Vue.js Declarative Rendering

Topics : Vue
Written on January 10, 2024

Vue.js is a JavaScript framework for building user interfaces, and it provides a declarative rendering system. Declarative rendering is a paradigm where you describe the desired outcome, and the framework takes care of the underlying logic to achieve that outcome. In Vue.js, this is often done through the use of templates.

1. **Template Syntax:** Vue.js uses an HTML-based template syntax to bind the DOM with the underlying Vue instance data. The template syntax allows you to declaratively describe the structure and behavior of the UI.

   ```
   <div id="app">
   {{ message }}
   </div>
   ```

2. **Vue Instance:** You create a Vue instance and associate it with the HTML element where you want your Vue application to take effect. The data properties defined in the Vue instance can be used in the template.

   ```
   var app = new Vue({
   el: '#app',
   data: {
   message: 'Hello, Vue!'
   }
   });
   ```

3. **Data Binding:** Vue.js uses two-way data binding to link the data properties in the Vue instance with the template. In the example above, the `{{ message }}` in the template is bound to the `message` property in the Vue instance. Any changes to the `message` property will automatically reflect in the rendered view.

   ```
   app.message = 'Updated message!';
   ```

4. **Conditional Rendering:** Vue.js allows you to conditionally render elements based on the data properties using directives like `v-if` and `v-else`.

   ```
   <p v-if="seen">Now you see me</p>
   <p v-else>Now you don't</p>
   ```

In this example, the element will only be rendered if the `seen` property in the data is true.