

【Egg.js 中间件进阶】 router.js 中使用中间件、框架默认中间件、Egg 中使用 Koa 中间件、中间件的通用配置

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、定义一个中间件在应用中使用中间件.....	1
二、在 router.js 路由中使用中间件.....	2
三、框架默认中间件的配置.....	3
四、Egg.js 中使用 koa 的中间件 （规范的 Koa 的中间件）	3
五、Egg.js 中使用 koa 的中间件 （非规范的 Koa 中间件）	4
六、Egg.js 中间件的通用配置.....	5

一、定义一个中间件在应用中使用中间件

```
module.exports=(option,app)=>{  
  
  return async function auth(ctx,next){  
  
    console.log(option);  
  
    console.log(new Date());  
  
    await next();  
  
  }  
}
```

```
}
```

- 1、新建 middleware/auth.js
- 2、在 config.default.js 中配置中间件

```
config.middleware = ['auth'];  
  
config.auth={  
  attr:'this is  attr'  
}
```

二、在 router.js 路由中使用中间件

- 1、新建 middleware/auth.js

```
module.exports=(option,app)=>{  
  return async function auth(ctx,next){  
    console.log(option);  
    console.log(new Date());  
    await next();  
  }  
}
```

2、router.js 挂载指定的中间件

```
module.exports = app => {  
  
  const { router, controller } = app;  
  
  //路由中获取中间件  
  
  const auth = app.middleware.auth({ attr: 'this is router.js middleware' });  
  
  router.get('/', auth, controller.home.index);  
  
  router.get('/news', controller.news.index);  
  
  router.get('/shop', controller.shop.index);  
  
};
```

三、框架默认中间件的配置

```
config.bodyParser={  
  jsonLimit: '10mb'    //Default is 1mb.  
}
```

四、Egg.js 中使用 koa 的中间件 （规范的 Koa 的中间件）

在 Egg.js 框架里面可以非常容易的引入 Koa 中间件生态。

使用 koa-compress 开启服务器 Gzip 压缩功能，以 koa-compress 为例，在 Koa 中使用的方法：

```
const koa = require('koa');
const compress = require('koa-compress');

const app = koa();

const options = { threshold: 2048 };
app.use(compress(options));
```

我们按照框架的规范来在应用中加载这个 Koa 的中间件：

```
// app/middleware/compress.js
// koa-compress 暴露的接口(`(options) => middleware`)和框架对中间件要求一致

module.exports = require('koa-compress');
```

```
config.middleware = ['compress'];
config.compress = {
  threshold: 1024 //它支持指定只有当 body 大于配置的 threshold 时才进行 gzip 压缩
};
```

五、Egg.js 中使用 koa 的中间件（非规范的 Koa 中间件）

```
// config/config.default.js
module.exports = {
  webpack: {
    compiler: {},
    others: {},
  },
};

// app/middleware/webpack.js

const webpackMiddleware = require('some-koa-middleware');

module.exports = (options, app) => {
  return webpackMiddleware(options.compiler, options.others);
}
```

六、Egg.js 中间件的通用配置

无论是应用层加载的中间件还是框架自带中间件，都支持几个通用的配置项：

enable: 控制中间件是否开启。

match: 设置只有符合某些规则的请求才会经过这个中间件。

ignore: 设置符合某些规则的请求不经过这个中间件。

<https://eggjs.org/zh-cn/basics/middleware.html>