

# Node.js02

---



- 课堂目标

- 1、koa使用
- 2、了解koa中的application对象及context对象
- 3、koa-views使用
- 4、koa-static使用
- 5、使用koa-router中间件搭建路由
- 6、了解常见http状态码
- 7、restful接口设计

- 本节知识点

- http模块实现模板加载及静态文件处理
- koa的安装及使用
- Application及context对象介绍
- 常用http状态码介绍
- 路由中间件koa-router介绍
- 中间件koa-views介绍

## koa介绍

---

koa是express原班人马打造的轻量、健壮、富有表现力的nodejs框架。目前koa有koa1和koa2两个版本；koa2依赖Node.js 7.6.0或者更高版本；koa不在内核方法中绑定任何中间件，它仅仅是一个轻量级的函数库，几乎所有功能都必须通过第三方插件来实现。

## koa使用

- koa安装
  - \$ npm i koa
- 一个简单的koa服务器

```
const Koa = require('koa');
const app = new Koa();

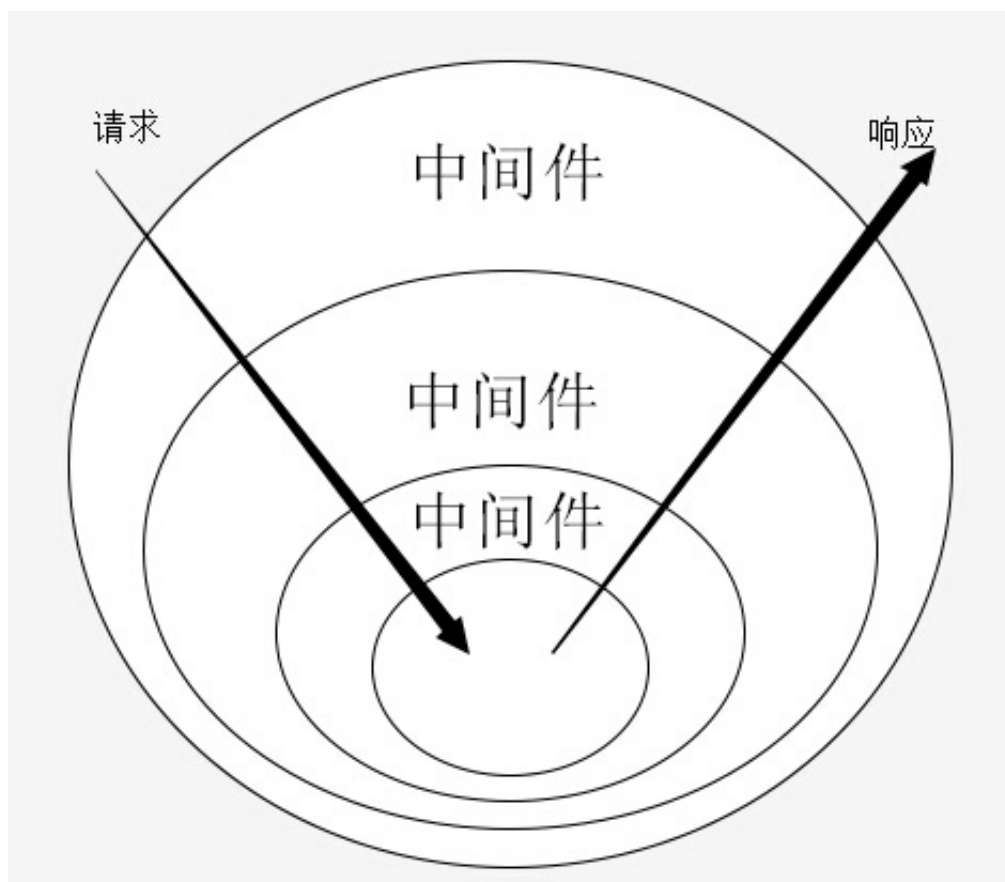
app.use(async ctx => {
  ctx.body = 'Hello World';
});

app.listen(3000);
```

- Koa 利用中间件 控制"上游", 调用"下游";
  - koa是包含一组中间件函数的对象；可以将app.use里的函数理解成中间件

```
//这里的middleWare函数就是一个中间件
let middleWare = async (ctx,next)=>{
  console.log("first middleWare");
  ctx.body = "hello world";
}
app.use(middleWare);
```

- 通过next()将控制转交给另一个中间件；
- 上述过程也可以通过"洋葱模型"来解释中间件执行顺序



**Application对象**

- application是koa的实例 简写app
- app.use 将给定的中间件方法添加到此应用程序,分为同步和异步，异步：通过es7中的async和await来处理
- app.listen设置服务器端口；
- app.on 错误处理；

## 上下文context对象常用属性及方法

- context 将node中的request和response 封装到一个对象中，并提供一些新的api提供给用户进行操作；
  - ctx.app:应用程序实例引用,等同于app;
  - ctx.req:Node 的 `request` 对象.
  - ctx.res:Node 的 `response` 对象.
  - ctx.request:koa中的Request对象；
  - ctx.response:koa中的response对象；
  - ctx.state: 对象命名空间，通过中间件传递信息；
  - ctx.throw:抛出错误；
- request及response别名
  - koa会把ctx.request上的属性直接挂载到ctx上如：
    - `ctx.header` //头信息；
    - `ctx.headers`
    - `ctx.method`
    - `ctx.method=`
    - `ctx.url`
    - `ctx.url=`

.....
  - 同样也会把ctx.response上的属性直接挂载到ctx上如：
    - `ctx.body`
    - `ctx.body=`
    - `ctx.status`
    - `ctx.status=`

....
  - ctx.status 获取响应状态。默认情况下，`response.status` 设置为 `404` 而不是像 node 的 `res.statusCode` 那样默认为 `200`。
  - http状态码：1xx(消息)、2xx(成功)、3xx(重定向)、4xx(请求错误)、5xx和6xx(服务器错误)
  - 常见http状态码 （302 location 跳转）

HTTP状态码	描述
100	继续。继续响应剩余部分，进行提交请求
200	成功
301	永久移动。请求资源永久移动到新位置
302	临时移动。请求资源零时移动到新位置
304	未修改。请求资源对比上次未被修改，响应中不包含资源内容
401	未授权，需要身份验证
403	禁止。请求被拒绝
404	未找到，服务器未找到需要资源
500	服务器内部错误。服务器遇到错误，无法完成请求
503	服务器不可用。零时服务过载，无法处理请求

## koa常用中间件介绍

### 一、koa-router

- 路由是引导匹配之意，是匹配url到相应处理程序的活动。
- koa-router安装

```
npm i koa-router -S
```
- Koa-router使用
- Koa-router推荐使用RESTful架构API。Restful的全称是Representational State Transfer 即表现层转移。
  - RESTful是一种软件架构风格、设计风格，而不是标准，只是提供了一组设计原则和约束条件。基于这个风格设计可以更简洁，更有层次；
  - 非RESTful架构api：
  - 使用RESTful架构设计api
  - REST设计一般符合如下条件：
    - 程序或者应用的事物都应该被抽象为资源
    - 每个资源对应唯一的URI(uri是统一资源标识符)
    - 使用统一接口对资源进行操作
    - 对资源的各种操作不会改变资源标识
    - 所有操作都是无状态的

### 二、koa-views

- Koa-views用于加载html模板文件；

- 安装 koa-views

```
npm i koa-views -S
```

- 使用koa-view

### 三、koa-static

- koa-static 是用于加载静态资源的中间件，通过它可以加载css、js等静态资源；
- 安装 koa-static

```
npm i koa-static
```

- 使用koa-static

```
const static = require("koa-static");  
app.use(static(__dirname+"/static")) //加载静态文件的目录
```

### 四、koa实现新闻列表

- 新闻列表页面
- 新闻详细页面
- 分页的实现

### 下期预告

- mysql数据库