/tsbuild -> ^4.0.2, @git.zone/tsbundle -> ^2.6.3, @git.zone/tstest -> ^3.1.4, @types/node -> ^25.0.3
- Bump runtime deps: @design.estate/dees-domtools -> ^2.3.6, lit -> ^3.3.2
- Change test script to run tstest in verbose mode and export default tap.start() in test entry

- Convert DeesElement properties to use 'accessor' syntax and make domtools non-optional in type
- Add exports for lit directives keyed and repeat in directives index
- Update packageManager to pnpm@10.27.0 and adjust npmextra.json: rename keys, add release registries and accessLevel

2025-11-16 - 2.1.3 - fix(CssManager)

Make CssManager a singleton and export the shared instance via getSingleton; update tests and dependencies

- Convert CssManager to a singleton by adding a private static instance and a public static getSingleton() method.
- Use CssManager.getSingleton() for the exported cssManager in ts/index.ts to ensure a single shared instance across the app.
- Add a Chromium-focused test (test.chromium.ts) and remove the browser-specific test file.
- Bump devDependencies (@git.zone/tsbuild, @git.zone/tsbundle, @git.zone/tstest) and update lit to ^3.3.1.
- Add readme.hints.md documenting the CssManager singleton pattern and supported access patterns.

2025-07-06 - 2.1.2 - fix(build)

Update build script in package.json to include 'tsfolders' in tsbuild command

- Changed build script from 'tsbuild --web --allowimplicitany && tsbundle npm' to 'tsbuild tsfolders --web --allowimplicitany && tsbundle npm'

2025-07-06 - 2.1.1 - fix(documentation)

Refine project documentation and metadata for clarity

- Update readme examples to better illustrate custom element usage
- Clarify CssManager theming and API usage in documentation
- Ensure package.json and commitinfo reflect accurate project details

2025-07-06 - 2.1.0 - feat(DeesElement)

Add invocation of the themeChanged hook in connectedCallback

- Now calls themeChanged (if defined) when the theme changes, enabling custom handlers for theme switches
- Improves lifecycle management by allowing extensions to react to bright/dark mode changes

2025-06-20 - 2.0.44 - fix(ci)

Remove obsolete GitLab CI configuration

- Deleted the .gitlab-ci.yml file to remove outdated CI configuration from the repository.

2025-06-20 - 2.0.43 - fix(dependencies)

Bump build and runtime dependencies to newer versions for improved tooling and compatibility.

- Bumped @git.zone/tsbuild from ^2.3.2 to ^2.6.4
- Bumped @git.zone/tsbundle from ^2.2.5 to ^2.4.0
- Bumped @git.zone/tstest from ^1.0.96 to ^2.3.1
- Bumped @push.rocks/tapbundle from ^5.6.3 to ^6.0.3
- Bumped @design.estate/dees-domtools from ^2.3.2 to ^2.3.3
- Bumped @push.rocks/smartrx from ^3.0.7 to ^3.0.10

2025-04-18 - 2.0.42 - fix(directives)

Add explicit type annotations to subscribeWithTemplate directive export

- Imported DirectiveResult type for better typing
- Defined SubscribeWithTemplateFn signature to ensure proper type inference
- Used type assertion with 'as SubscribeWithTemplateFn' to improve type safety

2025-04-18 - 2.0.41 - fix(directives)

Refactor export statements in directives index for consistency

- Changed individual export of 'resolve' and 'subscribe' to wildcard exports in ts/directives/index.ts
- Simplified module export structure without altering functionality

2025-04-18 - 2.0.40 - fix(dees- element)

Refactor project structure and update dependency versions. Internal modules (e.g. dees-element classes and directives) have been reorganized and deprecated paths removed, and package.json now includes an updated packageManager field.

- Updated versions for @git.zone/tsbuild, tsbundle, ttest, tapbundle, and lit.
- Changed dependency @design.estate/dees-domtools from ^2.0.61 to ^2.3.2.
- Reorganized internal file structure: moved code from dees-element.classes.* to classes.* and re-exported via index.
- Added packageManager field in package.json for pnpm configuration.

2024-10-04 - 2.0.39 - fix(core)

Update dependency version for @design.estate/dees-domtools

- Bumped @design.estate/dees-domtools dependency to version ^2.0.61 to include latest patches and improvements.

2024-10-02 - 2.0.38 - fix(dependencies)

Bump @design.estate/dees-domtools to 2.0.60

- Updated @design.estate/dees-domtools from 2.0.59 to 2.0.60

2024-10-02 - 2.0.37 - fix(dependencies)

Update dependencies to latest versions.

- Updated @git.zone/tsbuild from ^2.1.82 to ^2.1.84
- Updated @types/node from ^20.14.9 to ^22.7.4
- Updated @push.rocks/tapbundle from ^5.0.23 to ^5.3.0
- Updated @design.estate/dees-domtools from ^2.0.57 to ^2.0.59
- Updated lit from ^3.1.4 to ^3.2.0

2024-07-01 - 2.0.36 - fix(core)

Ensure documentation completeness and code quality improvements

- Improved the README.md file to provide more detailed usage instructions and examples.
- Refactored code for better readability and performance.
- Updated npmextra.json to ensure accurate project metadata and configurations.

2024-07-01 - 2.0.35 - fix(dependencies)

Update dependency versions and development dependencies

- Update @git.zone/tsbuild to version ^2.1.82
- Update @git.zone/tstest to version ^1.0.90
- Update @push.rocks/tapbundle to version ^5.0.23
- Update @types/node to version ^20.14.9
- Update lit to version ^3.1.4

2024-04-20 - 2.0.34 - Documentation

Updated project documentation.

- Refreshed the documentation to include recent changes and improvements.

2023-10-31 - 2.0.33 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-10-26 - 2.0.32 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-10-23 - 2.0.31 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-10-23 - 2.0.30 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-09-17 - 2.0.29 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-09-17 - 2.0.28 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-09-04 - 2.0.27 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-09-04 - 2.0.26 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-08-07 - 2.0.25 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-08-07 - 2.0.24 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-08-07 - 2.0.23 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-03-29 - 2.0.22 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-03-29 - 2.0.21 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-03-26 - 2.0.20 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2023-03-15 - 2.0.19 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-12-31 - 2.0.18 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-12-31 - 2.0.17 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-05-21 - 2.0.16 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-05-12 - 2.0.15 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-05-02 - 2.0.14 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-05-01 - 2.0.13 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-04-22 - 2.0.12 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-04-21 - 2.0.11 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-04-19 - 2.0.10 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-04-19 - 2.0.9 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-04-18 - 2.0.8 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-04-15 - 2.0.7 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-29 - 2.0.6 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-25 - 2.0.5 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-24 - 2.0.4 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-23 - 2.0.3 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-16 - 2.0.2 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-16 - 2.0.1 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-16 - 2.0.0 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-03-16 - 1.0.36 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-01-07 - 1.0.35 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-01-06 - 1.0.34 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-01-06 - 1.0.33 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2022-01-06 - 1.0.32 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-12-14 - 1.0.31 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-12-13 - 1.0.30 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-12-10 - 1.0.29 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-12-10 - 1.0.28 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-11-27 - 1.0.27 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-27 - 1.0.26 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-16 - 1.0.25 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-16 - 1.0.24 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-13 - 1.0.23 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-08 - 1.0.22 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-08 - 1.0.21 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-09-08 - 1.0.20 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-29 - 1.0.19 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-29 - 1.0.18 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-29 - 1.0.17 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-28 - 1.0.16 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-28 - 1.0.15 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-28 - 1.0.14 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-27 - 1.0.13 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-27 - 1.0.12 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.

2021-03-27 - 1.0.11 - Core

Minor core fixes and optimizations.

- Updated core functionalities to improve performance.