# Express服务启动常规七步

1. 引用模块

var express=require('express'),

compression=require('compression'),

bodyParser=require('body-parser'),

mongo=require('mongoskin');

1. 实例化express.js对象和其它对象

var app=express();

var db=mongo.db('mongodb://localhost:27017/test',{native\_parser:true});

1. 连接数据库（MongoDB,Redis,MySQL等）

var session=require('express-session');

var RedisSession=require('connect-redis')(session);

app.use(session({

store : new RedisStroe(options),

secret : 'express'

}));

1. 配置（模块引擎，文件扩展名等）
   1. app.set('view engine','jade');
   2. app.set('port',process.env.PORT || 3000);
2. 定义中间件（异常处理，静态文件目录，cookies解析器等）

app.use(bodyParser.json());

1. 定义路由

app.get('/',renderHomePage);

1. 启动服务（指定宿主和端口）

var server=http.createServer(app);

var boot=function(){

server.listen(app.get('port'),function(){

console.info('Express server listening on port ' + app.get('port'));

});

};

var shutdown=function(){

server.close();

}

if(require.main==module){

boot();

}else{

console.info('Running app as a module');

exports.boot=boot;

exports.shutdown=shutdown;

exports.port=app.get('port');

}

# 数据处理工作流

当Express.js监听到一个请求时，通过一系列中间件和路由进行处理，一般有以下四步：

1. Cookie解析
2. 参数解析
3. 鉴权并获取数据
4. 展示数据并结束响应

# Express 设置

## 系统设置

1. 无须再定义,大部分有默认值，可不设置
2. 常用设置
   * env
   * view cache
   * view engine
   * views
   * trust proxy
   * jsonp callback name
   * json replacer and json spaces
   * case sensitive routing
   * strict routing
   * x-powered-by
   * etag
   * query parser
   * subdomain offset

### env

1. 常用值（development\test\stage\preview\production）
2. 写值
   1. app.set(‘env’,’development’);
   2. process.env.NOVE\_ENV=development;
   3. $NODE\_ENV=development node app（推荐）
   4. set the NODE\_ENV variable on the machine(环境变量)

### view cache

1. 用于缓存视图模板，便于重复访问
2. 生产环境上，默认启用。

### view engine

视图模板引擎(EJS\Jade\Handlebar等)，默认使用Jade。

### views

app.set(‘views’,path.join(\_\_dirname,’templates’))

使用绝对路径，其中\_\_dirname代表app.js所在目录的绝对路径。

### trust proxy

使用反向代理服务器（如Nginx\Varnish）时启用。表示允许信任X-Forward-\*请求头。

默认不启用。

### jsonp callback name

1. 解决跨域访问限制的两种方式：
   1. 使用JSONP
      1. 只支持GET
      2. 支持老版本浏览器（不支持CORS）
      3. 可能导致XSS攻击
   2. 服务响应时增加“跨域访问资源共享头”（CORS Headers）（**推荐**）
      1. 支持各种HTTP请求
      2. 更好的异常处理
      3. 可限制访问范围
2. 默认值为callback
3. 用法
   1. localhost:3000/jsonp?cb=UpdateView

### json replacer and json spaces

### case sensitive routing

### strict routing

### x-powered-by

### etag

etag可看作通过浏览器发出请求时，服务器响应内容的唯一标识。再次访问时，如果etag未变化，则使用缓存，不再重新请求服务端。

### query parser

### subdomain offset

## 自定义设置

1. 需要自行定义