

V 韩国代购粉红色的高峰时段华盛顿收到回复司机电话分机

Sdvsgshvbybvbybvbybvbybv 吧 v 吧 v 吧 v 吧 v 吧 v 吧 v 吧 v 吧 v 吧 v 吧 v
sdfshdfsfgjsdjjasoidhskjdhsjdhsjk

asdsajdgggggggggkCSDN 首页

博客

下载

学习

社区

GitCode

云服务

猿如意

pdfh5

搜索

发布

Vue 移动端网页（H5）预览 pdf 文件（pdfh5 和 vue-pdf）

置顶

theMuseCatcher

已于 2022-10-31 15:12:34 修改

17570

收藏 37

分类专栏： Vue css 文章标签： vue.js

版权

华为云开发者联盟

该内容已被华为云开发者联盟社区收录，社区免费抽大奖🎉，赢华为平板、Switch 等好礼！
加入社区

Vue

同时被 2 个专栏收录

37 篇文章 2 订阅

订阅专栏

css

30 篇文章 0 订阅

订阅专栏

一、使用插件 pdfh5 预览 pdf

参考文档：pdfh5 - npm

项目相关依赖版本信息

预览效果如下图所示：

1.上下滚动和缩放查看

2.左上角固定显示当前页码和总页数

3.右下角一键回到顶部按钮

4.在每页 pdf 上添加图片水印

①安装插件：yarn add pdfh5 (pdfh5@1.4.2)

②使用 pdfh5 插件预览 pdf

```
<template>
  <div class="m-pdf">
    <div id="pdf"></div>
  </div>
</template>
<script>
import Pdfh5 from 'pdfh5'
import 'pdfh5/css/pdfh5.css'
export default {
  name: 'Pdfh5',
  data () {
    return {
      pdfh5: null,
      // 可引入网络文件或者本地文件
      pdfUrl: '/Markdown.pdf' // 如果引入本地 pdf 文件，需要将 pdf 放在 public 文件夹
      // 下，引用时使用绝对路径 (/：表示 public 文件夹)
    }
  },
  mounted () {
    // pdfh5 实例化时传两个参数：selector 选择器，options 配置项参数，会返回一个 pdfh5
    // 实例对象，可以用来操作 pdf，监听相关事件
    // pdfh5 = new Pdfh5(selector, options) goto 初始到第几页，logo 设置每一页 pdf 上的
    // 水印
    this.pdfh5 = new Pdfh5('#pdf', { pdfurl: this.pdfUrl, goto: 1, logo: { src:
    require('images/logo.png'), x: 420, y: 700, width: 120, height: 120 } })
    // 监听 pdf 准备开始渲染，此时可以拿到 pdf 总页数
```

```

    this.pdfh5.on('ready', function () {
      console.log('总页数:' + this.totalNum)
    })
    // 监听 pdf 加载完成事件，加载失败、渲染成功都会触发
    this.pdfh5.on('complete', (status, msg, time) => {
      console.log('状态:' + status + ', 信息:' + msg + ', 耗时:' + time + '毫秒')
    })
  }
}
</script>
<style lang="less" scoped>
.m-pdf { // 保证 pdf 区域铺满整个屏幕
  // 方法 1: 使用 vw 和 vh 视窗单位, 1vw=视窗宽度的 1%; 1vh=视窗高度的 1%
  // width: 100vw;
  // height: 100vh;
  // 方法 2: 使用 fixed 定位
  position: fixed;
  top: 0;
  bottom: 0;
  right: 0;
  left: 0;
}
</style>

```

二、使用插件 vue-pdf 预览 pdf

相关参考文档: vue-pdf - npm

①安装插件: yarn add vue-pdf (@vue-pdf@4.3.0)

②使用 vue-pdf 插件在移动端网页进行 pdf 预览

一次性加载全部 pdf, 上下滚动查看 pdf (初次加载较慢, 交互体验较差)

```

<template>
  <div class="m-vue-pdf">
    <div v-show="loaded">
      <Pdf v-for="index in numPages" :key="index" :src="pdfUrl" :page="index" />
    </div>
  </div>
</template>
<script>
import Pdf from 'vue-pdf'
export default {
  name: 'VuePdf',
  components: {

```

```

    Pdf
  },
  data () {
    return {
      loaded: false,
      numPages: null,
      // 可引入网络文件或者本地文件
      pdfUrl: '/Markdown.pdf' // 如果引入本地 pdf 文件，需要将 pdf 放在 public 文件夹
      下，引用时使用绝对路径 (/：表示 public 文件夹)
    }
  },
  created () {
    this.loadPdf()
  },
  methods: {
    // 上下滚动 pdf 加载
    loadPdf () {
      this.pdfUrl = Pdf.createLoadingTask(this.pdfUrl)
      this.pdfUrl.promise.then(pdf => {
        this.$nextTick(() => {
          this.numPages = pdf.numPages // pdf 总页数
          this.loaded = true
        })
      })
    }
  }
}
</script>
<style lang="less" scoped>
.m-vue-pdf {
  overflow: hidden;
}
</style>

```

2.分页加载预览 pdf，左上角自定义显示当前页码和总页数，通过监听手指上下(或左右)滑动进行翻页

效果如下图所示：

```

<template>
  <div class="m-vue-pdf">
    <div class="u-page" v-show="loaded">
      {{currentPage}}/{{totalCount}}
    </div>
  </div>
</template>

```

```

</div>
<Pdf
  :src="pdfUrl"
  :page="currentPage"
  @num-pages="totalCount=$event"
  @page-loaded="currentPage=$event"
  @loaded="loadPdfHandler"
/>
</div>
</template>
<script>
import Pdf from 'vue-pdf'
export default {
  name: 'VuePdfPaging',
  components: {
    Pdf
  },
  data () {
    return {
      loaded: false,
      rootSize: 0,
      startx: 0,
      starty: 0,
      endx: 0,
      endy: 0,
      currentPage: 1, // pdf 文件页码
      totalCount: 0, // pdf 文件总页数
      // 可引入网络文件或者本地文件
      pdfUrl: '/Markdown.pdf' // 如果引入本地 pdf 文件，需要将 pdf 放在 public 文件夹
    }
  },
  created () {
    var docEl = window.document.documentElement // 即<html></html>标签
    var width = window.screen.availWidth || window.screen.width
    this.rootSize = width / 15 // 设置 rem 相对单位大小
    docEl.style.fontSize = this.rootSize + 'px'
  },
  mounted () {
    document.body.addEventListener('touchmove', this.touchMove, { passive: false }) // 禁用 H5 页面的下拉刷新
    document.body.addEventListener('touchstart', this.touchStart, false) // true: 表示在捕获阶段调用事件处理程序; false: 表示在冒泡阶段调用事件处理程序
    document.body.addEventListener('touchend', this.touchEnd, false)
  }
}

```

```

},
methods: {
  touchMove (e) {
    e.preventDefault()
    this.$once('hook:beforeDestroy', function () {
      removeEventListener('touchmove', this.touchMove)
    })
  },
  touchStart (e) {
    console.log('start:', e)
    this.startx = e.touches[0].pageX
    this.starty = e.touches[0].pageY
    console.log('start:', this.startx, this.starty)
    this.$once('hook:beforeDestroy', function () {
      removeEventListener('touchstart', this.touchStart)
    })
  },
  touchEnd (e) {
    console.log('end:', e)
    this.endx = e.changedTouches[0].pageX
    this.endy = e.changedTouches[0].pageY
    console.log('end:', this.endx, this.endy)
    // y 方向上 (或 x 方向上) 滑动超过 50px 进行翻页
    if ((this.starty - this.endy) / this.rootSize > 5 || (this.startx - this.endx) / this.rootSize >
5) {
      this.nextPage()
    }
    if ((this.endy - this.starty) / this.rootSize > 5 || (this.endx - this.startx) / this.rootSize >
5) {
      this.prePage()
    }
    this.$once('hook:beforeDestroy', function () {
      removeEventListener('touchend', this.touchEnd)
    })
  },
  prePage () {
    if (this.currentPage > 1) {
      this.currentPage--
    }
  },
  nextPage () {
    if (this.currentPage < this.totalCount) {
      this.currentPage++
    }
  }
}

```

```

    },
    // pdf 加载时
    loadPdfHandler (e) {
      this.currentPage = 1 // 加载的时候先加载第一页
      this.loaded = true
    }
  }
}

```

```

</script>

```

```

<style lang="less" scoped>

```

```

.m-vue-pdf {
  overflow: hidden;
  .u-page {
    position: fixed;
    left: 2rem;
    top: 1rem;
    z-index: 999;
    font-size: 1.5rem;
    color: #FFF;
    padding: 0.2rem 0.8rem;
    border-radius: 0.64rem;
    background: rgba(0, 0, 0, 0.5);
    letter-spacing: 0.1rem;
  }
}

```

```

</style>

```

3.采用分页预览 pdf，包括自定义跳转到指定页数，上一页，下一页按钮（初次加载更快，交互体验更好）

效果如下图：

```

<template>
  <div class="m-vue-pdf" v-show="loaded">
    <div class="m-page">
      <div class="m-left">
        <span class="u-label">跳至</span>
        <input v-model='jumpNum' type="number" class="u-input" />
        <span class="u-unit">页</span>
        <a class="u-button" @click="onJump">确定</a>
        <div class="assist"></div>
      </div>
      <div class="m-right">

```

```

<svg viewBox="64 64 896 896" data-icon="left" :class="['u-arrow', { gray:
currentPage===1 }]" @click="onPrePage" aria-hidden="true" focusable="false"><path
d="M724 218.3V141c0-6.7-7.7-10.4-12.9-6.3L260.3 486.8a31.86 31.86 0 0 0 50.3l450.8
352.1c5.3 4.1 12.9 4 12.9-6.3v-77.3c0-4.9-2.3-9.6-6.1-12.6l-360-281 360-281.1c3.8-3 6.1-
7.7 6.1-12.6z"></path></svg>
<span class="u-num">{{currentPage}}/{{totalCount}}</span>
<svg :class="['u-arrow', { gray: currentPage===totalCount }]"
@click="onNextPage" viewBox="64 64 896 896" data-icon="right" aria-hidden="true"
focusable="false"><path d="M765.7 486.8L314.9 134.7A7.97 7.97 0 0 0 302 141v77.3c0 4.9
2.3 9.6 6.1 12.6l360 281.1-360 281.1c-3.9 3-6.1 7.7-6.1 12.6V883c0 6.7 7.7 10.4 12.9
6.3l450.8-352.1a31.96 31.96 0 0 0 50.4z"></path></svg>
<div class="assist"></div>
</div>
</div>
<Pdf
:src="pdfUrl"
:page="currentPage"
@num-pages="totalCount=$event"
@page-loaded="currentPage=$event"
@loaded="loadPdfHandler">
</Pdf>
</div>
</template>
<script>
import Pdf from 'vue-pdf'
export default {
  name: 'VuePdfJump',
  components: {
    Pdf
  },
  data () {
    return {
      loaded: false,
      numPages: '',
      currentPage: 1, // pdf 文件页码
      totalCount: 0, // pdf 文件总页数
      jumpNum: null,
      // 可引入网络文件或者本地文件
      pdfUrl: '/Markdown.pdf' // 如果引入本地 pdf 文件，需要将 pdf 放在 public 文件夹
      下，引用时使用绝对路径 (/: 表示 public 文件夹)
    }
  },
  created () {
    var docEl = window.document.documentElement

```



```

var width = window.screen.availWidth || window.screen.width
var rootSize = width / 15
docEl.style.fontSize = rootSize + 'px'
},
methods: {
  onJump () {
    if (this.jumpNum) {
      if (Number(this.jumpNum) <= this.totalCount && Number(this.jumpNum) >= 1) {
        this.currentPage = Number(this.jumpNum)
        this.jumpNum = null
      } else {
        this.$toast('请输入正确的跳转页码', this.toastParams)
      }
    } else {
      this.$toast('请输入要跳转的页码', this.toastParams)
    }
  },
  onPrePage () {
    if (this.currentPage > 1) {
      this.currentPage--
    }
  },
  onNextPage () {
    if (this.currentPage < this.totalCount) {
      this.currentPage++
    }
  },
  // pdf 加载时
  loadPdfHandler (e) {
    this.currentPage = 1 // 加载的时候先加载第一页
    this.loaded = true
  }
}
}
</script>
<style lang="less" scoped>
.m-vue-pdf {
  overflow: hidden;
  .m-page {
    position: sticky;
    top: 0;
    left: 0;
    right: 0;
    height: 1.5rem;
  }
}

```