复习：

主要是结合 express + mysql + art-template 完成个人博客作品

后端管理：管理所需的数据

前端展示：借助 ajax + art-template 实现作品的制作

结合 MVC 设计模式，优化项目结构

M：Model 模型层 --> 模型数据的处理

V：View 显示层 --> 就是对应的前端模板页面 html

C：Controller 控制层 --> 通过模型层调用模型层中的方法获取模型数据，再来将显示层跟数据进行绑定，实现view上的模板渲染工作

还没讲的知识点：

Koa MongoDB Linux(winScp) webpack Vue Less Sass 微信小程序 / 微信公众号 React php 简历 / 面试

1. koa

是什么：是一个用于快速搭建 web 服务的框架；是跟 express 框架同一个开发团队开发出来的下一代 nodejs web 服务框架

为什么：在express基础上，做出了优化

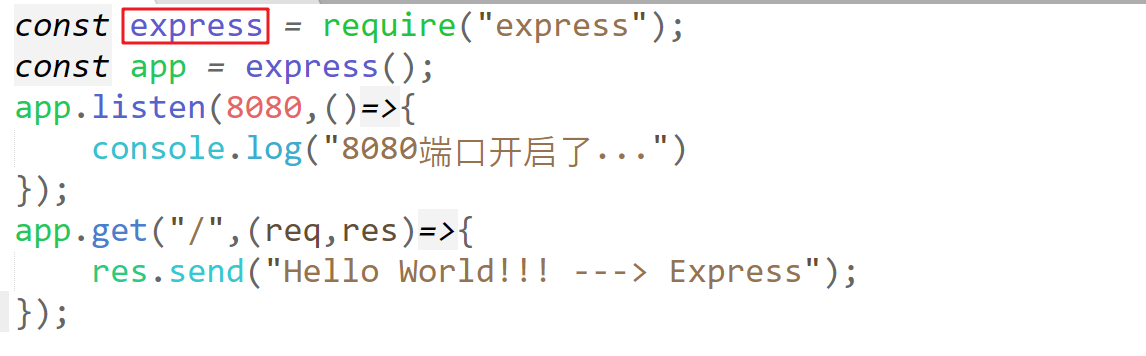
怎么学：因为是在express基础上，提出的新框架，所以在 koa 中随处可见 express 框架的影子

官网：<https://koajs.com/>

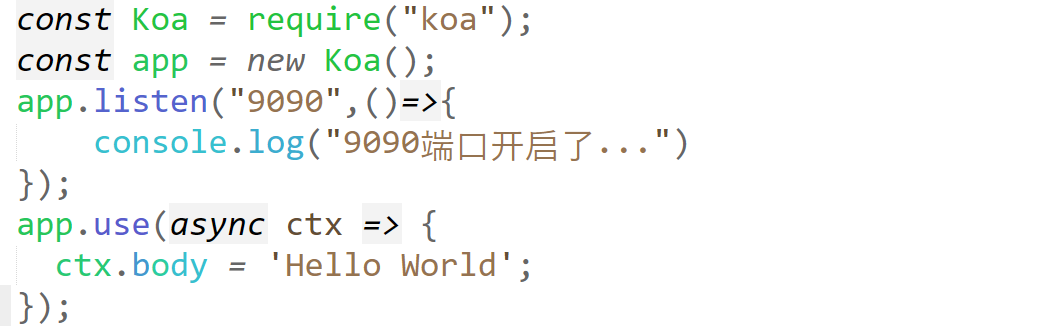
中文官网：<https://koa.bootcss.com/>

2. 通过最简单的案例，来接触 koa ，并比较 express 跟 koa 的差别

对于 express 的实现：



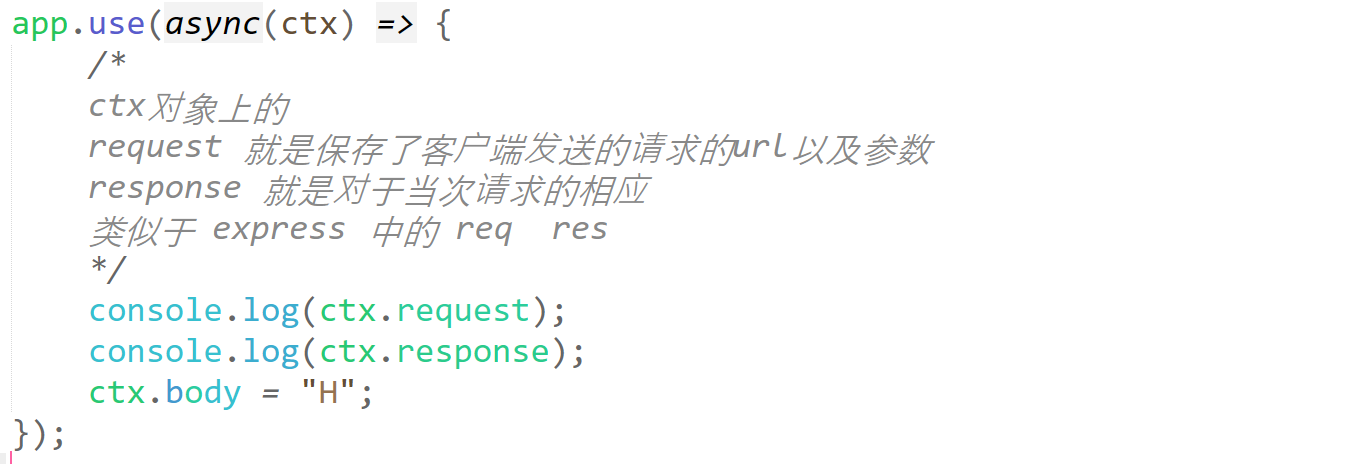
对于 koa 的实现：



3. 对于 koa 中的一系列参数以及一系列常用的外部库的介绍

a>> 跟 express 相比较而言，通过Koa创建出的app对象，不提供 get / post 方法，而是将这些方法强行封装到了 router 中

b>> 在 koa 中对于请求的获取上，提供了一个参数，这个参数叫 context 上下文，这个对象上包含了之前的 res res 这些参数



c>> 结合 ctx 来获取 get 请求传递过来的参数

目的 写法 简洁写法

获取url ctx.request.url ctx.url

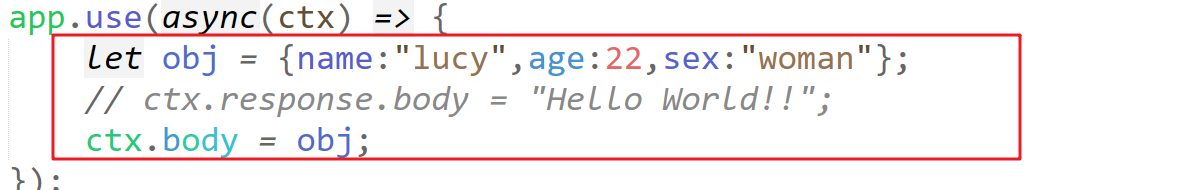
获取参数 ctx.request.query ctx.query

获取参数 ctx.request.querystring ctx.querystring

对于请求后的响应的操作

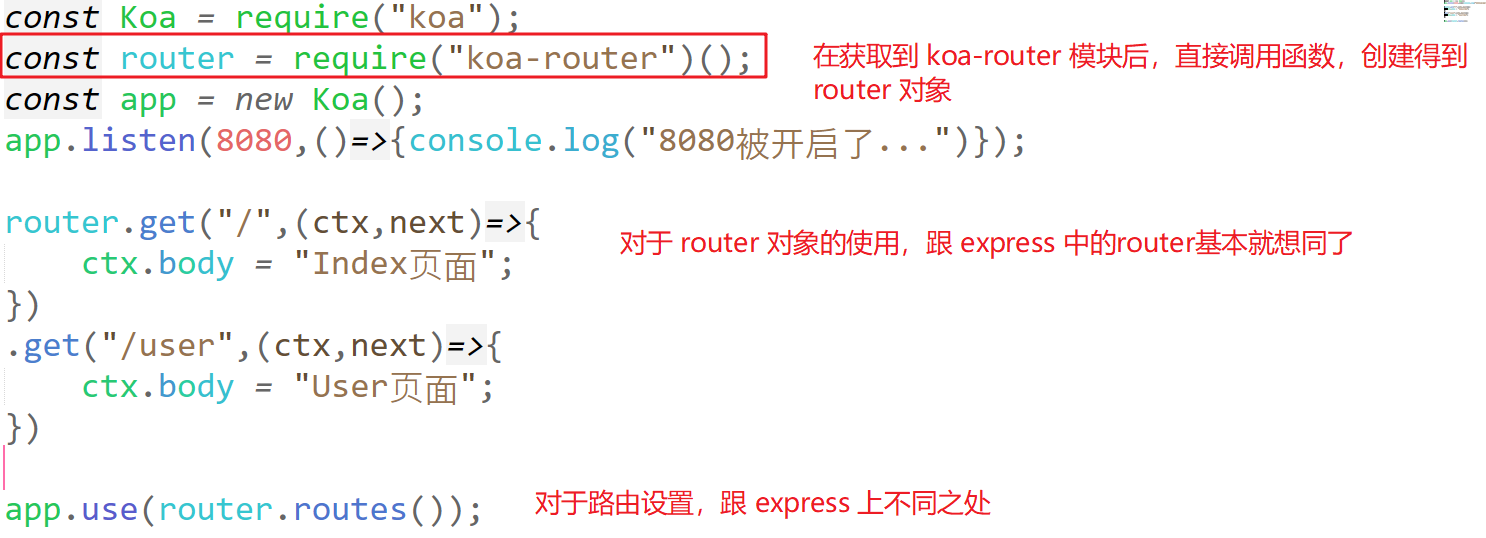
ctx.response.body ----> 这个写法就可以实现数据返回，页面显示

可以是字符串，也可以是 js 对象；ctx.response.body可以简写为 ctx.body



4. 对于 koa 中 路由的介绍

借助 npmjs.com 这个网站，在这个网站上阅读指定 库 的基本使用



注意点的介绍：

a>> 对于路由实现分文件的写法



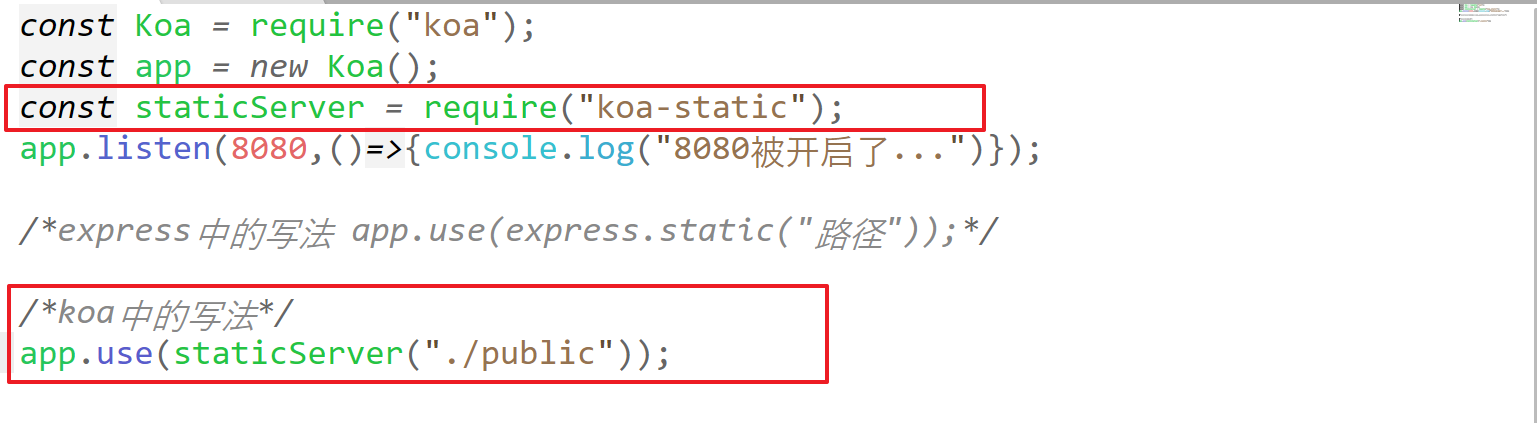
b>> 关于前后端路由的区分



5. 对于静态文件的读取

借助 koa 所依赖的中间件( 第三方外部库 )

npm install koa-static



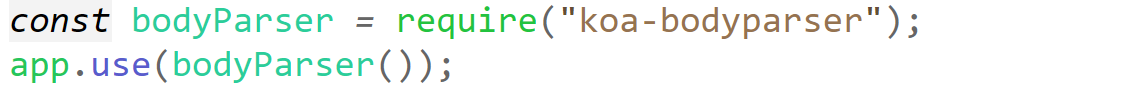
6. 对于 post 请求参数的获取

在express框架中，要想读取post请求传递过来的参数，需要借助 body-parser，然后通过设置 bodyParser 属性，使得 express 中能够通过 req.body 获取到post请求传递过来的参数；

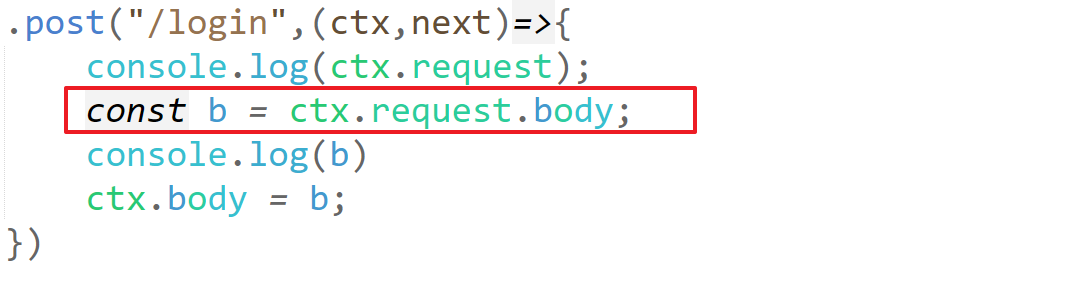
在koa 中，借助 koa-bodyparser

a>> npm install koa-bodyparser

b>> 读取到 koa-bodyparser 模块后，设置使用这个模块；



c>> 在对应的请求中的 ctx.request 中，就含有了post传递过来的参数了



7. 对于模板渲染的实现

在express框架中，结合 art-template 以及 express-art-template 来进行模板渲染的实现；

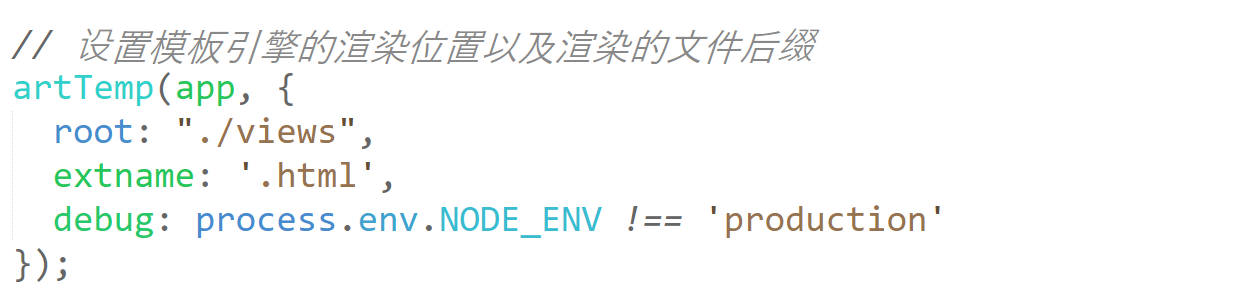
在 koa 框架中，结合 art-template 以及 koa-art-template

a>> 下载安装所需的外部库

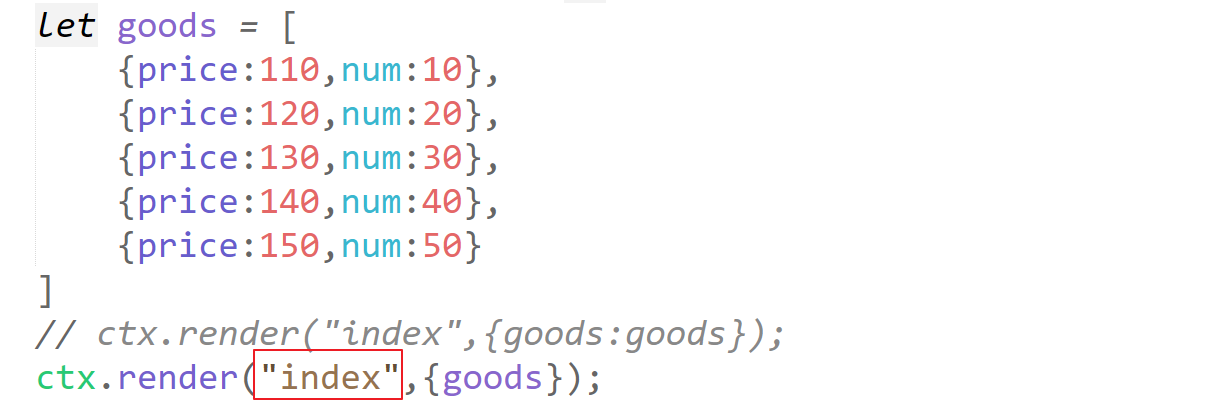
npm install art-template koa-art-template

b>> 引入对应的 **koa-art-template** 库

c>> 配置好模板文件的读取文件 以及 文件后缀读取



d>> 借助 ctx 中的 render 方法，来进行模板渲染



8. 对于 cookie 以及 session 的实现

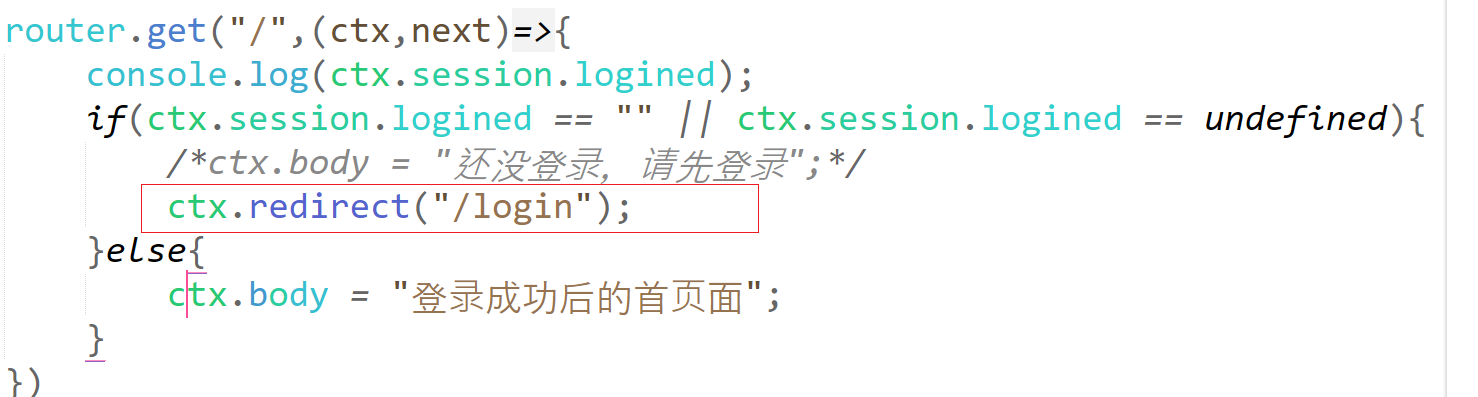
对于 koa 中 cookie 以及 session 的实现，这边需要依赖于外部的一个中间件(外部库)

koa-session



对于 koa 中的后端重定向的实现

a>> ctx.redirect("/login") ;

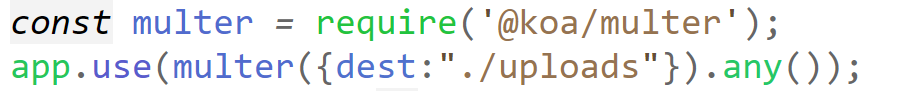


9. 对于 koa 中实现图片上传的操作

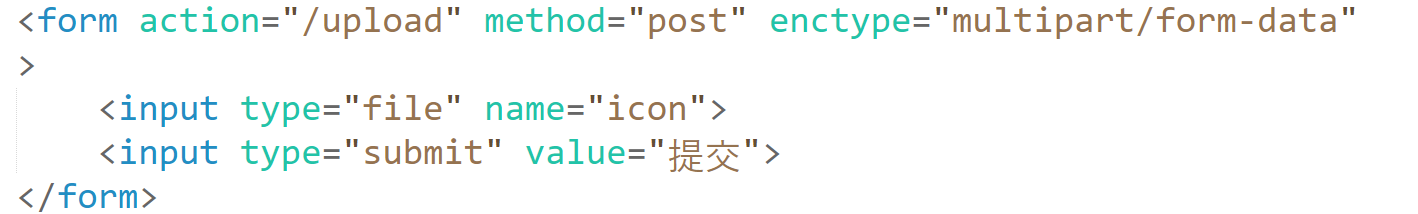
先使用 @koa/multer 来进行文件上传操作；

a>> npm install @koa/multer multer

b>> 在 app.js 中设置对应的配置



c>> 对于文件上传操作的实现





来使用 koa-body 来进行文件的上传

a>> npm install koa-body

b>> 在 app.js 文件中，读取得到 koa-body 模块对象，并设置好对应的参数



10. 对于脚手架的介绍

在express中，可以自己手动项目的结构，也可以借助 express-generator 来进行全局安装，并通过这个 express-generator 脚手架来快速搭建 express 项目的结构

在 koa 中，也可以自己手动搭建项目结构，也可以借助 **koa-generator** ，通过全局安装这个 koa-generator ，来进行 koa 项目的快速搭建

koa 项目名称 ---> 快速创建 koa 项目，此时这个 koa 项目中 koa 版本是 1.x 版本

koa2 项目名称 ---> 快速搭建 2.x 版本的 koa 项目，这边选择 koa2 来进行项目的搭建

作业：

结合 koa + mysql 实现如下案例

