Techniques for Creating Reusable Components

Introduction

Reusable components in Blazor streamline development by allowing you to adapt components to different needs without rewriting code. This guide will show you how to use parameterization, templates, and inheritance to make components flexible and maintainable.

Instructions

1. Set Up Component Parameterization

- Add Parameters: Open the component's `.razor` file and define parameters with the `[Parameter]` attribute. This attribute allows different values to be passed in and modifies the component's behavior.
- Usage: Adjust properties like labels, colors, or sizes through parameters, enabling easy reuse across the application.
- Example: For a button component, add a label parameter to adapt the button for "Submit," "Cancel," or "Reset" simply by setting the label when you use it.

2. Apply Templates for Content Flexibility

- Define a Template: In your component, add a `RenderFragment` parameter, enabling custom content (e.g., UI elements) to be passed in.
- Customize Content: Specify template content in the parent component, allowing the same component structure to display different content as needed.
- Example: In a list component, use a template to change how items appear, like showing product images in one view and prices in another, without altering the list structure.

3. Use Component Inheritance for Shared Logic

- Create a Base Component: Develop a base component with shared functionality, such as form validation logic.
- Extend the Component: Inherit from the base component to create specialized components with unique fields or features.

- Example: Extend a base form component to create login and registration forms. Each specialized form can add specific fields while keeping the shared validation in the base component.

Conclusion

Combining parameterization, templates, and inheritance allows you to create adaptable, reusable components that streamline Blazor development. These techniques reduce code duplication and make your applications more accessible to manage, extend, and customize.