

勘误:

1. 路由守卫中调用login()报错: 提示该方法返回值和当前函数需要的不匹配
2. video管理页面问题

性能优化

1. 前端优化点: 模块异步加载

- 懒加载
 - app-routing添加异步加载

```
{path: 'user', loadChildren: './user/user.module#UserModule'}
```

- 子模块路由模块去除根路由path:"

```
{  
  path: '', component: UserComponent, children: [...]  
}
```

- app模块中相关导入删除掉

```
imports: [  
  // UserModule,  
]
```

- 模块预加载
 - 路由模块配置

```
RouterModule.forRoot(routes, {  
  preloadingStrategy: SelectivePreloadingStrategy  
})
```

- 路由配置:

```
{  
  path: 'main',  
  loadChildren: './main/main.module#MainModule',  
  data: {preload: true}  
}
```

- 策略实现:

```
export class SelectivePreloadingStrategy implements PreloadingStrategy {
  preload(route: Route, load: () => Observable<any>): Observable<any> {
    if (route.data && route.data.preload) {
      return load();
    } else {
      return of(null);
    }
  }
}
```

2. 后端优化点：静态资源缓存、响应体压缩、环境变量的重要性

- 静态资源缓存，设置static中间件maxAge选项

```
app.use(express.static(path.join(__dirname, 'public'), {
  maxAge: 86400 // 静态文件缓存
})));
```

- 响应体压缩 (gzip)
 - 安装compression模块：npm i compression -S
 - 使用

```
const compression = require('compression')
app.use(compression());
```

- 环境变量设置：NODE_ENV=production
 - 程序中修改：process.env.NODE_ENV = 'production'
 - 启动脚本中修改：
 - linux: export NODE_ENV=production
 - win: set NODE_ENV=production
 - 跨平台：安装cross-env模块：npm i cross-env -D
cross-env NODE_ENV=production
- 尽可能避免同步方法
 - 异步文件日志记录[winstonjs](#)

项目发布：

- 发布：npm run build 或 ng build
 - 添加--prod选项可以生成生产环境配置
 - 修改index.html中可以修改上下文地址，则将来访问地址变为：localhost:8080/student/index.html
- 子域名配置：
 - 通过vhost模块

- nginx反向代理:

- 安装nginx: [下载](#)
- 配置nginx:

```
server {  
    listen      80;  
    server_name www.mykaikeba.com;  
    #charset koi8-r;  
    #access_log logs/host.access.log main;  
    location / {  
        proxy_pass http://127.0.0.1:8080;  
    }  
}  
  
server {  
    listen      80;  
    server_name student.mykaikeba.com;  
    #charset koi8-r;  
    #access_log logs/host.access.log main;  
    location / {  
        proxy_pass http://127.0.0.1:8081;  
    }  
}
```

- 配置host, 配两个域名

127.0.0.1 [www.mykaikeba.com](#) 127.0.0.1 student.mykaikeba.com

- nginx启停:

启动: 进入nginx目录, start nginx.exe 或 nginx.exe

推荐前者, 命令行处于活动状态, 还可以进行其他操作

停止: nginx.exe -s stop 或 nginx.exe -s quit

重新载入: nginx.exe -s reload

- spa刷新404问题

- 在student服务器项目中安装模块: npm i -S connect-history-api-fallback
- 使用:

```
const history = require('connect-history-api-fallback');  
app.use(history());
```

注意在static前面

部署: 阿里云ECS, windows

- 数据库、node等环境准备
- 文件上传ftp
 - 阿里云控制台开启21端口: 添加安全组
 - 服务器安装ftp服务

- 服务器添加ftp用户
 - 客户端连接并上传
- pm2: <http://pm2.keymetrics.io>
 - 安装: `npm i -g pm2`
 - 启动项目: `pm2 start ./bin/www -n appName`
 - 停止项目: `pm2 stop appName`
 - 重启项目: `pm2 restart appName`
 - 重新加载: `pm2 reload appName`
- 配置反向代理