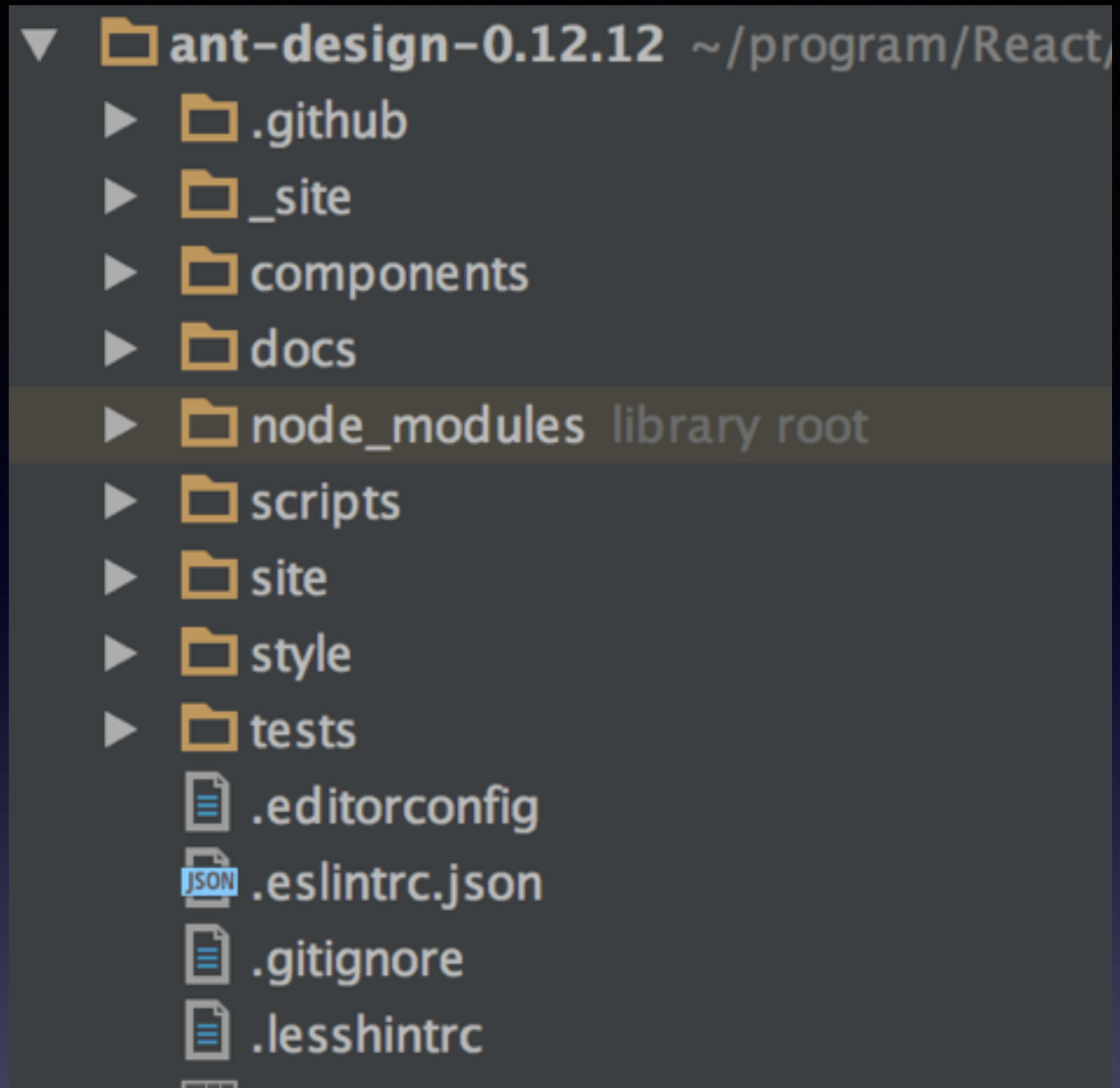


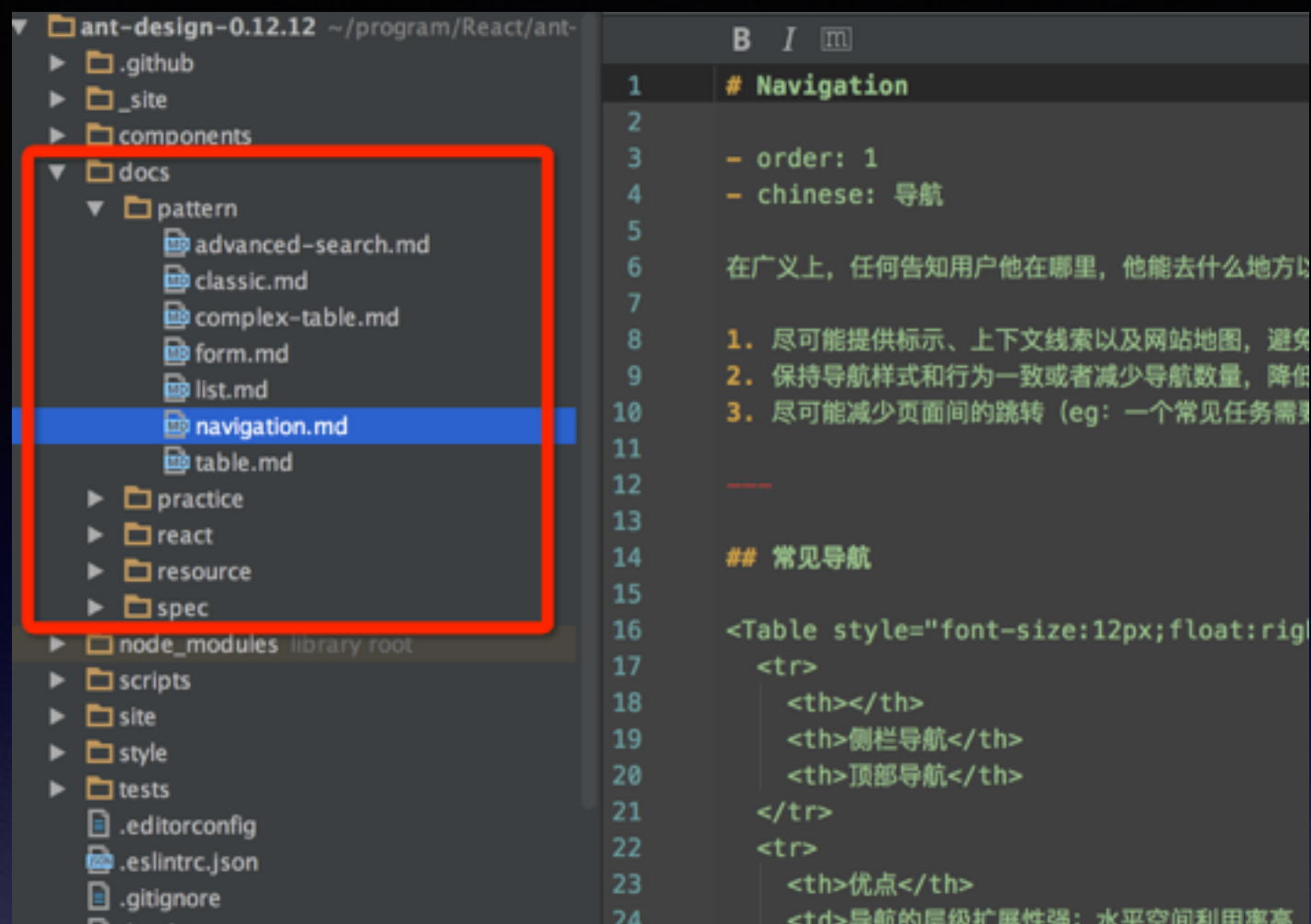
蚂蚁金服源码解析

——markdown 文档的展示

长什么样？



源码基于 ant-design-0.12.12



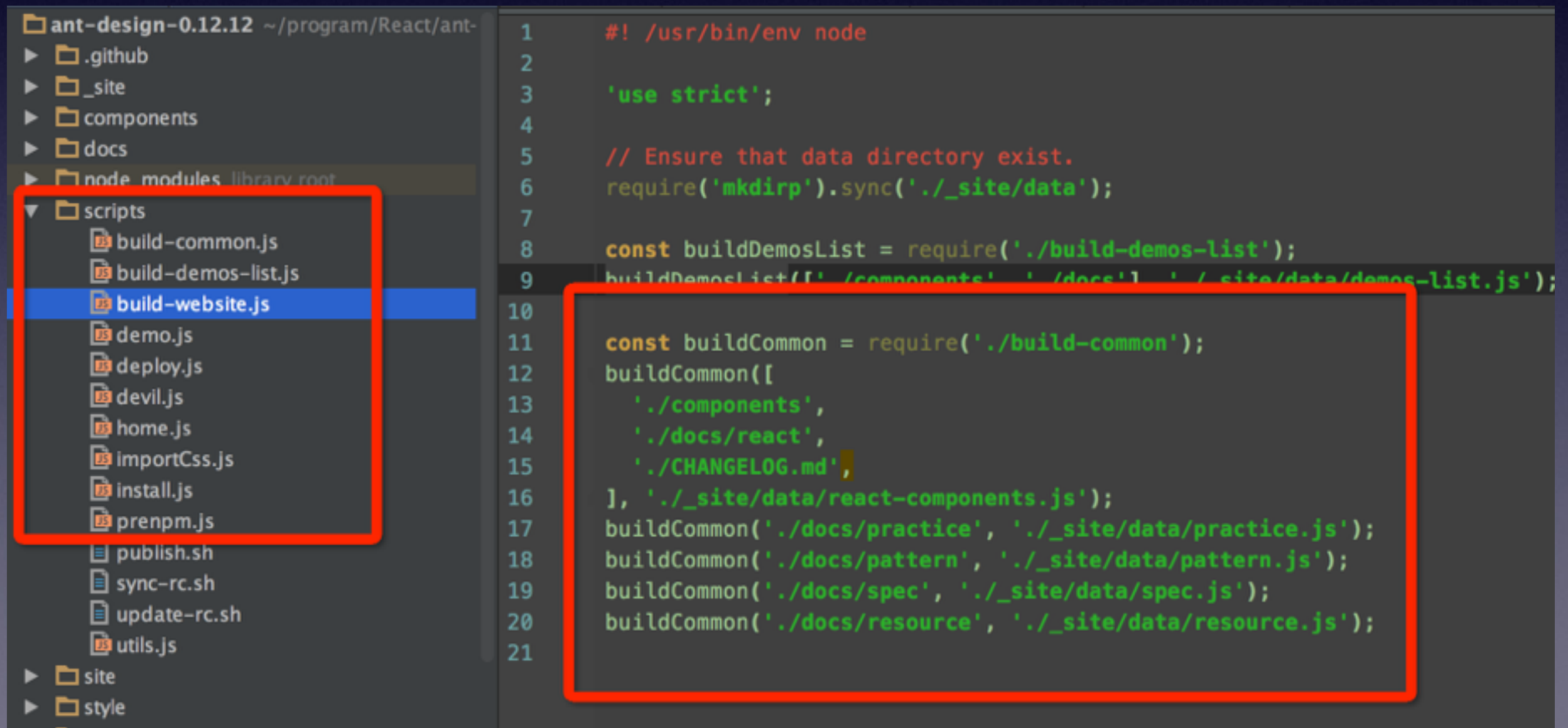
如图所示, 左图是要显示的 markdown文档, 右图是展示效果, 展示内容分为几种类型:

- 各级标题
- 正常内容
- 带项目符合和编号的内容
- 内部导航
- 内置网页
- 图片和视频等
- 其他

怎么长的？

服务端预处理

- 示例配置了简单的服务器，通过 node，对 docs 下面的源文件进行了预处理，具体代码如图



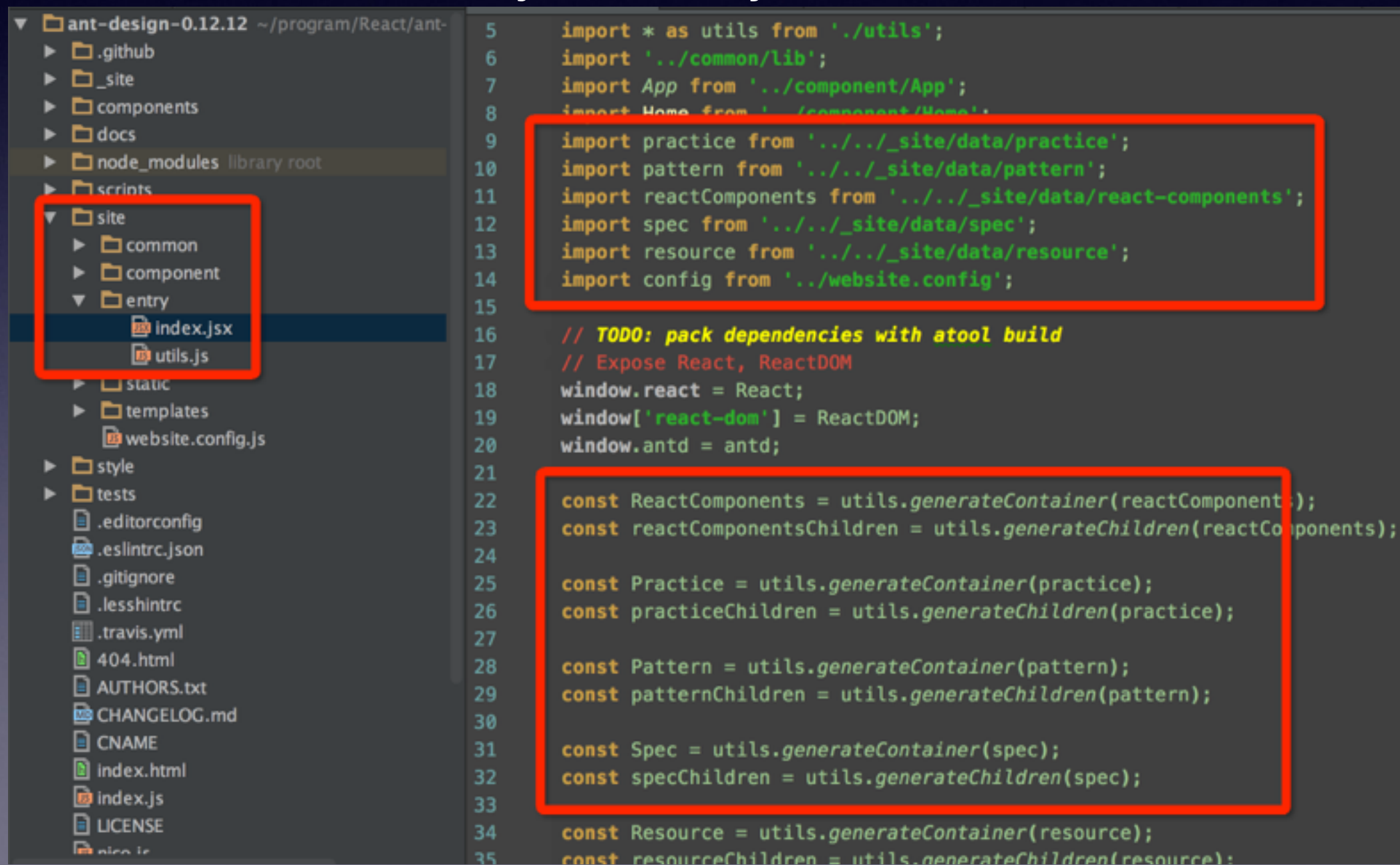
服务端预处理

- 程序运行之后，会初始化一个_site文件夹，前端真正渲染的是这个文件夹下面的内容，而不是 docs 下面的内容，如下图



前端处理流程

- 入口是 site/entry/index.jsx, 代码如图所示

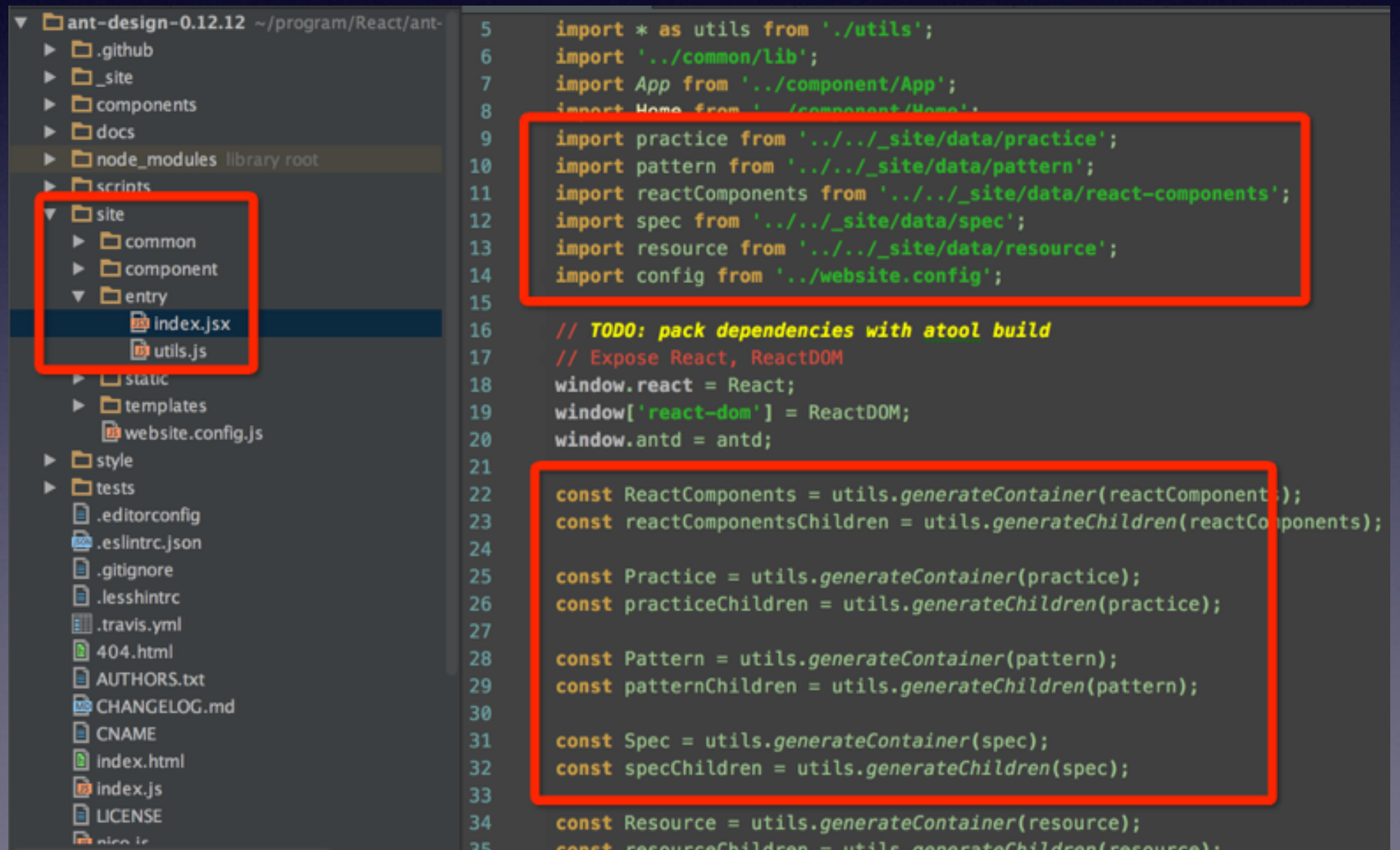


The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the directory structure of the project, with the file `site/entry/index.jsx` highlighted. The code editor shows the content of `site/entry/index.jsx`, which is a JavaScript file that imports various modules and uses `utils.generateContainer` and `utils.generateChildren` to generate components and their children.

```
5 import * as utils from './utils';
6 import '../common/lib';
7 import App from '../component/App';
8 import Home from '../component/Home';
9 import practice from '../../../_site/data/practice';
10 import pattern from '../../../_site/data/pattern';
11 import reactComponents from '../../../_site/data/react-components';
12 import spec from '../../../_site/data/spec';
13 import resource from '../../../_site/data/resource';
14 import config from '../website.config';
15
16 // TODO: pack dependencies with atool build
17 // Expose React, ReactDOM
18 window.react = React;
19 window['react-dom'] = ReactDOM;
20 window.antd = antd;
21
22 const ReactComponents = utils.generateContainer(reactComponents);
23 const reactComponentsChildren = utils.generateChildren(reactComponents);
24
25 const Practice = utils.generateContainer(practice);
26 const practiceChildren = utils.generateChildren(practice);
27
28 const Pattern = utils.generateContainer(pattern);
29 const patternChildren = utils.generateChildren(pattern);
30
31 const Spec = utils.generateContainer(spec);
32 const specChildren = utils.generateChildren(spec);
33
34 const Resource = utils.generateContainer(resource);
35 const resourceChildren = utils.generateChildren(resource);
```

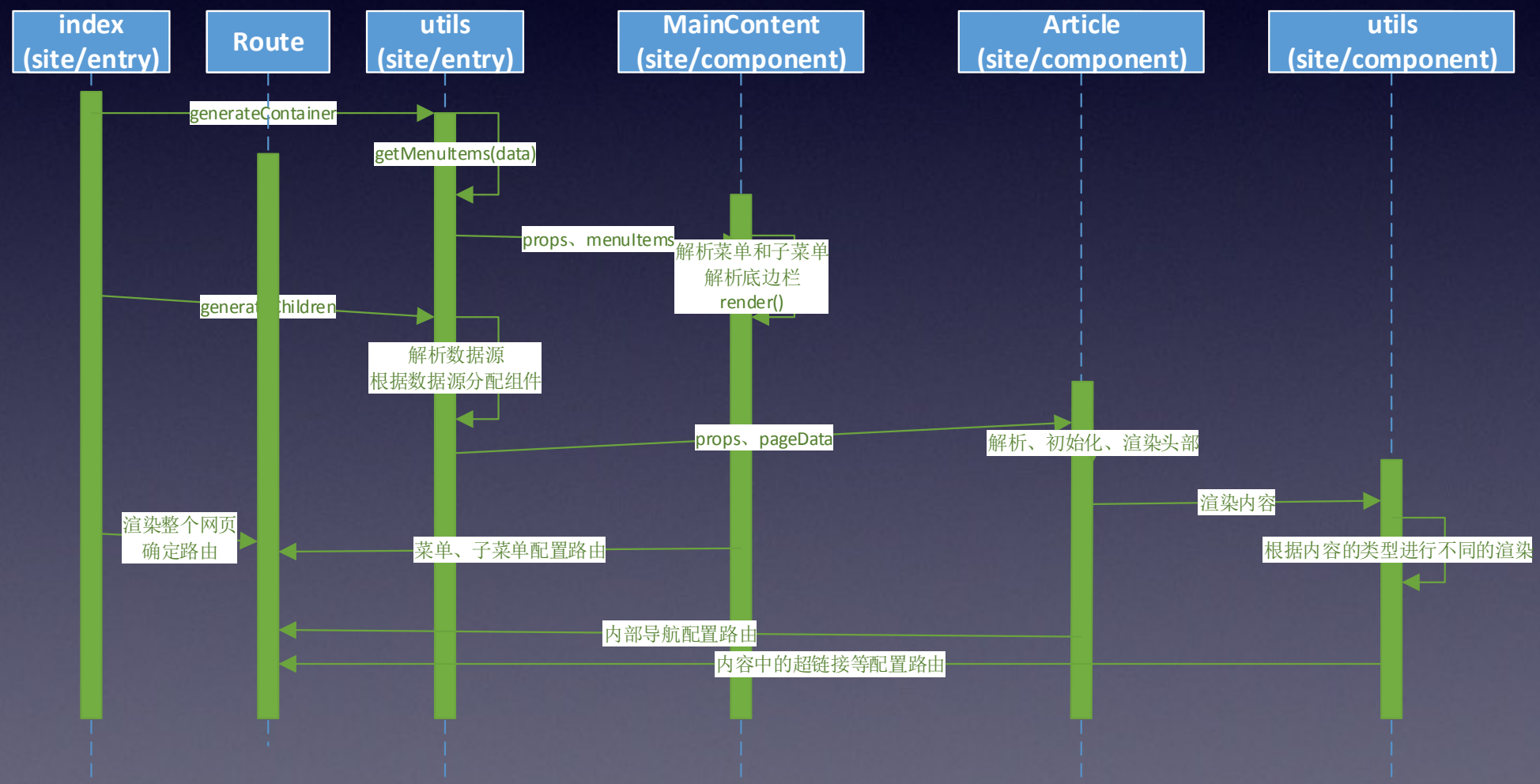

前端处理流程

- 每一个菜单对应docs 下面的一个文件夹, 每一个文件夹以一个组件的形式渲染, 每一个组件下的文件作为 children 渲染

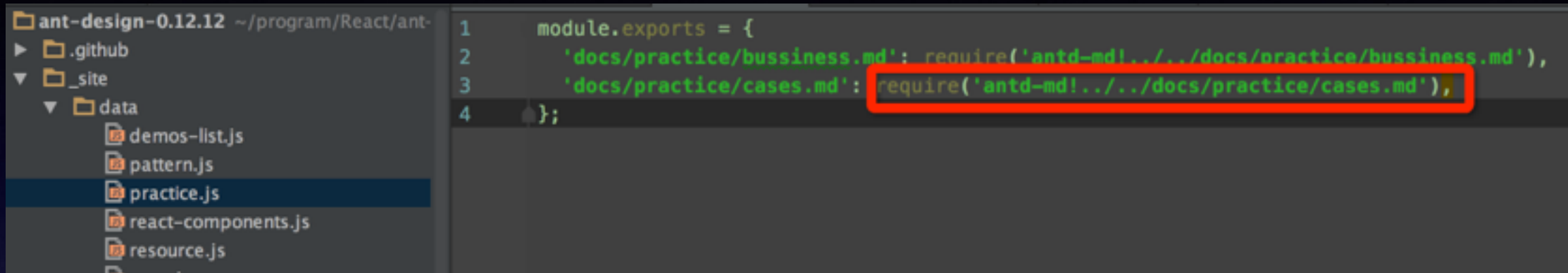


前端处理流程

- 以**实践**为例，对应docs 下面的**practice**文件夹，处理流程如下图



数据形式



```
1 module.exports = {  
2   'docs/practice/bussiness.md': require('antd-md!../docs/practice/bussiness.md'),  
3   'docs/practice/cases.md': require('antd-md!../docs/practice/cases.md'),  
4 };
```

- 红框中的代码，使用了 antd-md 组件，会将文件解析成数据

```
"devDependencies": {  
  "antd-md-loader": "0.1.0-beta.11",  
  "babel-plugin-transform-runtime": "6.23.0"
```


- 解析菜单时，处理的数据形式如下图所示

```
function getMenuItems(data) {  
  const menuMeta = Object.keys(data)  
    .map((key) => data[key])  
    .map((file) => file.meta);  
  console.log(menuMeta);  
  const menuItems = {};  
  menuMeta.sort((a, b) => {  
    return parseInt(a.order, 10) - parseInt(b.order, 10);  
  })).forEach((meta) => {  
    const category = meta.category || 'topLevel';  
    if (!menuItems[category]) {  
      menuItems[category] = {};  
    }  
  
    const type = meta.type || 'topLevel';  
    if (!menuItems[category][type]) {  
      menuItems[category][type] = [];  
    }  
  
    menuItems[category][type].push(meta);  
  });  
}
```

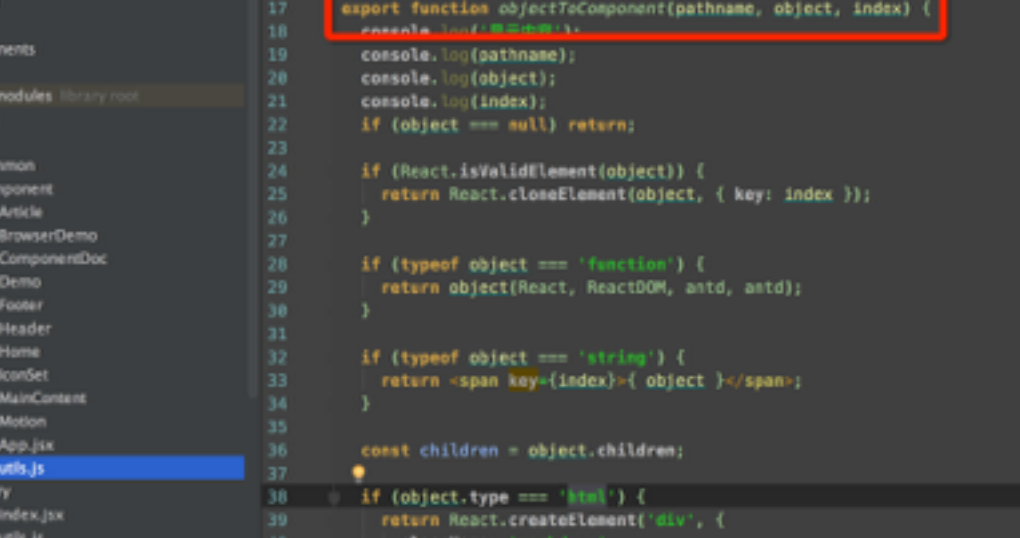
```
{order: "1", disabled: "true", chinese: "业务组件", english: "Business", fileName: "docs/practice/bussiness.md"}  
{order: "0", chinese: "实践案例", english: "Cases", fileName: "docs/practice/cases.md"}  
{order: "6", chinese: "高级搜索", english: "Advanced Search", fileName: "docs/pattern/advanced-search.md"}  
{order: "0", disabled: "true", chinese: "典型页面", english: "Classic", fileName: "docs/pattern/classic.md"}  
{order: "5", chinese: "表格: 复杂数据", english: "Complex Table", fileName: "docs/pattern/complex-table.md"}
```


- 解析文档时，处理的数据形式如下图所示

```
export function generateChildren(data) {
  const pagesData = getPagesData(data);
  const children = pagesData.map((pageData, index) => {
    console.log("pageData");
    console.log(pageData);
    const meta = pageData.meta;

    const hasDemos = demosList[meta.fileName];
    const Wrapper = !hasDemos ?
      (props) => <Article {...props} content={pageData} /> :
      (props) => <ComponentDoc {...props} doc={pageData} />;
    return (
      <Route key={index}
        path={fileNameToPath(meta.fileName)}
        component={Wrapper} />
    );
  });
}
```

```
Object
  api: [] (0)
  description: Array (26)
    0 > {type: "p", children: "Ant Design 是面向中台的 UI 设计语言。"}
    1 > {type: "p", children: "从 2015 年 4 月起, Ant Design 在蚂蚁金服中后台产品线迅速推广, 对接多条业务线, 覆盖速度快、实现效果好等特点, 并"}
    2 > {type: "p", children: "Ant Design 目前在外部也有 <a href='\"https://github.com/ant-design/issues/477\"'>许多产品实践</a>, 如"}
    3 > {type: "h2", children: "最佳实践"}
    4 > {type: "hr"}
    5 > {type: "h3", children: "金融云"}
    6 > {type: "p", children: "金融云是面向金融机构深度定制的行业云计算服务; 助力金融机构向新金融转型升级, 推动平台、数据和技术方面的能力全面对"}
    7 > {type: "p", children: "<a class='\"outside-link\"' href='\"https://www.cloud.alipay.com/\"' target='\"_blank\"'>立即访问</a>"}
    8 > {type: "hr"}
    9 > {type: "h3", children: "OceanBase Cloud Platform"}
    10 > {type: "p", children: "OceanBase 是一款真正意义上的云端分布式关系型数据库, 而 OceanBase Cloud P_ (云平台) 是以 OceanBase 数据库"}
    11 > {type: "p", children: "<a class='\"outside-link internal\"' href='\"http://oceanbase.alipay.com/\"' target='\"_blank\"'>立即访问"}
    12 > {type: "hr"}
    13 > {type: "h3", children: "服务宝体验平台"}
    14 > {type: "p", children: "体验平台是收集用户与公司所有的接触点 (包括来电咨询 / 微博等渠道) 的数据, 通过数据挖掘和体验同学运营的方式推送给公"}
    15 > {type: "p", children: "<a class='\"outside-link internal\"' href='\"http://tiyan.alipay.com/\"' target='\"_blank\"'>立即访问"}
    16 > {type: "hr"}
    17 > {type: "h3", children: "AntV"}
    18 > {type: "p", children: "AntV 将数据图形小组近几年在探索数据可视化过程中取得的成果进行总结和沉淀, 并分享给所有需要数据可视理论的人。"}
    19 > {type: "p", children: "<a class='\"outside-link internal\"' href='\"http://antv.alipay.net/\"' target='\"_blank\"'>立即访问"}
    20 > {type: "html", children: "<style> preview-image-boxes { margin-top: -36px; } .toc { display: none; } </style>"}
    21 > {type: "html", children: "<script>(function() { var links = Array.apply(n_sh('timeout'); }, 1500); })();</script>"}
  intro: [] (0)
  meta: {order: "0", chinese: "实践案例", english: "Cases", fileName: "docs/practice/cases.md"}
```



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'github', 'site', 'components', 'docs', 'node_modules', 'scripts', and 'site'. The 'site' folder is expanded, showing subfolders like 'common', 'component', and 'entry'. The 'entry' folder is selected, showing files like 'index.jsx', 'utils.js', 'static', 'templates', and 'website.config.js'. The code editor shows the 'utils.js' file with the following code:

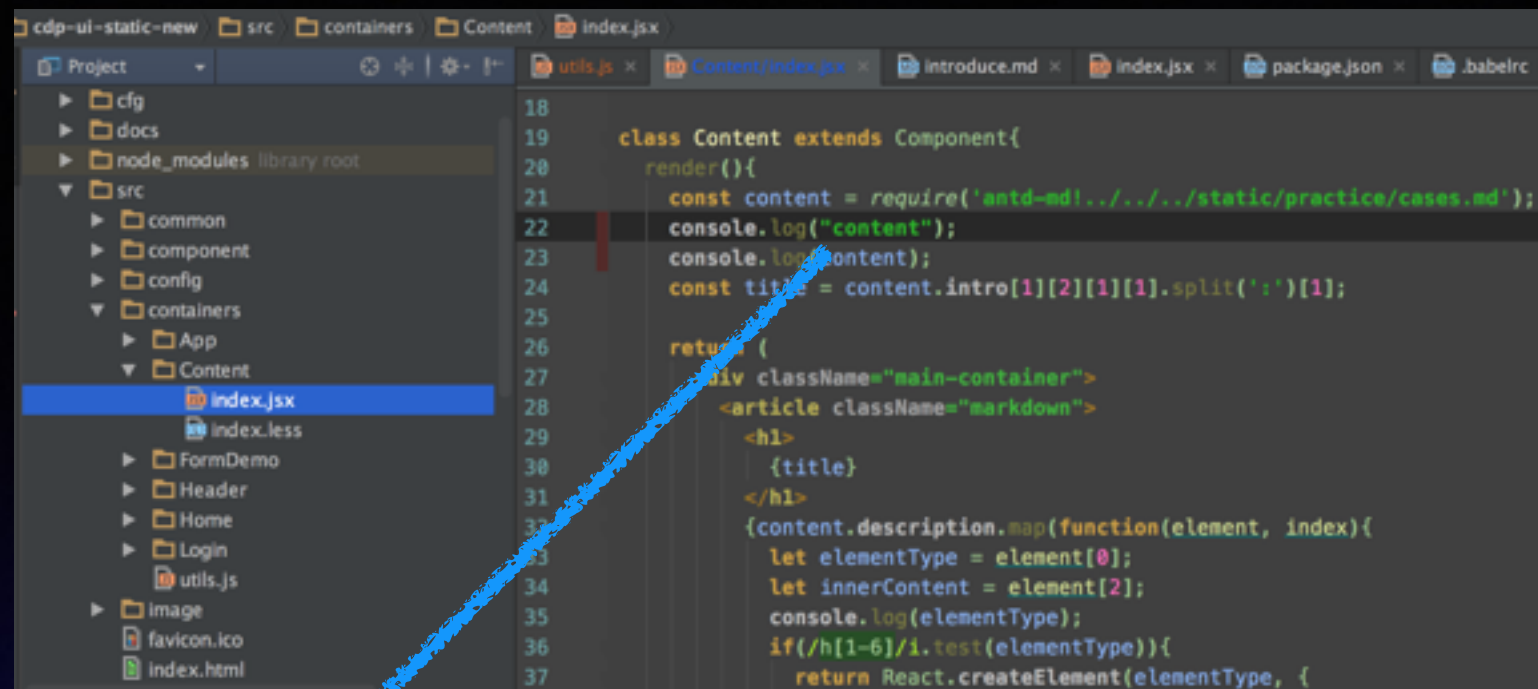
```

16
17
18 export function objectToComponent(pathname, object, index) {
19   console.log(pathname);
20   console.log(object);
21   console.log(index);
22   if (object === null) return;
23
24   if (React.isValidElement(object)) {
25     return React.cloneElement(object, { key: index });
26   }
27
28   if (typeof object === 'function') {
29     return object(React, ReactDOM, antd, astd);
30   }
31
32   if (typeof object === 'string') {
33     return <span key={index}>{ object }</span>;
34   }
35
36   const children = object.children;
37
38   if (object.type === 'html') {
39     return React.createElement('div', {
40       className: 'markdown',
41       key: index,
42       dangerouslySetInnerHTML: { __html: children }
43     });


```

能否模仿？

demo 里，简单实现了这种展示方式，数据源和 0.12.12 解析出来的有区别，如下图所示，实现方式见左图，把 Article 组件中的实现直接搬过来的



```
18
19 class Content extends Component{
20   render(){
21     const content = require('antd-md!../../static/practice/cases.md');
22     console.log("content");
23     console.log(content);
24     const title = content.intro[1][2][1][1].split(':')[1];
25
26     return (
27       <div className="main-container">
28         <article className="markdown">
29           <h1>
30             {title}
31           </h1>
32           {content.description.map(function(element, index){
33             let elementType = element[0];
34             let innerContent = element[2];
35             console.log(elementType);
36             if (/h[1-6]/.test(elementType)){
37               return React.createElement(elementType, {
```



```
content
Object
  api: [] (0)
  description: Array (28)
    0 ▶ ["p", {}, "Ant Design 是面向中台的 UI 设计语言。"] (3)
    1 ▶ ["p", {}, "从 2015 年 4 月起, Ant Design 在蚂蚁金服中后台产品线迅速推广, 对接多条业务线, 覆盖速度快、实现效果好等特点, 并且提供从界面设计到前端开发的全链路生态, 可以大大提升设计和开"]
    2 ▶ ["p", {}, "Ant Design 目前在外部也有 ", ["a", {title: null, href: "https://github.com/ant-design/ant-design/issues/477"}, "许多产品实践"], ", 如果你的公司和产品从"]
    3 ▶ ["h2", {}, "最佳实践"] (3)
    4 ▶ ["hr", {}] (2)
    5 ▶ ["h3", {}, "金融云"] (3)
    6 ▶ ["p", {}, Array, {}, Array, {}, Array] (7)
    7 ▶ ["p", {}, "金融云是面向金融机构深度定制的行业云计算服务; 助力金融机构向新金融转型升级, 推动平台、数据和技术方面的能力全面对外开放。"] (3)
    8 ▶ ["p", {}, ["a", {class: "outside-link", href: "https://www.cloud.alipay.com/", target: "_blank"}], "立即访问", [""]] (5)
    9 ▶ ["hr", {}] (2)
    10 ▶ ["h3", {}, "OceanBase Cloud Platform"] (3)
    11 ▶ ["p", {}, Array, {}, Array, {}, Array, {}, Array] (7)
    12 ▶ ["p", {}, "OceanBase 是一款真正意义上的云端分布式关系型数据库, 而 OceanBase Cloud P... (云平台) 是以 OceanBase 数据库为基础的云服务, 可以帮助用户快速创建、使用 OB 服务。"] (3)
    13 ▶ ["p", {}, ["a", {class: "outside-link internal", href: "http://oceanbase.alipay.com/", target: "_blank"}], "立即访问", [""]] (5)
    14 ▶ ["hr", {}] (2)
    15 ▶ ["h3", {}, "服务宝体验平台"] (3)
    16 ▶ ["p", {}, Array, {}, Array, {}, Array, {}, Array, {}, Array, {}, ...] (11)
    17 ▶ ["p", {}, "体验平台是收集用户与公司所有的接触点 (包括来电咨询 / 微博等渠道) 的数据, 通过数据挖掘和体验同学运营的方式推送给公司内部的业务团队 / 产品经理, 并推动体验问题解决, 从而实现良性"]
    18 ▶ ["p", {}, ["a", {class: "outside-link internal", href: "http://tiyan.alipay.com/", target: "_blank"}], "立即访问", [""]] (5)
    19 ▶ ["hr", {}] (2)
    20 ▶ ["h3", {}, "AntV"] (3)
    21 ▶ ["p", {}, Array, {}, Array, {}, Array] (7)
    22 ▶ ["p", {}, "AntV 将数据图形小组近几年在探索数据可视化过程中取得的成果进行总结和沉淀, 并分享给所有需要数据可视理论的人。"] (3)
    23 ▶ ["p", {}, ["a", {class: "outside-link internal", href: "http://antv.alipay.net/", target: "_blank"}], "立即访问", [""]] (5)
    24 ▶ ["style", {}, ".preview-image-boxes {margin-top: -36px;} .pr... background: #fff; .toc {display: none;}"] (3)
    25 ▶ ["script", {}, "(function() {var links = Array.apply(null, doc...nk.style.display = 'block';});})"] (3)
    26 ▶ ["p", {}, "function ping(url, callback) {var img = new... finish('timeout');}, 1500);})"] (3)
    27 ▶ ["", {}] (2)
  intro: [{"h1", "Cases"}, Array] (2)
```

App ready.



首页

实践

实践案例

Ant Design 是面向中台的 UI 设计语言。

从 2015 年 4 月起，Ant Design 在蚂蚁金服中后台产品线迅速推广，对接多条业务线，覆盖系统 40 个以上。定位于中台业务的 Ant Design 兼顾专业和非专业的设计人员，具有学习成本低、上手速度快、实现效果好等特点，并且提供从界面设计到前端开发的全链路生态，可以大大提升设计和开发的效率。

Ant Design 目前在外部也有

最佳实践

金融云

金融云是面向金融机构深度定制的行业云计算服务；助力金融机构向新金融转型升级，推动平台、数据和技术方面的能力全面对外开放。

OceanBase Cloud Platform

OceanBase 是一款真正意义上的云端分布式关系型数据库，而 OceanBase Cloud Platform（云平台）是以 OceanBase 数据库为基础的云服务，可以帮助用户快速创建、使用 OB 服务。

服务宝体验平台

有什么困难？

问题和难点

- ES 的很多语法都不熟，比如 箭头函数和 map 的用法

```
const jumper = content.description.filter((node) => {  
  return node.type === 'h2';  
}).map((node) => {  
  return (  
    <li key={node.children}>  
      <Link to={{ pathname: location.pathname, query: { scrollTo: node.children } }}  
        dangerouslySetInnerHTML={{ __html: node.children }} />  
    </li>  
  );  
});
```

```
const menuMeta = Object.keys(data)  
  .map((key) => data[key])  
  .map((file) => file.meta);
```

问题和难点

- React 的基础的内容只掌握了皮毛，比如createElement的用法，route 和组件的灵活使用

```
return React.createElement('div', {  
  className: 'markdown',  
  key: index,  
  dangerouslySetInnerHTML: { __html: children }  
});
```

```
const pagesData = getPagesData(data);  
const children = pagesData.map((pageData, index) => {  
  console.log("pageData");  
  console.log(pageData);  
  const meta = pageData.meta;  
  
  const hasDemos = demosList[meta.fileName];  
  
  const Wrapper = !hasDemos ?  
    (props) => <Article {...props} content={pageData} /> :  
    (props) => <ComponentDoc {...props} doc={pageData} />;  
  
  return (  
    <Route key={index}  
      path={fileNameToPath(meta.fileName)}  
      component={Wrapper} />  
  );  
});
```

怎么突破？

重要资料

- React 的API: <http://facebook.github.io/react/docs/top-level-api.html>
- JS 或者 ES 相关语法查询: https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Object/keys