# 一 GET请求

const Koa = require('koa');  
const app = new Koa();  
/\*  
ctx中封装了两个请求对象：  
ctx:koa封装的上下文对象，包含req与res  
ctx.request：node源生请求对象，适合做底层开发  
 \*/  
app.use(async function(ctx){  
 *// console.log(ctx.request);  
 // console.log(ctx.req);* console.log(ctx.url); *//login?name=lisi* console.log(ctx.request.query); *//{ name: 'lisi' }* console.log(ctx.query); *//{ name: 'lisi' }* console.log(ctx.querystring); *//name=lisi* ctx.body = 'hello';  
});  
app.listen(8000);

# 二 POST请求

const Koa = require('koa');  
const app = new Koa();  
  
app.use(async function (ctx) {  
 if (ctx.url == '/' && ctx.method == 'GET') {  
 let html = `<form action="/" method="post">  
 <input type="text" id="username" name="username" value="">  
 <input type="text" id="password" name="password" value="">  
 <input type="submit" value="登录">  
</form>`;  
 ctx.body = html;  
 } else if (ctx.url == '/' && ctx.method == 'POST') {  
 let parseData = await parsePostData(ctx);  
 ctx.body = parseData;  
 } else {  
 ctx.body = '404';  
 }  
});  
app.listen(8001,()=>{  
 console.log('start...');  
});  
  
*//解析post方法*function parsePostData(ctx) {  
 return new Promise((resolve, reject) => {  
 try {  
 let postdata = '';  
 ctx.req.on('data', (data) => {  
 postdata += data  
 });  
 ctx.req.addListener("end", function () {  
 resolve(postdata);  
 });  
 } catch (error) {  
 reject(error);  
 }  
 });  
}  
  
*//转换为json方法*function parseQueryStr(queryStr){  
 let queryData={};  
 let queryStrList = queryStr.split('&');  
 console.log(queryStrList);  
 for( let [index,queryStr] of queryStrList.entries() ){  
 let itemList = queryStr.split('=');  
 console.log(itemList);  
 queryData[itemList[0]] = decodeURIComponent(itemList[1]);  
 }  
 return queryData  
}

# 三 body-parser中间件

*// npm install --save koa-bodyparser*const Koa = require('koa');  
const bodyParser = require('koa-bodyparser');  
  
const app = new Koa();  
  
app.use(bodyParser());  
app.use(async (ctx)=>{  
 if (ctx.url == '/' && ctx.method == 'GET') {  
 let html = `<form action="/" method="post">  
 <input type="text" id="username" name="username" value="">  
 <input type="text" id="password" name="password" value="">  
 <input type="submit" value="登录">  
</form>`;  
 ctx.body = html;  
 } else if(ctx.url == '/' && ctx.method == 'POST'){  
 let postData= ctx.request.body;  
 ctx.body=postData;  
 } else {  
 ctx.body = '404'; *//{"username":"22222","password":"22555"}* }  
});  
app.listen(8001);

# 四 简单的路由设计

const Koa = require('koa');  
const fs = require('fs');  
const app = new Koa();  
  
app.use(async (ctx)=>{  
 let url = ctx.url;  
 let html = await route(url);  
 ctx.body = html;  
});  
  
app.listen(8000);  
  
  
function render(page) {  
 return new Promise((resolve,reject)=>{  
 let pageUrl = `./views/${page}`;  
 fs.readFile(pageUrl,'binary',(err,data)=>{  
 if (err) {  
 reject(err);  
 } else {  
 resolve(data);  
 }  
 });  
 });  
}  
  
async function route(url) {  
 let page = '404.html';  
 switch (url){  
 case '/':  
 page = 'index.html';  
 break;  
 case '/todo':  
 page = 'todo.html';  
 break;  
 case '/404':  
 page = '404.html';  
 break;  
 default:  
 break;  
 }  
 let html = await render(page);  
 return html;  
}

# 五 路由中间件

*// npm install --save koa-router*const Koa = require('koa');  
const Router = require('koa-router');  
  
const app = new Koa();  
const router = new Router({  
 prefix:'/test' *//路由前缀---全局的*});  
  
router.get('/',function (ctx,next) {  
 ctx.body = 'index....';  
});  
router.get('/todo',function (ctx,next) {  
 ctx.body = 'todo....';  
});  
  
*//加载路由中间件*app.use(router.routes());  
  
*//只允许get*app.use(router.allowedMethods());  
  
app.listen(3000);

路由设计2

*// npm install --save koa-router*const Koa = require('koa');  
const Router = require('koa-router');  
  
const app = new Koa();  
  
let home = new Router(); *//子路由*let page = new Router(); *//子路由*home.get('/test',async (ctx)=>{ *//http://localhost:8000/home/test* ctx.body = 'home test...';  
}).get('/todo',async (ctx)=>{ *//http://localhost:8000/home/to do* ctx.body = 'home todo...';  
});  
  
page.get('/test',async (ctx)=>{  
 ctx.body = 'page router...';  
});  
  
  
*//父路由*let router = new Router();  
*//装载子路由*router.use('/home',home.routes());  
router.use('/page',page.routes());  
  
*//加载路由中间件*app.use(router.routes());  
  
  
app.listen(8000);

# 六 cookie

const Koa = require('koa');  
const app = new Koa();  
  
app.use(async(ctx)=>{  
 if(ctx.url=== '/index'){  
 ctx.cookies.set(  
 'MyName','Test',{  
 domain:'localhost', *// 写cookie所在的域名  
 // path:'/index', // 写cookie所在的路径，只在该路径下需要cookie* maxAge:1000\*60\*60\*24, *// cookie有效时长* expires:new Date('2018-12-31'), *// cookie失效时间* httpOnly:false, *// 是否只用于http请求中获取* overwrite:false *// 是否允许重写* }  
 );  
 ctx.body = 'cookie is ok';  
 }else{  
 if( ctx.cookies.get('MyName')){  
 ctx.body = ctx.cookies.get('MyName');  
 }else{  
 ctx.body = 'Cookie is none';  
 }  
 }  
});  
  
app.listen(3000,()=>{  
 console.log('[demo] server is starting at port 3000');  
});

# 七 ejs

/\*  
cnpm install --save koa-views //koa对模板的支持需要该中间件  
npm install --save ejs  
 \*/  
const Koa = require('koa');  
const views = require('koa-views');  
const path = require('path');  
const app = new Koa();  
  
*// 加载模板引擎*app.use(views(path.join(\_\_dirname, './views'), {  
 extension: 'ejs'  
}));  
  
app.use( async ( ctx ) => {  
 let title = 'hello koa2';  
 await ctx.render('index', {  
 title  
 });  
});  
  
app.listen(3000,()=>{  
 console.log('[demo] server is starting at port 3000');  
});

# 八 静态资源

*// npm install --save koa-static*const Koa = require('koa');  
const static = require('koa-static');  
  
const app = new Koa();  
  
*// http://localhost:3000/1.css*app.use(static(\_\_dirname+ '/public'));  
  
app.use( async ( ctx ) => {  
 ctx.body = 'hello world';  
});  
  
app.listen(3000, () => {  
 console.log('[demo] static-use-middleware is starting at port 3000')  
});