38 首页的设计与开发

更新时间：2019-08-23 09:54:13

> 学习从来无捷径，循序渐进登高峰。
>
> —— 高永祚

我们先来制作首页的切图。首页一般是用来展示网站最核心内容的，可以从首页直达各个核心模块。我们实战中要做的这个网站是内容导购类网站，也就是发布一些实用的文章，例如游记、攻略、商品体验报告等，然后在文章中会包含一些推荐的商品，和蚂蜂窝有点类似。在这种网站中，最核心的就是文章，应该从各个角度来让用户找到感兴趣的内容。

我们这个首页整体的样式设计如下：

这个网页的最顶端是一个搜索框，可以直接使用样式库里的搜索组件，用户可以通过关键字搜索自己需要的内容。接下来是各种文章的分类，用户能通过分类直达各个分类对应的文章列表。再下面是猜你喜欢，这个地方的内容通常根据用户以前浏览的内容做的相关推荐，或者是官方想重点推广的文章，从这个模块的“查看更多”可以到达列表页，也可以点击每一个文章直达文章详情页。最底下是一个公共的网站导航条，这里我们放三个内容，“首页”就是我们这个页面，“发现”是官方推荐的文章列表页，“我的”栏目就是个人中心。下面我们先来开发这个首页。

文件的建立

我们把这些页面相关的文件还放在原来“tuitui-ui”这个项目里，在项目根目录下建立page目录，然后再建立 /page/index.html 作为这个首页的HTML页面，建立 /page/index.css 作为首页的 CSS 文件。我们先放上 HTML 的基础结构：

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="initial-scale=1, width=device-width, maximum-scale=1, user-scalable=no">
    <link rel="stylesheet" href="../dist/tuitui-ui.css">
    <link rel="stylesheet" href="./index.css">
    <title>推推优品-首页</title>
  </head>
  <body class="page-index">
    <!-- 首页 -->
  </body>
</html>
```

这个空文件里引入了两个样式，一个是/dist目录下的样式库文件，当切图制作完成后，交付给 JS 工程师的时候一定要说明项目依赖外部文件的位置，至于用哪种方式引入依赖，就和 JS 的技术栈有关了。在 HTML 文件还要注意一个地方，就是给 body 元素加上了“page-index”这个类，用来隔离不同页面间的样式，这个页面自定义的样式里都用这个类限制住，这样就不会产生样式冲突了。

这个页面还需要的另外一个样式就是 index.css，用来放页面里比较个性化的样式，也可以是对 UI 样式库的覆盖。/page/index.css 文件我们也先建立一个空文件：

```
/*
 * @Author: Rosen
 * @Date: 2019-08-17 11:41:09
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-08-17 22:21:02
 */
```

这样基础文件就建立好了，下面我们来开发首页的各块内容。

导航条的实现

我们先实现这个最简单的内容，这个导航条就是 tuitui-ui 里的导航，我们只需要把它的项稍作修改即可：

```
<!-- 底部导航条 -->
<div class="tt-navbar">
  <a class="navbar-item active">
    <i class="fa fa-home icon"></i>
    <span class="name">首页</span>
  </a>
  <a class="navbar-item" href="./list.html">
    <i class="fa fa-search icon"></i>
    <span class="name">发现</span>
  </a>
  <a class="navbar-item" href="./mine.html">
    <i class="fa fa-user-o icon"></i>
    <span class="name">我的</span>
  </a>
</div>
```

把这段代码放在 body 的前面就可以实现导航条的样式，这时应该会感觉实现的太快了。这样导航条的样式就完成了。

搜索组件的实现

首页的搜索功能我们会使用样式库里的搜索组件来完成。这里也没有特别需要强调的内容，直接调用即可。在 body 元素里追加：

```
<div class="tt-content">
  <!-- 带suggest的搜索框 -->
  <div class="tt-search">
  <!-- 搜索中的状态 -->
  <!-- <div class="tt-search on-focus"> -->
    <!-- 搜索框 -->
    <form class="tt-search-form" action="#">
      <div class="tt-search-input-wrap">
        <i class="fa fa-search tt-search-icon"></i>
        <input type="text" class="tt-search-input" placeholder="搜索" autocomplete="off" required/>
        <i class="fa fa-close tt-search-clear"></i>
      </div>
      <span class="tt-search-cancel">取消</span>
    </form>
    <!-- 搜索建议 -->
    <ul class="tt-search-suggest">
      <li class="tt-suggest-item">手机</li>
      <li class="tt-suggest-item">iPhone XS Max</li>
      <li class="tt-suggest-item">华为P30</li>
      <li class="tt-suggest-item">小米 MIX3</li>
      <li class="tt-suggest-item">诺基亚1110</li>
    </ul>
  </div>
</div>
```

这段代码放上以后，搜索框的样式就出来了。这里要注意下，我们在制作切图的时候，会把有状态切换的组件的切换方式都给出来，这样 JS 工程师才知道这个组件样式怎么转换，这里我们就用下面的代码标出了搜索组件在搜索状态中的 class 结构：

```
<!-- 带suggest的搜索框 -->
<div class="tt-search">
<!-- 搜索中的状态 -->
<!-- <div class="tt-search on-focus"> -->
```

这样搜索功能的样式也完成了。

分类导航的实现

下来我们实现文章的分类入口，这个可以使用没有边框的网格组件来实现。我们先放上网格组件的 HTML 代码，把下列代码添加到 tt-search 组件的后面：

```
<!-- 文章分类 -->
<div class="tt-grid tt-grid-4 no-border yp-category">
    <a href="./list.html" class="tt-grid-item item-1">
        <i class="fa fa-female tt-grid-icon"></i>
        <p class="tt-grid-label">女装</p>
    </a>
    <a href="./list.html" class="tt-grid-item item-2">
        <i class="fa fa-book tt-grid-icon"></i>
        <p class="tt-grid-label">图书</p>
    </a>
    <a href="./list.html" class="tt-grid-item item-3">
        <i class="fa fa-briefcase tt-grid-icon"></i>
        <p class="tt-grid-label">箱包</p>
    </a>
    <a href="./list.html" class="tt-grid-item item-4">
        <i class="fa fa-diamond tt-grid-icon"></i>
        <p class="tt-grid-label">珠宝首饰</p>
    </a>
        <a href="./list.html" class="tt-grid-item item-5">
        <i class="fa fa-soccer-ball-o tt-grid-icon"></i>
        <p class="tt-grid-label">体育用品</p>
    </a>
    <a href="./list.html" class="tt-grid-item item-6">
        <i class="fa fa-tv tt-grid-icon"></i>
        <p class="tt-grid-label">电视</p>
    </a>
    <a href="./list.html" class="tt-grid-item item-7">
        <i class="fa fa-mobile-phone tt-grid-icon"></i>
        <p class="tt-grid-label">手机</p>
    </a>
    <a href="./list.html" class="tt-grid-item item-8">
        <i class="fa fa-print tt-grid-icon"></i>
        <p class="tt-grid-label">打印设备</p>
    </a>
</div>
```

这段代码的样式会是下面这样：

这里我们换了图标的样式，另外这个组件还需要定义一些个性化的样式。比如作为入口，它每个格子的高度有些大了，另外全灰色的图标看起来也不美观，所以我们对它进行一些优化。在 /page/index.css 里加入如下代码，来修复这两个问题：

```
/* 缩小网格的高度 */
.page-index .tt-grid > .tt-grid-item{
    padding: .5rem 0;
}
/* 网格导航的颜色设置 */
.page-index .tt-grid > .tt-grid-item.item-1 > .tt-grid-icon{
    color: #f93c32;
}
.page-index .tt-grid > .tt-grid-item.item-2 > .tt-grid-icon{
    color: #1dbdd4;
}
.page-index .tt-grid > .tt-grid-item.item-3 > .tt-grid-icon{
    color: #eb1d57;
}
.page-index .tt-grid > .tt-grid-item.item-4 > .tt-grid-icon{
    color: #00bb4b;
}
.page-index .tt-grid > .tt-grid-item.item-5 > .tt-grid-icon{
    color: #fd9707;
}
.page-index .tt-grid > .tt-grid-item.item-6 > .tt-grid-icon{
    color: #fc5024;
}
.page-index .tt-grid > .tt-grid-item.item-7 > .tt-grid-icon{
    color: #9a00b0;
}
.page-index .tt-grid > .tt-grid-item.item-8 > .tt-grid-icon{
    color: #f3443c;
}
```

这样，分类导航的入口样式就完成了。现在分类入口的样式看起来就好的多了。

猜你喜欢列表的实现

从效果图上来看，猜你喜欢这块内容用的就是在Panel组件里放了一个复杂列表，我们先放上HTML代码。

```html
<!-- 猜你喜欢 -->
<h1 class="tt-panel-title">
  猜你喜欢
  <a href="./list.html" class="tt-panel-link">查看更多>></a>
</h1>
<div class="tt-panel-body no-padding">
  <div class="tt-list">
    <a href="./article.html" class="tt-list-item">
      <div class="item-img-wrap">
        <img class="item-img" src="img/list-img.jpg" alt="旅行必不可少">
      </div>
      <div class="item-content-wrap">
        <h1 class="item-title">大美新疆 | 玩遍这片大好山河，这些装备必不可少</h1>
        <p class="item-desc">发布时间：2019-8-18</p>
        <p class="item-desc">阅读1923次 | 收藏398次</p>
      </div>
    </a>
    <a href="./article.html" class="tt-list-item">
      <div class="item-img-wrap">
        <img class="item-img" src="img/list-img.jpg" alt="旅行必不可少">
      </div>
      <div class="item-content-wrap">
        <h1 class="item-title">大美新疆 | 玩遍这片大好山河，这些装备必不可少</h1>
        <p class="item-desc">发布时间：2019-8-18</p>
        <p class="item-desc">阅读1923次 | 收藏398次</p>
      </div>
    </a>
    <a href="./article.html" class="tt-list-item">
      <div class="item-img-wrap">
        <img class="item-img" src="img/list-img.jpg" alt="旅行必不可少">
      </div>
      <div class="item-content-wrap">
        <h1 class="item-title">大美新疆 | 玩遍这片大好山河，这些装备必不可少</h1>
        <p class="item-desc">发布时间：2019-8-18</p>
        <p class="item-desc">阅读1923次 | 收藏398次</p>
      </div>
    </a>
    <a href="./article.html" class="tt-list-item">
      <div class="item-img-wrap">
        <img class="item-img" src="img/list-img.jpg" alt="旅行必不可少">
      </div>
      <div class="item-content-wrap">
        <h1 class="item-title">大美新疆 | 玩遍这片大好山河，这些装备必不可少</h1>
        <p class="item-desc">发布时间：2019-8-18</p>
        <p class="item-desc">阅读1923次 | 收藏398次</p>
      </div>
    </a>
  </div>
</div>
```

这里面我们用到了一张图片，可以在page下建立img目录，然后把图片放进新建的目录里。

上面的 HMTL 中在 tt-panel-title 元素里放了一个“查看更多”的链接，但是 Panel 组件里没有实现“查看更多”这个功能，这又是一个比较常见的使用场景，所以我们准备在样式库中源码中加入这个样式。首先在 /src/content.css 的 tt-panel-title 样式的下面添加上下列代码：

```css
/* 标题导航 */
.tt-content .tt-panel-title > .tt-panel-link{
  float: right;
  margin-right: 1rem;
  color: #aaa;
}
```

因为之前已经发过一次版，所以在修改源文件的时候，要把组件升级一个版本。我们先在 README.md 文件中加入变更记录，最后这一章完结的时候再重新发一个版本。在 README.md 文件中追加下面的说明：

```
## 变更记录
0.1.1
- 【Initial】初始版本

0.1.2
- 【Add】Panel组件的标题部分添加“查看更多”这种引导链接的样式。
```

这样我们就能记住都改过什么内容了。这时候在项目根目录下执行一下打包命令，重新生成 dist 文件：

```
./shell/build.sh
```

这样查看更多的样式就生效了：

这里还有个小问题，就是猜你喜欢和上面分类的网格中间挨的太紧了，我们给分类入口加上一个下边距来撑开它俩之间的位置，在 /page/index.css 的开头位置加上下面的样式：

```
/* 调整间距 */
.page-index .yp-category{
    margin-bottom: .5rem;
}
```

这样看起来整个页面就和效果图很像了。

到这，我们首页的功能区就实现完了。但是经过测试，发现搜索框在搜索状态中的时候，样式会有错乱：

本该遮罩住全屏的搜索组件并没有生效，反而跑到主页元素后面去了。这是因为我们在开发样式库的时候没有给内容区添加过定位，这样的情况固定定位的元素可以覆盖住没有定位的内容区。但是这个页面里的组件有使用相对定位的，所以层级关系就不对了，我们来修复一下这个问题。

在 /src/search.css 文件中的 tt-search 选择器里加上“z-index: 110;”这条属性，并且在 README.md 文件中变更记录里追加上下面一段话：

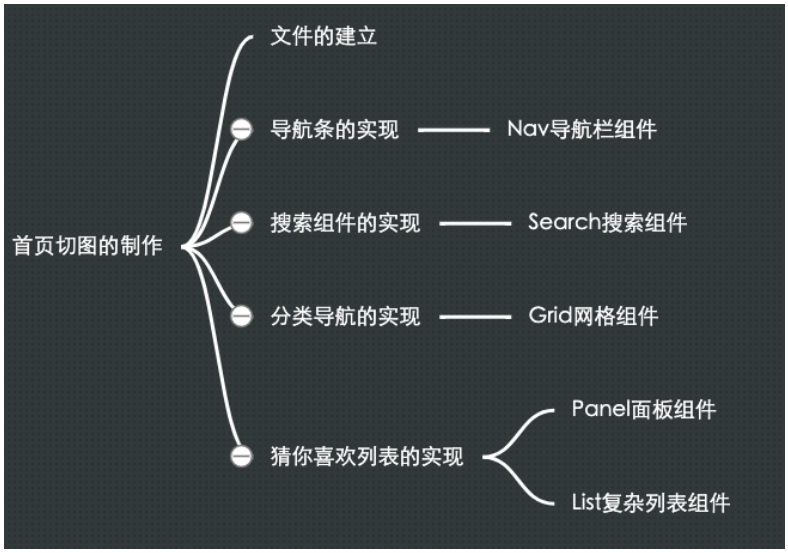
```
- 【Bug Fix】Search 组件指定 z-index，解决和有定位元素同时使用时的层级错乱问题。
```

这样再重新打包后，我们需要的搜索组件的样式就正常了。

结语

到这里我们这一节的内容就完成了。从头读起来这一节的内容好像有很多，但实际上 HTML 文件基本都是从 UI 库的 Demo 里复制过来进行简单的修改，而页面自己的样式文件只对三个地方做了简单的修改。这样就可以快速完成一个页面的切图了。

我们这一节内容的结构如下：



我们这一节的内容到这里就结束了，下一节将进入列表页的制作。

}

37 常用页面切图的设计与开发

39 列表页的设计与开发