#### Angular脱坑记

##### 之拦截器

问题描述

Angular 6.1.5

HTTP请求拦截器（HttpInterceptor）是Angular 6的一个特性，可用于：

1. 统一报错拦截（pipelining）
2. 身份验证
3. 请求缓存

解决

1. 填充HttpInterceptor Interface

responseStatusFilter.ts

@Injectable()  
export class ResponseStatusFilter implements HttpInterceptor {  
 constructor(  
 private missionService: MissionService  
 ) { }  
  
 intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {  
 if (!environment.enableInterceptors) {  
 return next.handle(req);  
 }  
 if (req.url.includes("/web/rest/dis/") || req.url.includes("/api/vmaster-pms/v1/")) {  
 return next.handle(req).do((event: HttpEvent<any>) => {  
 if (event instanceof HttpResponse) {  
 if (event.status === 200) {  
 // Success.  
 console.info("[ResponseStatusFilter] intercept(): url=", req.url);  
 console.info("[ResponseStatusFilter] intercept(): method=", req.method);  
 console.info("[ResponseStatusFilter] intercept(): body=", req.body);  
 }  
 }  
 }, (err: any) => {  
 if (err instanceof HttpErrorResponse) {  
 if (err.status >= 400) {  
 this.missionService.alertConfirm({  
 type: "error",  
 message: err.message  
 });  
 }  
 }  
 });  
 }  
  
 return next.handle(req);  
  
 }  
}

1. 在app module中挂载拦截器

app.module.ts

providers: [  
 {  
 provide: HTTP\_INTERCEPTORS,  
 useClass: ResponseStatusFilter,  
 multi: true  
 }],

1. 效果

