



GPU Component

Rust GUI components for building fantastic cross-platform desktop application by using [GPU](#).

[Get Started](#)[Components](#)

60+ Components

Comprehensive library of cross-platform desktop UI components for building feature-rich applications.

High Performance

Virtualized Table and List components for smooth rendering of large datasets with minimal memory footprint.

Themeable

Built-in theme system with with 20+ themes, and dark mode out of the box.

Flexible Layouts

Dock layout for panel arrangements, resizable panels, and freeform layouts for any application structure.

Data Visualization

Built-in chart components for visualizing data with Line, Bar, Area, and Pie charts.

Code Editor

High-performance code editor with LSP support, syntax highlighting, powered by Tree-sitter and Rope.

Simple and Intuitive API

Get started with just a few lines of code. Stateless components make it easy to build complex UIs.

```
Button::new("ok")
    .primary()
    .label("Click Me")
    .on_click(|_, _, _| println!("Button clicked!"))
```

rs

Install GPU Component

Add the following to your `Cargo.toml` :

```
gpui-component = "0.2"
gpui = "0.2"
```

Hello World

The following `src/main.rs` is a simple "Hello, World!" application:

rs

```
use gpui::*;
use gpui_component::{button::*, *};

pub struct HelloWorld;
impl Render for HelloWorld {
    fn render(&mut self, _: &mut Window, _: &mut Context<Self>) -> impl IntoElement {
        div()
            .v_flex()
            .gap_2()
            .size_full()
            .items_center()
            .justify_center()
            .child("Hello, World!")
            .child(
                Button::new("ok")
                    .primary()
                    .label("Let's Go!")
                    .on_click(|_, _, _| println!("Clicked!")),
            )
    }
}

fn main() {
    let app = Application::new();

    app.run(move |cx| {
        // This must be called before using any GPUI Component features.
        gpui_component::init(cx);

        cx.spawn(async move |cx| {
            cx.open_window(WindowOptions::default(), |window, cx| {
                let view = cx.new(|_| HelloWorld);
                // This first level on the window, should be a Root.
                cx.new(|cx| Root::new(view.into(), window, cx))
            })?;

            Ok::<_, anyhow::Error>{()}
        })
        .detach();
    });
}
```

```
});  
}
```

Run the program with the following command:

```
$ cargo run
```

sh

GPUI Component is an open source project under the Apache-2.0 License, developed by [Longbridge](#).
[Contributors](#) | [llms-full.txt](#) | [Report Bug](#) | [Discussion](#)