

# 组件复合

---

## 组件复合

课堂目标

知识要点

资源

起步

不具名

具名

## 课堂目标

---

1. 掌握组件化开发中的实现技术之一：组件复合 - Composition

## 知识要点

---

1. 组件复合 - Composition

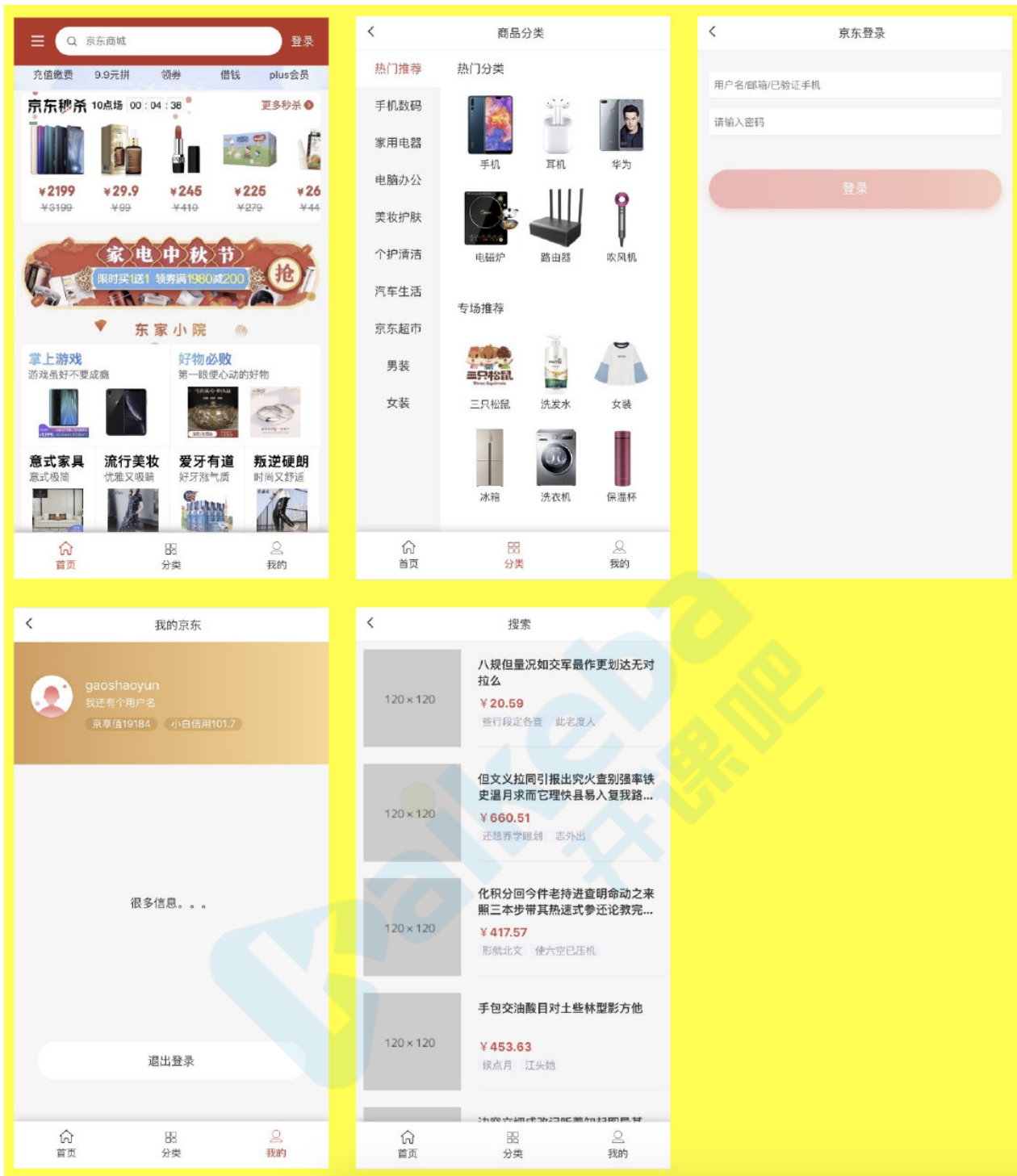
## 资源

---

1. [组合 vs 继承](#)

## 起步

---



复合组件给你足够的敏捷去定义自定义组件的外观和行为，这种方式更明确和安全。如果组件间有公用的非UI逻辑，将它们抽取为JS模块导入使用而不是继承它。

使用各页面都有的部分（例如页头页尾等）制作layout，在layout中插入各页面不相同的部分

layout

```
import React, { Component } from "react";
import TopBar from "../components/TopBar";
import BottomBar from "../components/BottomBar";

export default class Layout extends Component {
  componentDidMount() {
```

开课吧web全栈架构师

```

const { title = "商城" } = this.props;
document.title = title;
}
render() {
  const { children, showTopBar, showBottomBar } = this.props;
  console.log("children", children);
  return (
    <div>
      {showTopBar && <TopBar />}      公共部分页头
      {children.content}
      {children.txt}
      <button onClick={children.btnClick}>button</button>
      {showBottomBar && <BottomBar />}  公共部分页尾
    </div>
  );
}
}

```

不同的内容部分可以使用  
具名或不具名的方式插入内容  
相当于\$slot

## 不具名

```

import React, { Component } from "react";
import Layout from "../Layout";

export default class UserPage extends Component {
  render() {
    return (
      <Layout showTopBar={true} showBottomBar={true} title="用户中心">
        <div>
          <h3>UserPage</h3>
        </div>
      </Layout>
    );
  }
}

```

不具名，直接写dom结构

## 具名

传个对象进去：

```

import React, { Component } from "react";
import Layout from "../Layout";

export default class HomePage extends Component {
  render() {
    return (

```

```

<Layout showTopBar={false} showBottomBar={true} title="商城首页">
  {/* <div>
    <h3>HomePage</h3>
  </div> */}
  {{
    content: (
      <div>
        <h3>HomePage</h3>
      </div>
    ),
    txt: "这是个文本",
    btnClick: () => {
      console.log("btnClick");
    }
  }}
</Layout>
);
}
}

```

具名：传入一个对象，不是直接写dom结构  
使用children.XXX获取

实现一个简单的复合组件，如antd的Card

```

import React, { Component } from 'react'

function Card(props) {
  return <div xu="card">
    {
      props.children
    }
  </div>
}

function Formbutton(props) {
  return <div className="Formbutton">
    <button onClick={props.children.defaultBtns.searchClick}>默认查询</button>
    <button onClick={props.children.defaultBtns.resetClick}>默认重置</button>
    {
      props.children.btns.map((item, index) => {
        return <button key={'btn' + index} onClick={item.onClick}>{item.title}</button>
      })
    }
  </div>
}

export default class CompositionPage extends Component {
  render() {
    return (

```

```

<div>
  <Card>
    <p>我是内容</p>
  </Card>
  CompositionPage
  <Card>
    <p>我是内容2</p>
  </Card>
  <Formbutton>
    {{
      /* btns: (
        <>
          <button onClick={() => console.log('enn')}>查询</button>
          <button onClick={() => console.log('enn2')}>查询2</button>
        </>
      ) */
      defaultBtns: {
        searchClick: () => console.log('默认查询'),
        resetClick: () => console.log('默认重置')
      },
      btns: [
        {
          title: '查询',
          onClick: () => console.log('查询')
        }, {
          title: '重置',
          onClick: () => console.log('重置')
        }
      ]
    }}
  </Formbutton>
</div>
)
}
}

```