

koa 中间件

主讲教师：（大地）

合作网站：<http://www.itying.com/>

中文文档：<http://www.itying.com/koa>

目录

一、 什么是 Koa 的中间件.....	1
二、 Koa 应用可使用如下几种中间件：	2
三、 Koa 中间件的执行顺序.....	4

一、什么是 Koa 的中间件

通俗的讲：中间件就是匹配路由之前或者匹配路由完成做的一系列的操作，我们就可以把它叫做中间件。

在 **express 中间件(Middleware)** 是一个函数，它可以访问请求对象(request object (req)), 响应对象 (response object (res)), 和 web 应用中处理请求-响应循环流程中的中间件，一般被命名为 **next** 的变量。在 Koa 中中间件和 **express** 有点类似。

中间件的功能包括：

- 执行任何代码。
- 修改请求和响应对象。
- 终结请求-响应循环。
- 调用堆栈中的下一个中间件。

如果我的 **get**、**post** 回调函数中，没有 **next** 参数，那么就匹配上第一个路由，就不会往下匹配了。如果想往下匹配的话，那么需要写 **next()**

二、Koa 应用可使用如下几种中间件:

应用级中间件
路由级中间件
错误处理中间件
第三方中间件

1. 应用级中间件

```
const Koa = require('koa');
const Router = require('koa-router');

const app = new Koa();
const router = new Router();
app.use(async (ctx, next) => {
  console.log(new Date());
  await next();
})
router.get('/', function (ctx, next) {
  ctx.body = "Hello koa";
})
router.get('/news', (ctx, next) => {
  ctx.body = "新闻页面"
});
app.use(router.routes()); //作用: 启动路由
app.use(router.allowedMethods()); //作用: 当请求出错时的处理逻辑
app.listen(3000, () => {
  console.log('starting at port 3000');
});
```

2. 路由中间件

```
router.get('/', async(ctx, next)=>{
  console.log(1)
  next()
})
router.get('/', function (ctx) {
  ctx.body="Hello koa";
})
```

3、错误处理中间件

```
app.use(async (ctx,next)=> {
  next();
  if(ctx.status==404){
    ctx.status = 404;
    ctx.body="这是一个 404 页面"
  }
});
```

4、第三方中间件

```
const static = require('koa-static');
const staticPath = './static';
app.use(static(
  path.join( __dirname, staticPath)
))

const bodyParser = require('koa-bodyparser');
app.use(bodyParser());
```

三、Koa 中间件的执行顺序

Koa 的中间件和 Express 不同，Koa 选择了洋葱圈模型。

中间件洋葱图：

