

上传组件

前端

el-upload标签

- :http-request中指定上传处理方法
- img标签的src属性绑定图片完整地址变量（计算属性）

```
<el-upload
  class="avatar-uploader"
  action="#"
  :show-file-list="false"
  :http-request="beforeUploadPhotoUpload"  上传方法
>
  
  <i v-else class="el-icon-plus avatar-uploader-icon" />  图片地址计算变量
```

引用fastdfs地址变量

```
<script>
import { validMobile, validIdcard } from '@utils/my-validate'
import { getDict } from '@utils/dict'
import DictSelect from '@components/DictSelect'
import settings from '@settings'  fastdfs地址所在变量
```

data中定义返回地址变量（不带ip的，用于提交到实体类）

```
data() {
  return {
    initFlag: false,
    active: 0,
    buttonLoading: false,
    screenWidth: 0,
    width: this.initWidth(),
    formObj: this.initRegister(),
```

```
methods: {
  validAddress(rule, value, callback) {...},
  initWidth() {...},
  next() {...},
  close() {...},
  initRegister() {
    return {
      uploadPhotoUrl: '',
      comment: ''
    }
  },
}
```

计算属性:定义图片fastdfs全路径

```
computed: {
  isVisible: {...},
  user() {...},
  uploadPhotoUrl() {
    return settings.pages.fastUrl + this.formObj.uploadPhotoUrl
  },
}
```

定义上传处理方法

```

methods: {
  validateAddress(rule, value, callback) {...},
  initWidth() {...},
  getHouse() {...},
  close() {...},
  initRegister() {...},
  setRegister(val) {...},
  submitForm() {...},
  reset() {...},
  beforeUploadPhotoUpload(params) {
    const file = params.file
    // 上传前校验类型、大小等
    // const fileType = file.type...
    const fd = new FormData()
    fd.append('file', file)
    const that = this
    this.$upload('fastdfs/upload', fd).then((res) => {
      that.formObj.uploadPhotoUrl = res.data.data.fileAddress
      this.$message({type: 'success'...})
    })
  }
}
}

```

后端

FdfsController

```

@PostMapping("/upload")
public FebsResponse upload(MultipartFile file) throws IOException {

```