

readme.md for @ecobridge.xyz/catalog

A sophisticated web component library for building desktop-like application interfaces with modern web technologies. Built on `@design.estate/dees-catalog`, this catalog provides specialized components for the ecobridge ecosystem — featuring complete app launcher experiences, authentication flows, and system-level UI components.

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.

▣ Features

- **▣ Desktop-like UI** — Full application launcher with top bar, status bar, and app grid
- **▣ Flexible Authentication** — PIN, password, and QR code login methods
- **▣ Dark/Light Mode** — Theme-aware components with automatic switching
- **▣ Virtual Keyboard** — Touch-friendly on-screen keyboard
- **▣ System Menus** — WiFi, battery, sound, and power controls
- **▣ Responsive Design** — Works on desktop and mobile
- **▣ Consistent Design System** — Unified theming with CSS custom properties
- **↩ TypeScript First** — Full type definitions for excellent DX

▣ Installation

```
pnpm add @ecobridge.xyz/catalog
```

```
npm install @ecobridge.xyz/catalog
```

📄 Quick Start

```
import { EcoApplauncher } from '@ecobridge.xyz/catalog';

const launcher = document.createElement('eco-applauncher');
launcher.mode = 'home';
launcher.apps = [
  { name: 'Settings', icon: 'lucide:settings', action: () => console.log('Settings') },
  { name: 'Files', icon: 'lucide:folder', action: () => console.log('Files') },
  { name: 'Terminal', icon: 'lucide:terminal', action: () => console.log('Terminal') },
];
document.body.appendChild(launcher);
```

📄 Components

EcoApplauncher

The main application launcher — a complete desktop-like interface with authentication, app grid, and system controls.

```
import { EcoApplauncher, type IAppIcon, type ILoginConfig } from '@ecobridge.xyz/catalog';

const launcher = document.createElement('eco-applauncher');

// Configure login
launcher.loginConfig = {
  allowedMethods: ['pin', 'password', 'qr'],
  pinLength: 6,
  welcomeMessage: 'Welcome Back',
  subtitle: 'Enter your credentials to continue',
  logoUrl: '/assets/logo.svg',
};
```

```
// Configure apps
launcher.apps = [
  {
    name: 'Dashboard',
    icon: 'lucide:layoutDashboard',
    action: () => openDashboard(),
  },
  {
    name: 'Settings',
    icon: 'lucide:settings',
    view: html`<eco-view-settings></eco-view-settings>`, // Embed view directly
  },
];

// Configure status bar
launcher.statusConfig = {
  showTime: true,
  showNetwork: true,
  showBattery: true,
  showSound: true,
  showKeyboard: true,
};

// Configure top bar
launcher.topBarConfig = {
  showSearch: true,
  showDate: true,
  showNotifications: true,
  showUser: true,
};

// Set status values
launcher.batteryLevel = 85;
launcher.networkStatus = 'online';
launcher.soundLevel = 70;
launcher.userName = 'John Doe';
launcher.notificationCount = 3;

// Handle events
launcher.addEventListener('login-attempt', (e) => {
```

```

const { method, value } = e.detail;
// Validate credentials
if (validateCredentials(method, value)) {
  launcher.setLoginResult(true);
} else {
  launcher.setLoginResult(false, 'Invalid credentials');
}
});

launcher.addEventListener('app-click', (e) => {
  console.log('App clicked:', e.detail.app.name);
});

launcher.addEventListener('power-action', (e) => {
  console.log('Power action:', e.detail.action); // 'shutdown' | 'restart' | 'sleep' | 'lock'
});

document.body.appendChild(launcher);

```

Properties

Property	Type	Default	Description
mode	'login' 'home'	'home'	Current view mode
loginConfig	ILoginConfig	{}	Login screen configuration
apps	IAppIcon[]	[]	Array of app icons to display
statusConfig	IStatusBarConfig	{}	Status bar visibility options
topBarConfig	ITopBarConfig	{}	Top bar visibility options
userName	string	'User'	Display name for user avatar
notificationCount	number	0	Number of notifications
batteryLevel	number 'charging'	100	Battery percentage or charging state
networkStatus	'online' 'offline' 'connecting'	'online'	Network connection status
soundLevel	number	50	Volume level (0-100)

Events

Event	Detail	Description
login-attempt	{ method, value }	Fired when user attempts to log in
login-success	—	Fired after successful login
login-failure	{ error }	Fired after failed login
app-click	{ app }	Fired when an app icon is clicked
power-action	{ action }	Fired when power menu action is selected
volume-change	{ volume }	Fired when volume is adjusted
wifi-toggle	{ enabled }	Fired when WiFi is toggled
keyboard-toggle	{ visible }	Fired when virtual keyboard is toggled

System Menu Components

Interactive status bar menus for system controls.

EcoApplauncherWifimenu

WiFi network selection and management.

```
import { EcoApplauncherWifimenu, type IWiFiNetwork } from '@ecobridge.xyz/catalog';

const wifiMenu = document.createElement('eco-applauncher-wifimenu');
wifiMenu.open = true;
wifiMenu.wifiEnabled = true;
wifiMenu.connectedNetwork = 'HomeNetwork';
wifiMenu.networks = [
  { ssid: 'HomeNetwork', strength: 4, secured: true },
  { ssid: 'OfficeWiFi', strength: 3, secured: true },
  { ssid: 'GuestNetwork', strength: 2, secured: false },
];
```

EcoApplauncherBatteryMenu

Battery status and power mode settings.

```
import { EcoApplauncherBatteryMenu } from '@ecobridge.xyz/catalog';

const batteryMenu = document.createElement('eco-applauncher-batterymenu');
```

```
batteryMenu.open = true;
batteryMenu.batteryLevel = 75;
batteryMenu.isCharging = false;
batteryMenu.batterySaverEnabled = false;
batteryMenu.timeRemaining = '3h 45m';
```

EcoApplauncherSoundmenu

Audio controls and device selection.

```
import { EcoApplauncherSoundmenu, type IAudioDevice } from '@ecobridge.xyz/catalog';

const soundMenu = document.createElement('eco-applauncher-soundmenu');
soundMenu.open = true;
soundMenu.volume = 70;
soundMenu.muted = false;
soundMenu.outputDevices = [
  { id: '1', name: 'Built-in Speakers', type: 'speaker' },
  { id: '2', name: 'AirPods Pro', type: 'headphones' },
];
soundMenu.activeDeviceId = '1';
```

EcoApplauncherPowermenu

Power actions menu.

```
import { EcoApplauncherPowermenu, type TPowerAction } from '@ecobridge.xyz/catalog';

const powerMenu = document.createElement('eco-applauncher-powermenu');
powerMenu.open = true;
powerMenu.addEventListener('power-action', (e) => {
  const action: TPowerAction = e.detail.action; // 'shutdown' | 'restart' | 'sleep' | 'lock'
  handlePowerAction(action);
});
```

EcoApplauncherKeyboard

Virtual on-screen keyboard for touch interfaces.

```
import { EcoApplauncherKeyboard } from '@ecobridge.xyz/catalog';
```

```
const keyboard = document.createElement('eco-applauncher-keyboard');
keyboard.visible = true;
keyboard.addEventListener('key-press', (e) => console.log('Key:', e.detail.key));
keyboard.addEventListener('backspace', () => console.log('Backspace'));
keyboard.addEventListener('enter', () => console.log('Enter'));
keyboard.addEventListener('space', () => console.log('Space'));
```

EcoScreensaver

An elegant screensaver with a floating time display that changes color on bounce — reminiscent of classic DVD screensavers but with modern design sensibility.

```
import { EcoScreensaver } from '@ecobridge.xyz/catalog';

// Show screensaver
const screensaver = await EcoScreensaver.show();

// Hide screensaver
EcoScreensaver.hide();

// Destroy instance completely
EcoScreensaver.destroy();

// With auto-activation delay (inactivity timer)
const screensaver = new EcoScreensaver();
screensaver.delay = 300000; // 5 minutes
document.body.appendChild(screensaver);
```

The screensaver features:

- Floating time display with smooth animation
- Color changes on boundary bounce
- Click-to-dismiss with radial reveal animation
- Elegant vignette effect
- Optional auto-activation after inactivity

Views

Pre-built view components for common screens.

EcoViewLogin

Standalone login view with PIN, password, and QR authentication.

```
import { EcoViewLogin, type ILoginConfig } from '@ecobridge.xyz/catalog';

const loginView = document.createElement('eco-view-login');
loginView.config = {
  allowedMethods: ['pin', 'password'],
  pinLength: 4,
  welcomeMessage: 'Welcome',
  subtitle: 'Sign in to continue',
  logoUrl: '/logo.svg',
};

loginView.addEventListener('login-attempt', (e) => {
  const { method, value } = e.detail;
  // Handle authentication
});
```

EcoViewHome

App grid view displaying application icons.

```
import { EcoViewHome } from '@ecobridge.xyz/catalog';

const homeView = document.createElement('eco-view-home');
homeView.apps = [
  { name: 'App 1', icon: 'lucide:box' },
  { name: 'App 2', icon: 'lucide:star' },
];
```

EcoViewSettings

Complete settings panel with categorized options.

```
import { EcoViewSettings } from '@ecobridge.xyz/catalog';

const settingsView = document.createElement('eco-view-settings');
```

```
settingsView.activePanel = 'general'; // 'general' | 'network' | 'display' | 'sound' | etc.
```

Settings panels include:

- **General** — Dark mode, text size, language, timezone
- **Network** — WiFi settings, available networks
- **Bluetooth** — Bluetooth toggle, device management
- **Display** — Brightness, Night Shift, resolution
- **Sound** — Volume, output device, sound effects
- **Notifications** — App notification settings
- **Privacy** — Location, camera, microphone permissions
- **Accounts** — User profile, connected accounts
- **Apps** — Default apps, installed applications
- **Updates** — Software update settings
- **About** — System information

Additional Views

- `EcoViewPeripherals` — Peripheral device management
- `EcoViewSaasshare` — SaaS sharing configuration
- `EcoViewSystem` — System-level settings

☐ Theming

The library uses CSS custom properties for consistent theming. Import the theme defaults for access to spacing, radius, shadows, and transitions.

```
import { themeDefaultStyles, themeDefaults } from '@ecobridge.xyz/catalog';

// Use in your component styles
@customElement('my-component')
export class MyComponent extends DeesElement {
  static styles = [
    themeDefaultStyles, // Adds CSS custom properties
    css`
      .card {
        padding: var(--dees-spacing-lg);
        border-radius: var(--dees-radius-lg);
        box-shadow: var(--dees-shadow-md);
        transition: transform var(--dees-transition-default) ease;
      }
    `
  ];
}
```

```
    }  
    ,  
    1;  
}
```

Available CSS Custom Properties

```
/* Spacing */  
--dees-spacing-xs: 4px;  
--dees-spacing-sm: 8px;  
--dees-spacing-md: 12px;  
--dees-spacing-lg: 16px;  
--dees-spacing-xl: 24px;  
--dees-spacing-2xl: 32px;  
--dees-spacing-3xl: 48px;  
  
/* Border Radius */  
--dees-radius-xs: 2px;  
--dees-radius-sm: 4px;  
--dees-radius-md: 6px;  
--dees-radius-lg: 8px;  
--dees-radius-xl: 12px;  
--dees-radius-full: 999px;  
  
/* Shadows */  
--dees-shadow-xs: 0 1px 2px 0 rgb(0 0 0 / 0.05);  
--dees-shadow-sm: 0 1px 3px rgba(0, 0, 0, 0.1);  
--dees-shadow-md: 0 2px 8px rgba(0, 0, 0, 0.15);  
--dees-shadow-lg: 0 4px 12px rgba(0, 0, 0, 0.15);  
  
/* Transitions */  
--dees-transition-fast: 0.1s;  
--dees-transition-default: 0.15s;  
--dees-transition-slow: 0.2s;  
--dees-transition-slower: 0.3s;  
  
/* Control Heights */  
--dees-control-height-sm: 32px;  
--dees-control-height-md: 36px;
```

```
--dees-control-height-lg: 40px;  
--dees-control-height-xl: 48px;
```

Development

```
# Install dependencies  
pnpm install  
  
# Start development server with hot reload  
pnpm run watch  
  
# Build for production  
pnpm run build  
  
# Run tests  
pnpm test
```

Dependencies

- `@design.estate/dees-catalog` — Base component library
- `@design.estate/dees-element` — Web component foundation (LitElement-based)
- `@design.estate/dees-domtools` — DOM utilities

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.

Revision #6

Created 2026-03-28 10:49:15 UTC by foss.global Team

Updated 2026-03-28 12:14:40 UTC by foss.global Team