Express

基于 Node.js 平台，快速、开放、极简的 Web 开发框架

13.2.1 Express环境搭建

Express 是一个保持最小规模的灵活的 Node.js Web 应用程序开发框架，为 Web 和移动应用程序提供一组强大的功能。

安装依赖

```
npm install express --save
```

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})
```

13.2.2 路由

路由配置

路由示例

```
var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('hello world')
})
```

路由方法

```
// GET method route
app.get('/', function (req, res) {
  res.send('GET request to the homepage')
```

```
})

// POST method route
app.post('/', function (req, res) {
  res.send('POST request to the homepage')
})
```

路由句柄

```
var express = require('express')
var router = express.Router()

// middleware that is specific to this router
router.use(function timeLog (req, res, next) {
  console.log('Time: ', Date.now())
  next()
})
// define the home page route
router.get('/', function (req, res) {
  res.send('Birds home page')
})
// define the about route
router.get('/about', function (req, res) {
  res.send('About birds')
})

module.exports = router
```

```
var birds = require('./birds')

// ...

app.use('/birds', birds)
```

13.2.3 托管静态文件

为了提供诸如图像、CSS 文件和 JavaScript 文件之类的静态文件，请使用 Express 中的 express.static 内置中间件函数。

此函数特征如下：

```
express.static(root, [options])
```

例如，通过如下代码就可以将 public 目录下的图片、CSS 文件、JavaScript 文件对外开放访问了：

```
app.use(express.static('public'))
```

现在，你就可以访问 public 目录中的所有文件了：

```
http://localhost:3000/images/kitten.jpg
http://localhost:3000/css/style.css
http://localhost:3000/js/app.js
http://localhost:3000/images/bg.png
http://localhost:3000/hello.html
```

13.2.4 Get与POST传递参数

Get请求参数

```
const express = require("express");
const app = express();
const url = require("url");

app.get("/list",(req,res) =>{
    const page = url.parse(req.url,true).query.page || 1;
    res.send({
        status:200,
        page:page
    })
})


app.listen(3000,function(){
    console.log("服务器运行在3000端口上");
})
```

POST请求参数

Get请求方式可以直接浏览器访问，但是post不可以的。所以，我们需要安装一个postman做测试

```
const express = require("express");
const app = express();

app.use(bodyParser.urlencoded({
    extended:true
}))

app.post("/login",(req,res) =>{
    const { username, password } = req.body;
    res.send({
        status:200,
```

```
        username:username,
        password:password
    })
})

app.listen(3000,function(){
    console.log("服务器运行在3000端口上");
})
```

13.2.5 使用中间件

这里我们正好借着中间件这个点，将数据引入到我们项目之中

```
// index.js

const express = require("express");
const app = express();
const bodyParser = require("body-parser");
const router = require("./router")

app.use(bodyParser.urlencoded({
    extended:true
}))

app.use("/api",router);

app.listen(3000,() =>{
    console.log(3000);
})
```

```
// router.js

const express = require("express");
const router = express.Router();
const sqlClient = require("./config")
const url = require("url");

/**
 * 注册
 */
router.post("/register", (req, res) => {
    const { username, password, email } = req.body;
    const sql = "insert into user values(null,?,?,?)";
    const arr = [username, password, email]
    sqlClient(sql, arr, result => {
        if (result.affectedRows > 0) {
            res.send({
                status: 200,
                msg: "注册成功"
```

```
            })
        } else {
            res.send({
                status: 401,
                msg: '注册失败'
            })
        }
    })
})

/**
 * 登陆
 */
router.post("/login", (req, res) => {
    const { username, password } = req.body;
    const sql = "select * from user where username=? and password=?";
    const arr = [username, password];
    sqlClient(sql, arr, result => {
        if (result.length > 0) {
            res.send({
                status: 200,
                username
            })
        } else {
            res.send({
                status: 401,
                msg: "登陆失败"
            })
        }
    })
})

module.exports = router;
```

```
// config.js

const mysql = require("mysql");

const client = mysql.createConnection({
    host:"localhost",
    user:"root",
    password:"",
    database:"itbaizhan"
})

const sqlClient = (sql,arr,callback) =>{
    client.query(sql,arr,(error,result) =>{
        if(error){
            console.log(error);
            return;
        }
```

```
            callback(result)
        })
    }

    module.exports = sqlClient
```