

## koa

Express是第一代最流行的web框架，它对Node.js的http进行了封装，用起来如下：

```
var express = require('express');
var app = express();
app.get('/', function (req, res) {
  res.send('Hello World!');
});
app.listen(3000, function () {
  console.log('Example app listening on port 3000!');
});
```

koa是Express的下一代基于Node.js的web框架。

随着新版Node.js开始支持ES6，Express的团队又基于ES6的generator重新编写了下一代web框架koa。和Express相比，koa 1.0使用generator实现异步，代码看起来像同步的：

```
var koa = require('koa');
var app = koa();

app.use('/test', function *() {
  yield doReadFile1();
  var data = yield doReadFile2();
  this.body = data;
});
app.listen(3000);
```

koa团队并没有止步于koa 1.0，他们非常超前地基于ES7开发了koa2，和koa 1相比，koa2完全使用Promise并配合async来实现异步。

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

app.use( async (ctx, next) => {
  console.log(` ${ctx.request.method} ${ctx.request.url} `); //打印方法和URL
  await next(); // 调用下一个middleware
});
```

```
app.use( async (ctx, next) => {  
    const start = new Date().getTime(); //获取当前时间  
    await next();    // 调用下一个middleware  
    const ms = new Date().getTime() - start; //耗费时间  
    console.log(`Time: ${ms}ms`);    //打印时间  
});
```

```
app.use( async (ctx, next) => {  
    await next();  
    ctx.response.type = 'text/html';  
    ctx.response.body = '<h1>hello koa!</h1>';  
    console.log("show")  
});
```

```
app.listen(3000);  
console.log('http://127.0.0.1:3000');
```

结果

GET /

show

4ms