

# Vue.js Slots

Topics : [Vue](#)

Written on [January 11, 2024](#)

In Vue.js, slots provide a mechanism for composing components and creating flexible, reusable layouts. Slots allow a component to define a placeholder in its template where the parent component can insert content. This makes it easy to create dynamic and customizable components. There are two types of slots in Vue.js: named slots and scoped slots.

## Named Slots:

Named slots allow you to define multiple placeholders in a component's template and give each slot a name. The parent component can then specify which content goes into which slot.

### 1. Child Component (ChildComponent.vue):

```
<template>
  <div>
    <h2><slot name="header">Default Header</slot></h2>
  </div><slot></slot></div>
  <footer><slot name="footer">Default Footer</slot></footer>
</div>
</template>
```

In this example, the component has three named slots: header, default (no name), and footer.

### 2. Parent Component:

```
<template>
  <child-component>
    <template v-slot:header>
      <h1>Custom Header</h1>
    </template>
```

```
    <p>This is the main content.</p>
```

```
    <template v-slot:footer>
      <p>Custom Footer</p>
    </template>
  </child-component>
</template>
```

```

<script>
import ChildComponent from './ChildComponent.vue';

export default {
  components: {
    'child-component': ChildComponent
  }
};
</script>

```

In the parent component, the `v-slot` directive is used to specify the content for each named slot.

## Scoped Slots:

Scoped slots allow the child component to pass data to the parent component. The parent component can then use that data to customize the content it provides to the child component.

### 1. Child Component (ChildComponent.vue):

```

<template>
<div>
<h2><slot name="header" :title="title">Default Header</slot></h2>
<div><slot :items="items"></slot></div>
<footer><slot name="footer">Default Footer</slot></footer>
</div>
</template>

<script>
export default {
  data() {
    return {
      title: 'Scoped Slot Example',
      items: [1, 2, 3, 4, 5]
    };
  }
};
</script>

```

In this example, the header slot receives a `title` prop, and the default slot receives an `items` prop.

### 2. Parent Component:

```

<template>
  <child-component>
    <template v-slot:header="{ title }">

```

```
    <h1>{{ title }}</h1>
  </template>

  <template v-slot="{ items }">
    <ul>
      <li v-for="item in items" :key="item">{{ item }}</li>
    </ul>
  </template>

  <template v-slot:footer>
    <p>Custom Footer</p>
  </template>
</child-component>
</template>

<script>
import ChildComponent from './ChildComponent.vue';

export default {
  components: {
    'child-component': ChildComponent
  }
};
</script>
```

In the parent component, the `v-slot` directive is used with destructuring to access the data passed from the child component.