

管理员页面

1. 使用bootstrap

安装: `npm i bootstrap@3`

安装bootstrap插件

multer文件上传

1. muler是一个中间件

2. 安装: `npm i multer -S`

3. 基本用法:

```
const multer = require('multer');
const upload = multer({
  dest: 'public/images'
})
router.post('/open-courses',
  [upload.single('file')],
  async (req, res, next) => {
    // 上传的文件在这里
    req.file
  }) ;
```

4. 上传文件格式和大小约束

```
const upload = multer({
  limits: {fileSize: 2 * 1024 * 1024}, //最大2M
  fileFilter: function (req, file, cb) {
    console.log(file);
    // 判断文件是否合法, 合法则处理, 不合法则拒绝
    if (file.mimetype === 'image/gif' ||
      file.mimetype === 'image/jpeg' ||
      file.mimetype === 'image/png') {
      // 接收文件
      cb(null, true);
    } else {
      cb(new Error('请上传图片格式'), false);
    }
  }
})
```

5. 自定义存储目录和文件名

```
const storage = multer.diskStorage({
  destination: function (req, file, cb) { // 存储目录
    cb(null, 'public/images');
  },
```

```
filename: function (req, file, cb) {
  let extname = '';
  switch (file.mimetype) {
    case 'image/jpeg':
      extname = '.jpg';
      break;
    case 'image/png':
      extname = '.png';
      break;
    case 'image/gif':
      extname = '.gif';
      break;
  }
  cb(null, Date.now() + extname);
}
}))
const upload = multer({
  storage,
})
```

数据校验

1. express-validator: npm i -S express-validator

2. 基本使用:

1) 定义校验函数

```
const {body, validationResult} = require('express-validator/check');
const validateFn = body('name').not().isEmpty().withMessage('公开课名称必填')
```

2) 应用校验

```
router.post('/open-courses',
  [upload.single('file'), validateFn1, validateFn2, ...], // 执行多个中间件, 用数组形式
  async (req, res, next) => {
```

3) 错误检测

```
let errors = validationResult(req);
if(errors.isEmpty()){
  //没错误, 校验通过
}else{
  //校验失败
  //错误信息格式化
  errors = errors.formatWith((error) => error.msg);
}
```

sql语句生成

```
const mysql = require('mysql');  
sql = mysql.format(`INSERT INTO open_course SET ?`, req.body);
```

数组结构

```
const validations = [b,c,d]  
const middlewares = [a, ...validations] // [a,b,c,d]
```

函数形参结构

```
// formatWith形参形如: {msg:'',value:'',....}  
errors.formatWith(({msg}) => msg)
```

作业:

1. 复习与吸收之前知识点
2. 预习angular:<https://angular.cn/>