Koa.js

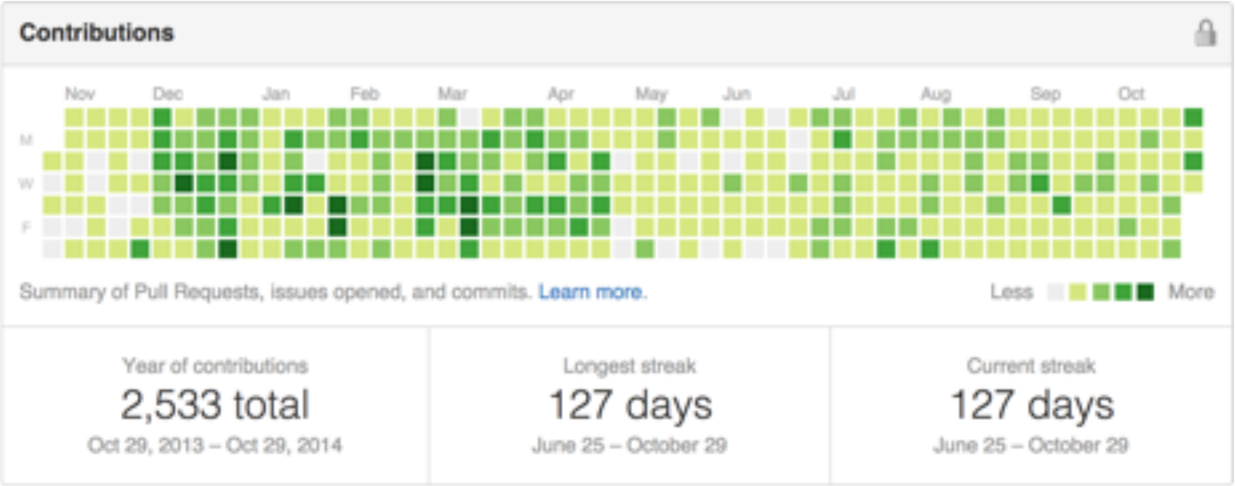
何翊宇(aka. 死马)

dead-horse

deadhorse_busi

2B码农死小马

koa CNPM



Contributions

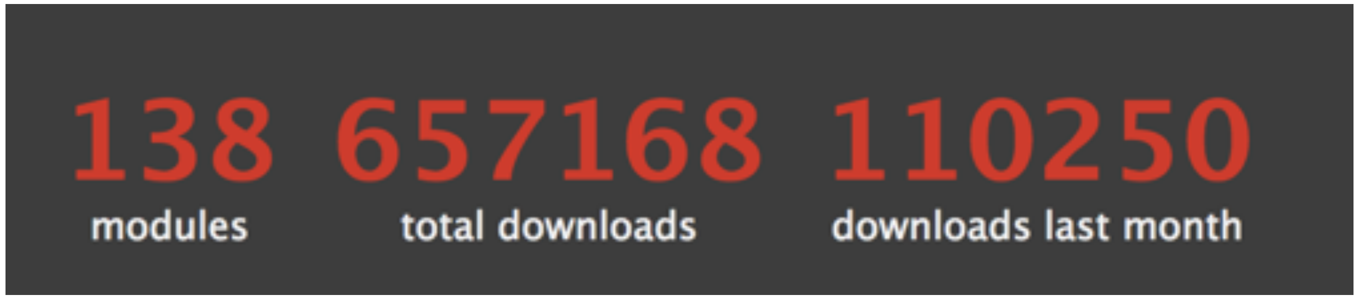
Summary of Pull Requests, issues opened, and commits. Learn more.

Less 🟩 More

Year of contributions
2,533 total
Oct 29, 2013 – Oct 29, 2014

Longest streak
127 days
June 25 – October 29

Current streak
127 days
June 25 – October 29



138 modules

657168 total downloads

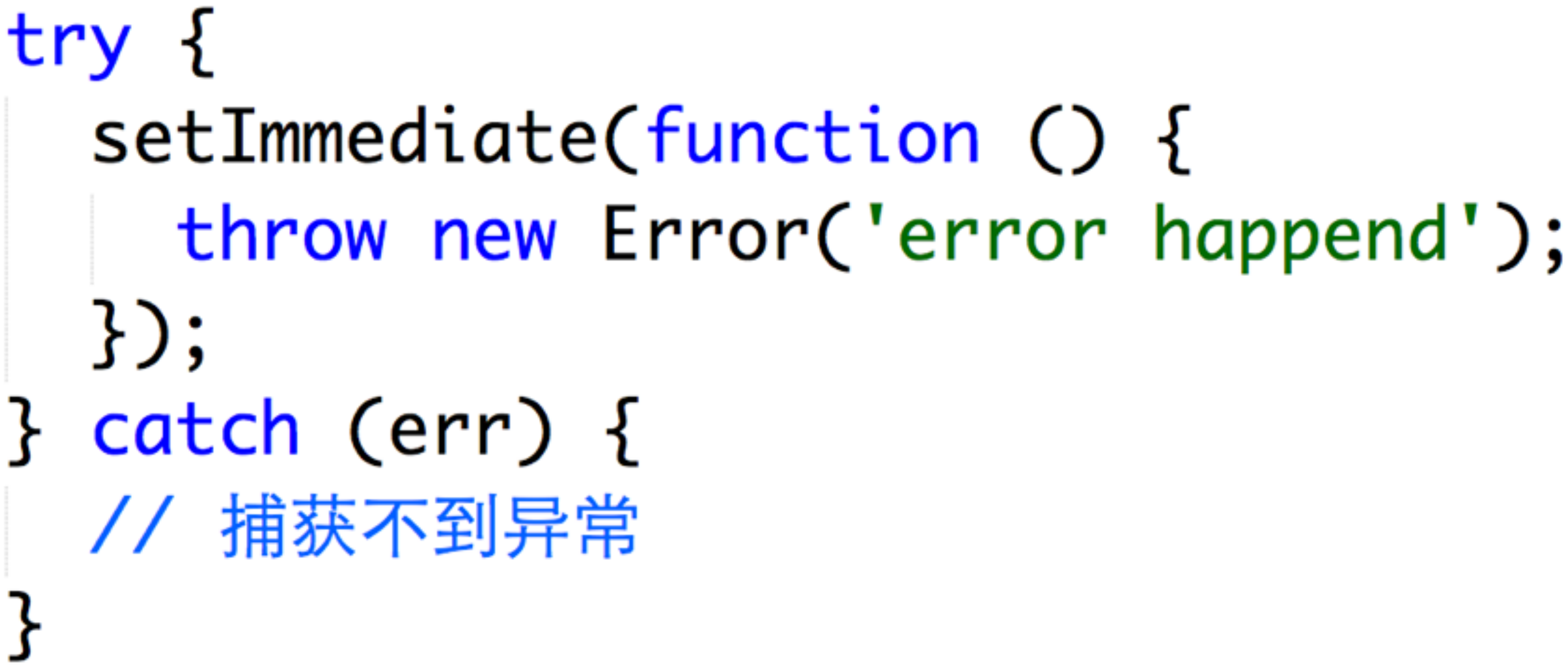
110250 downloads last month

什么是 Koa？

- 提供最基础功能的 web 框架
- TJ Holowaychuk 出品
- 和 express 共享大部分的底层模块
- 通过 generator 提供更优的异步控制和异常处理

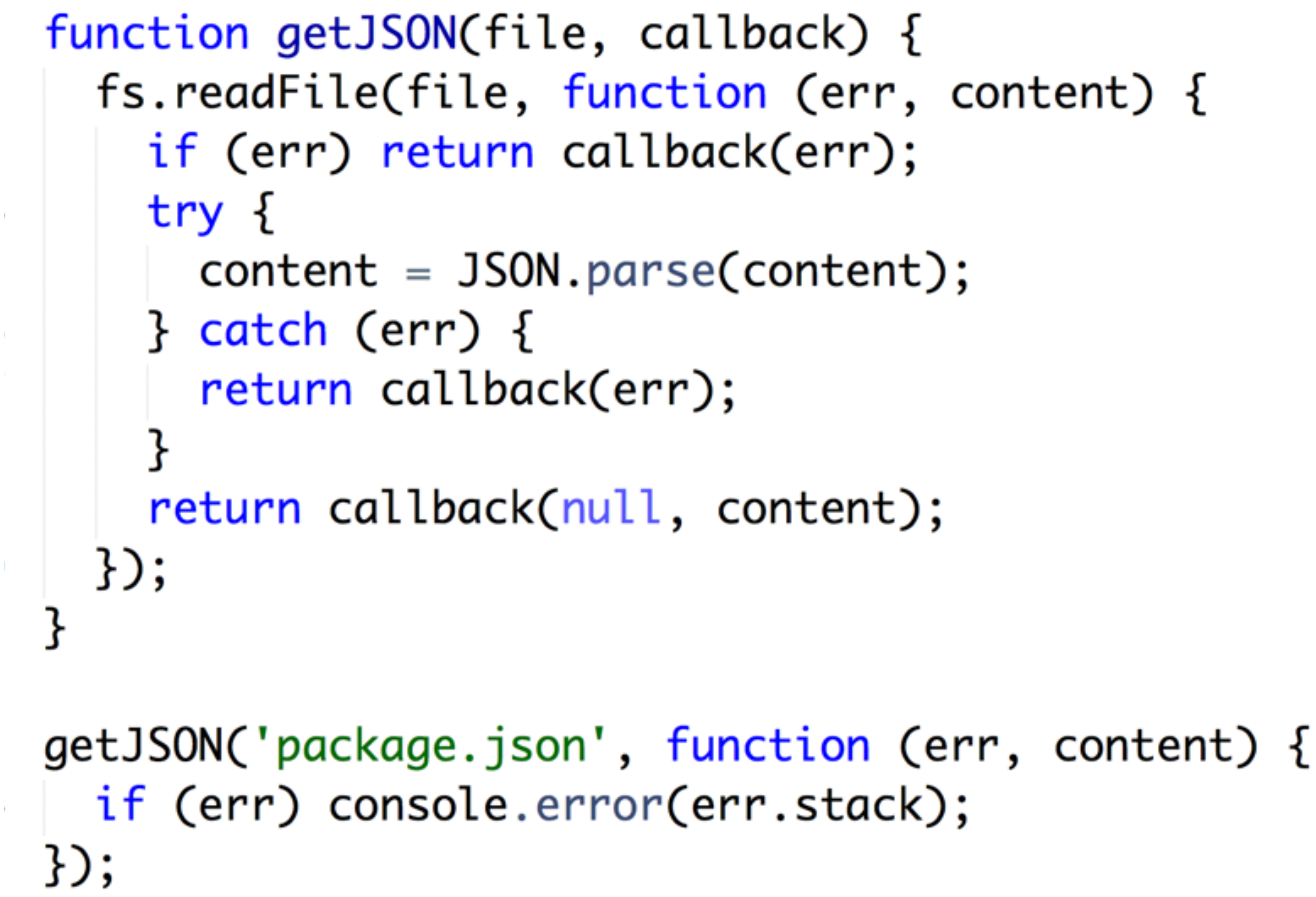
异常

无法捕获异步中的异常



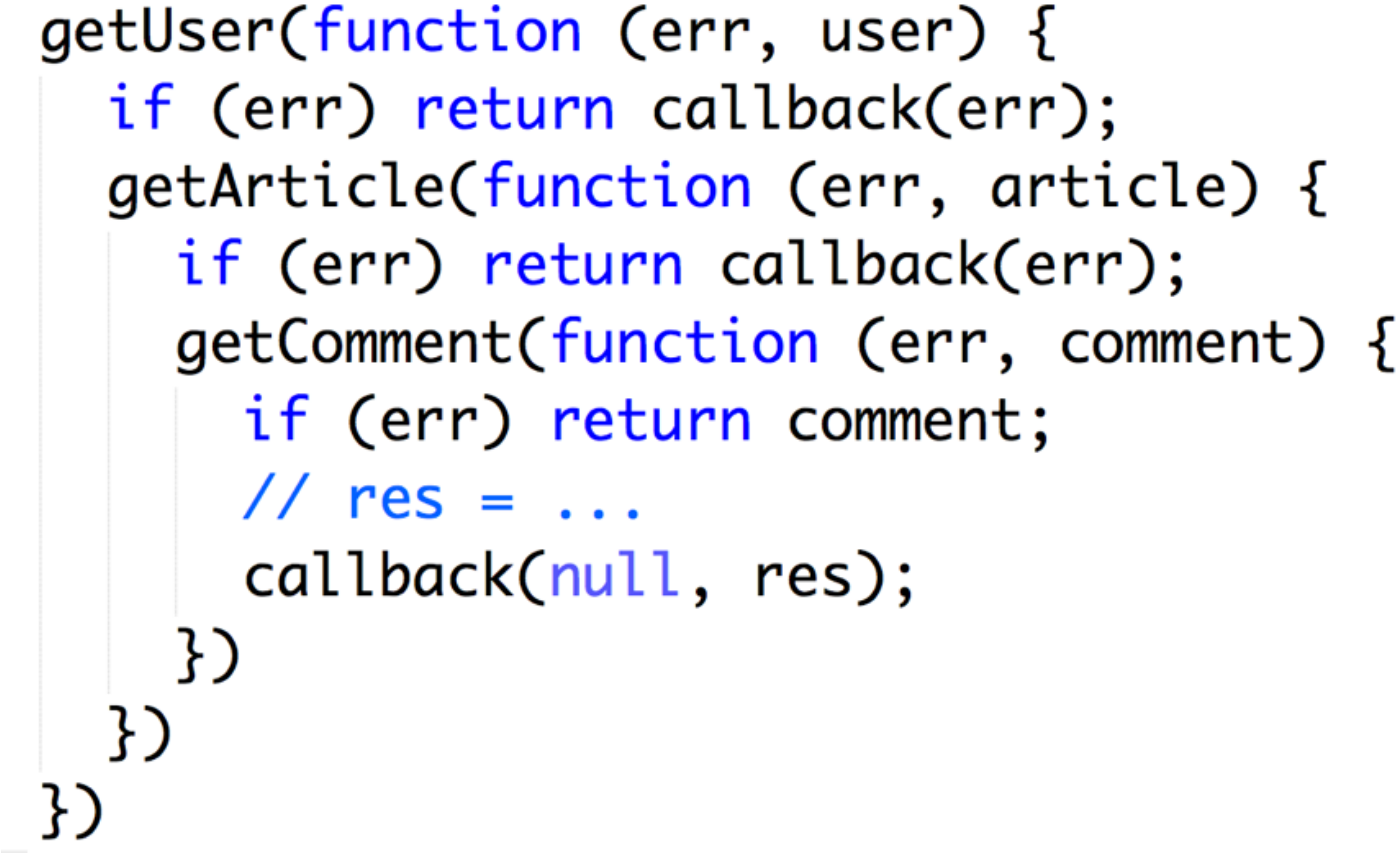
```
try {
  setImmediate(function () {
    throw new Error('error happend');
  });
} catch (err) {
  // 捕获不到异常
}
```

通过 callback 处理异常



```
function getJSON(file, callback) {
  fs.readFile(file, function (err, content) {
    if (err) return callback(err);
    try {
      content = JSON.parse(content);
    } catch (err) {
      return callback(err);
    }
    return callback(null, content);
  });
}

getJSON('package.json', function (err, content) {
  if (err) console.error(err.stack);
});
```

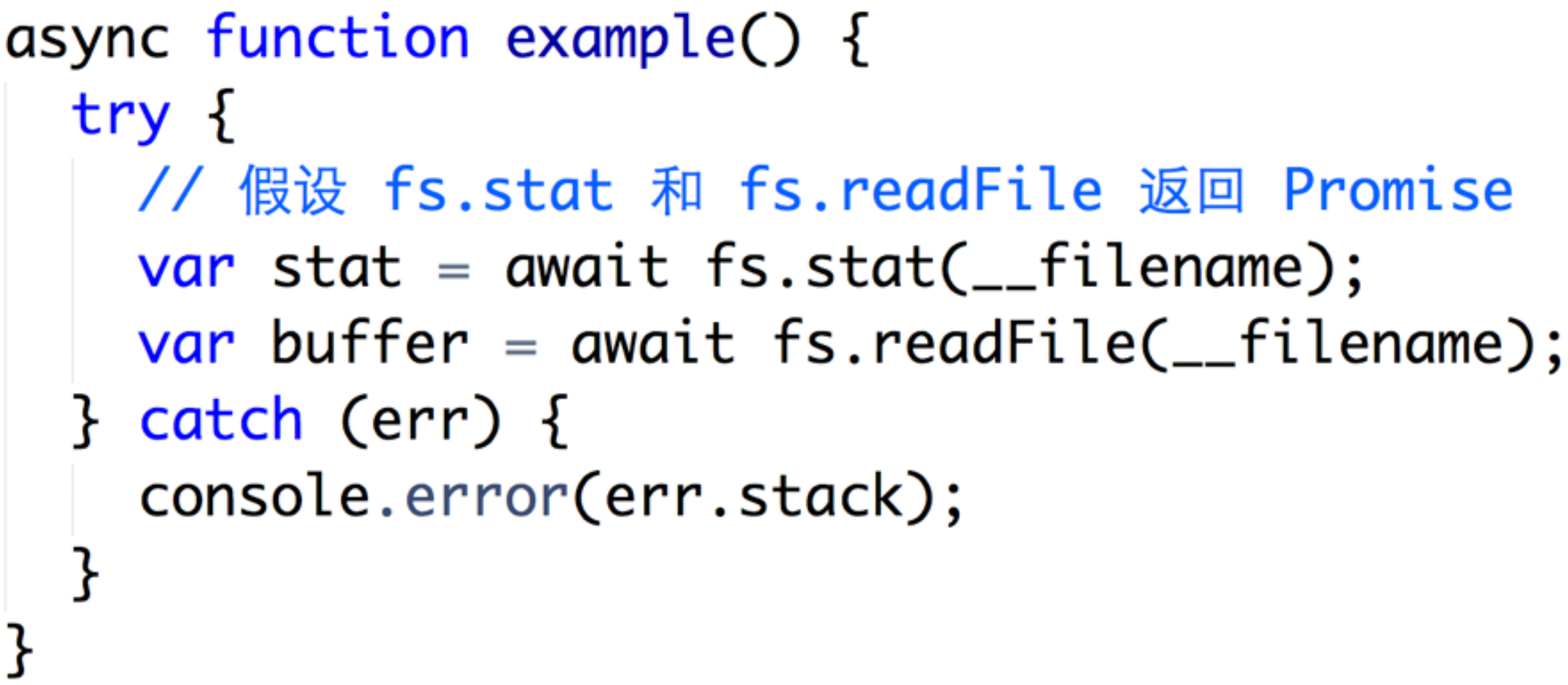
Callback Hell



```
getUser(function (err, user) {
  if (err) return callback(err);
  getArticle(function (err, article) {
    if (err) return callback(err);
    getComment(function (err, comment) {
      if (err) return comment;
      // res = ...
      callback(null, res);
    })
  })
})
```

- 重构代码 callbackhell.com
- async
- promise
- ES7 async function
- ES6 generator and co

async function

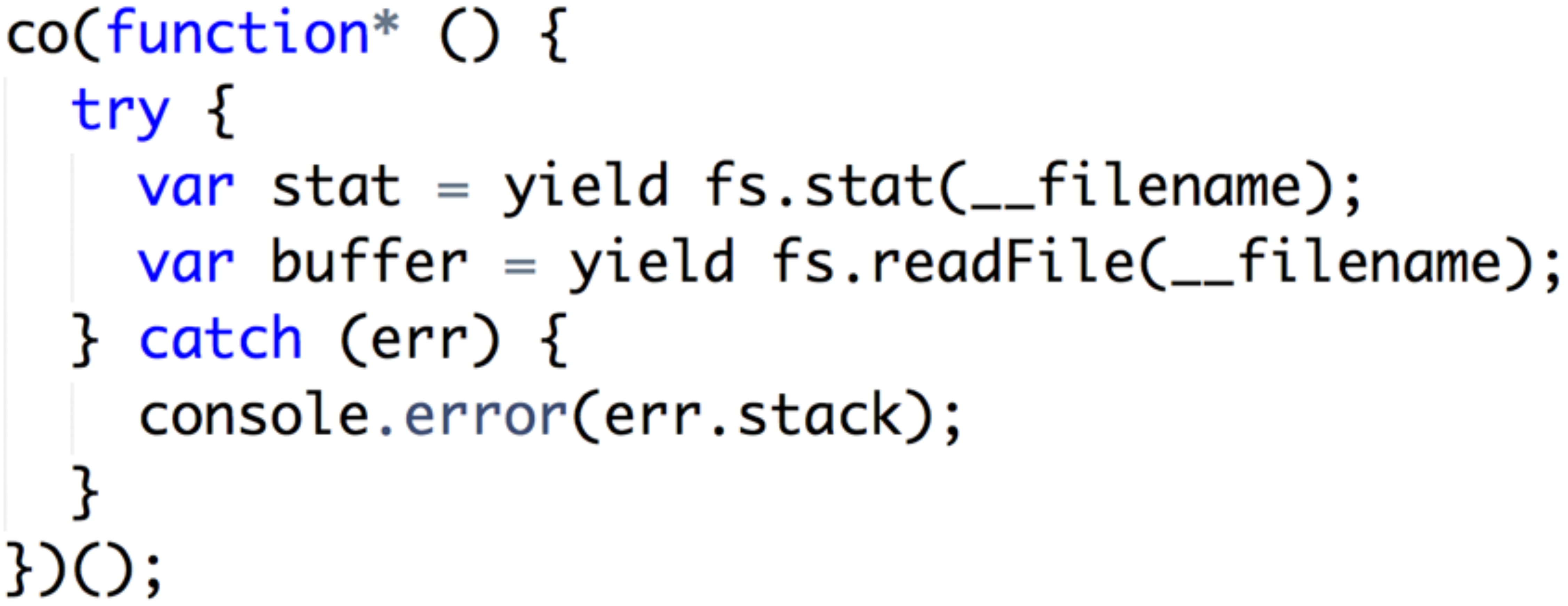


```
async function example() {
  try {
    // 假设 fs.stat 和 fs.readFile 返回 Promise
    var stat = await fs.stat(__filename);
    var buffer = await fs.readFile(__filename);
  } catch (err) {
    console.error(err.stack);
  }
}
```

async function 未至

generator 已在

generator and co

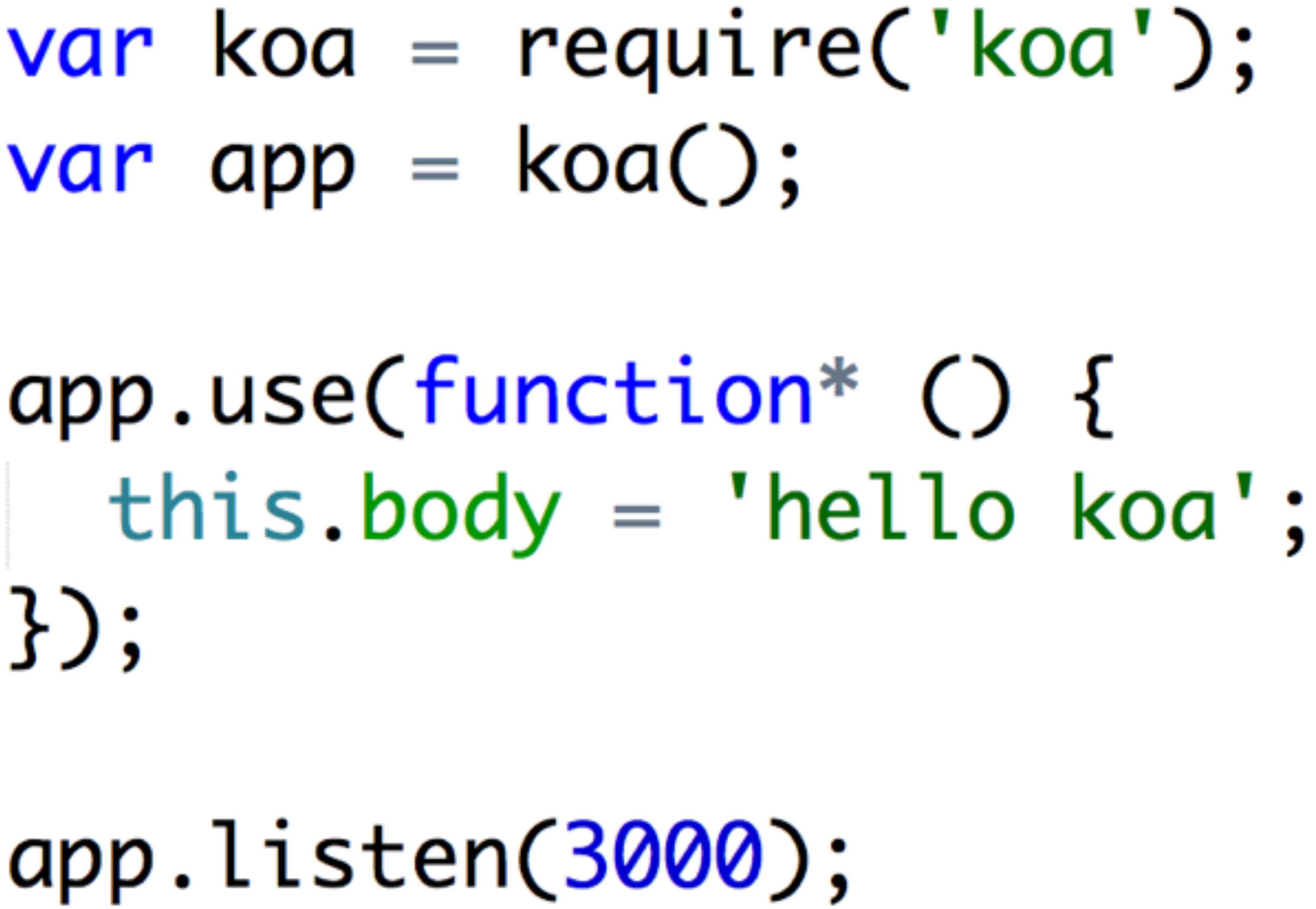


```
co(function* () {
  try {
    var stat = yield fs.stat(__filename);
    var buffer = yield fs.readFile(__filename);
  } catch (err) {
    console.error(err.stack);
  }
})();
```

前戏结束

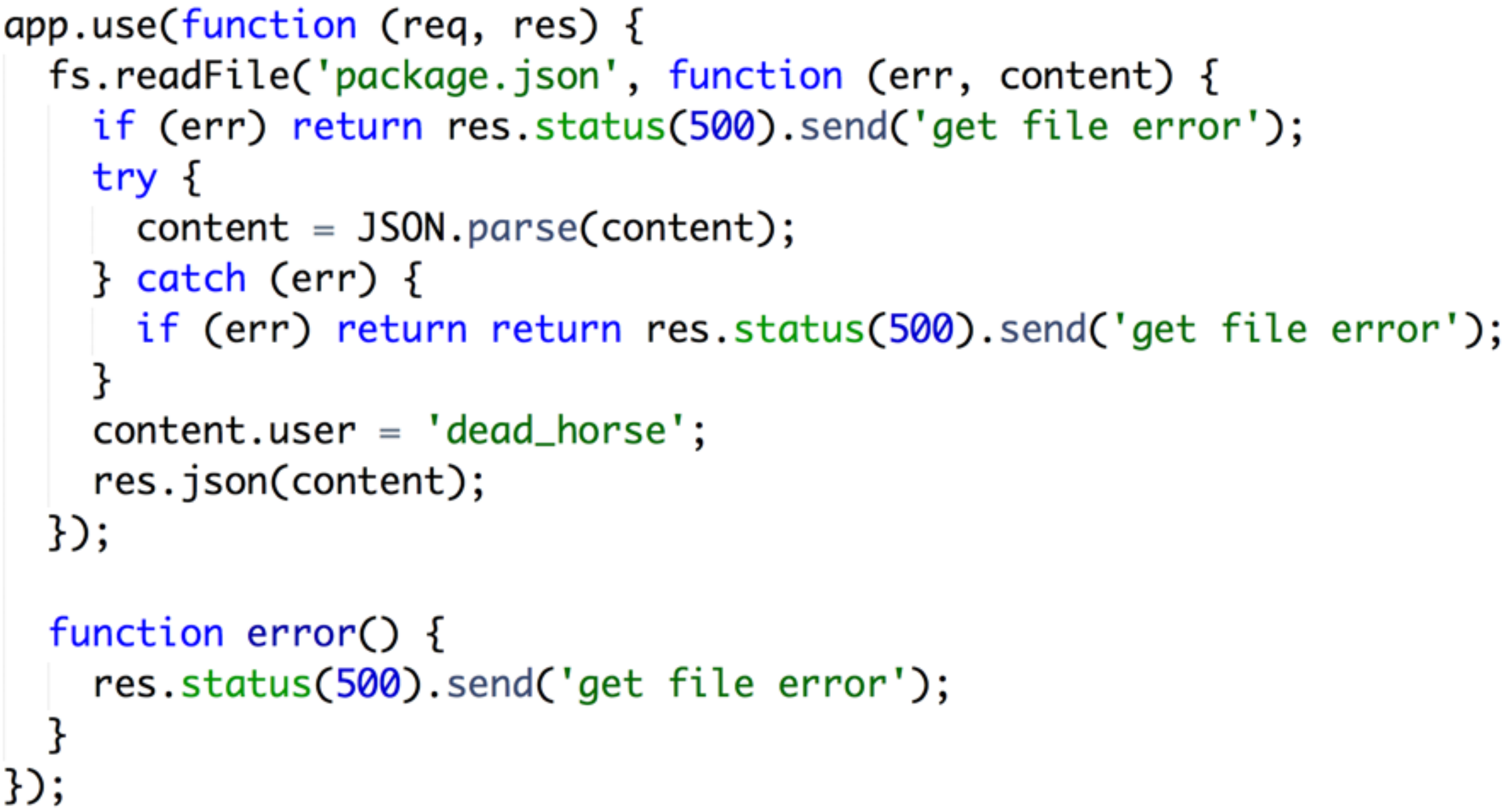
进入正题

Hello Koa



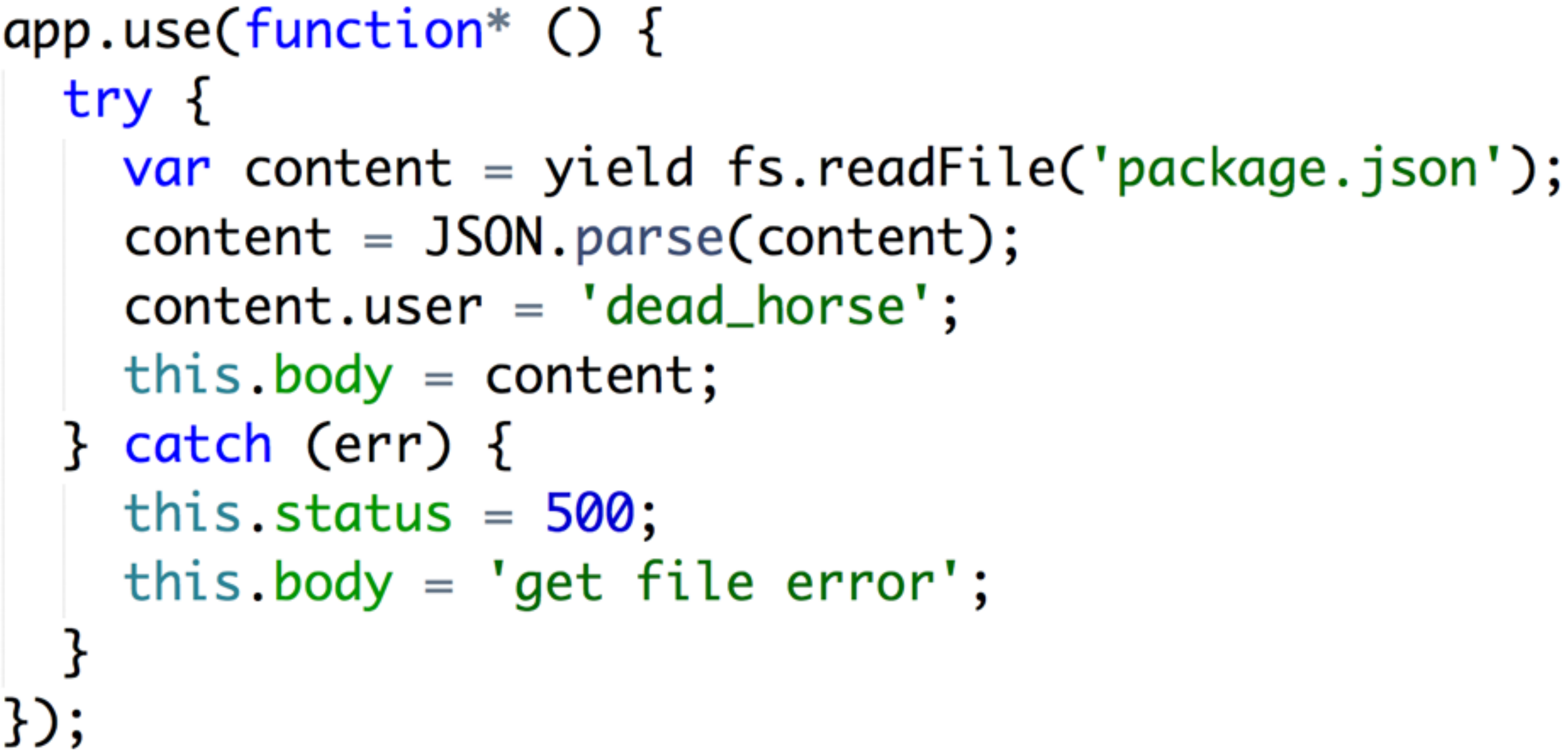
```
var koa = require('koa');
var app = koa();

app.use(function* () {
  this.body = 'hello koa';
});

app.listen(3000);
```

异常处理 in express



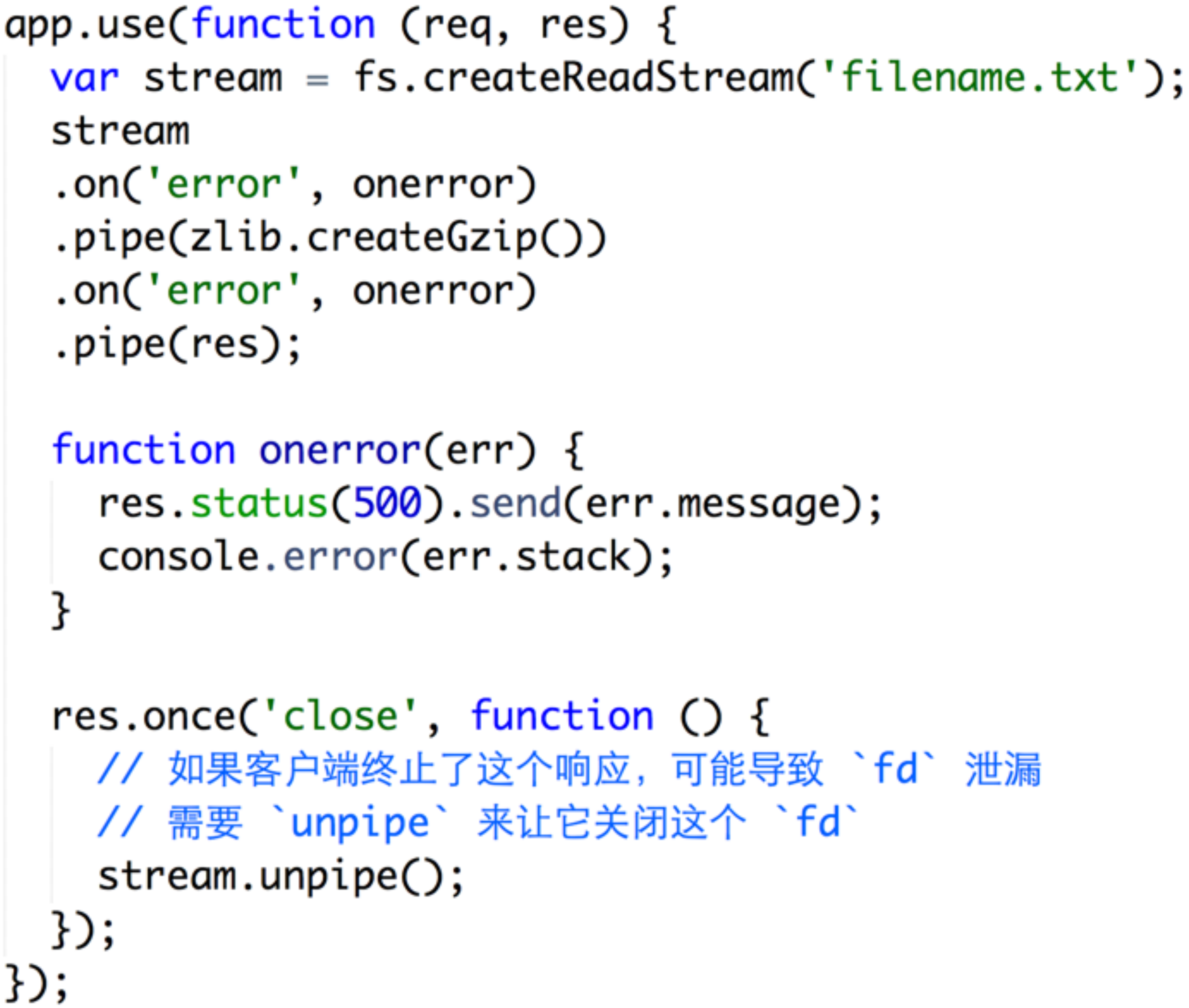
```
app.use(function (req, res) {
  fs.readFile('package.json', function (err, content) {
    if (err) return res.status(500).send('get file error');
    try {
      content = JSON.parse(content);
    } catch (err) {
      if (err) return return res.status(500).send('get file error');
    }
    content.user = 'dead_horse';
    res.json(content);
  });

  function error() {
    res.status(500).send('get file error');
  }
});
```

异常处理 in koa



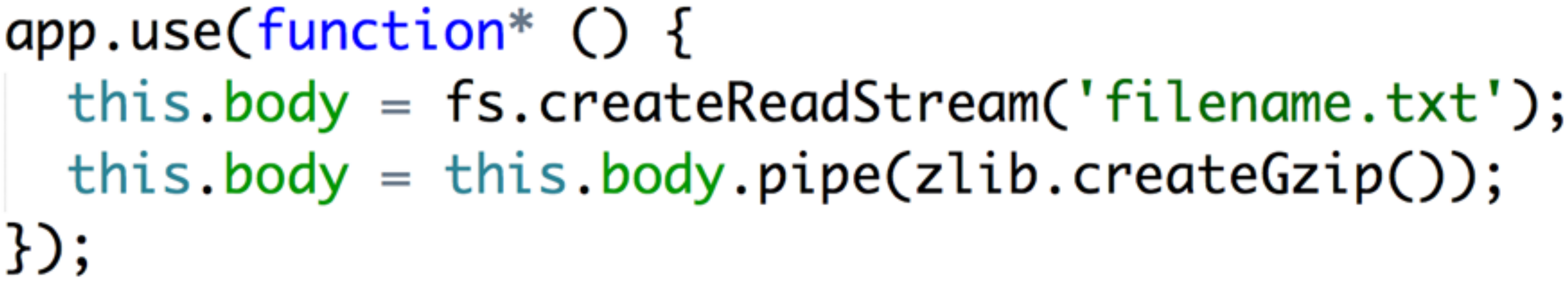
```
app.use(function* () {
  try {
    var content = yield fs.readFile('package.json');
    content = JSON.parse(content);
    content.user = 'dead_horse';
    this.body = content;
  } catch (err) {
    this.status = 500;
    this.body = 'get file error';
  }
});
```

stream in express



```
app.use(function (req, res) {
  var stream = fs.createReadStream('filename.txt');
  stream
  .on('error', onerror)
  .pipe(zlib.createGzip())
  .on('error', onerror)
  .pipe(res);

  function onerror(err) {
    res.status(500).send(err.message);
    console.error(err.stack);
  }

  res.once('close', function () {
    // 如果客户端终止了这个响应，可能导致 `fd` 泄漏
    // 需要 `unpipe` 来让它关闭这个 `fd`
    stream.unpipe();
  });
});
```

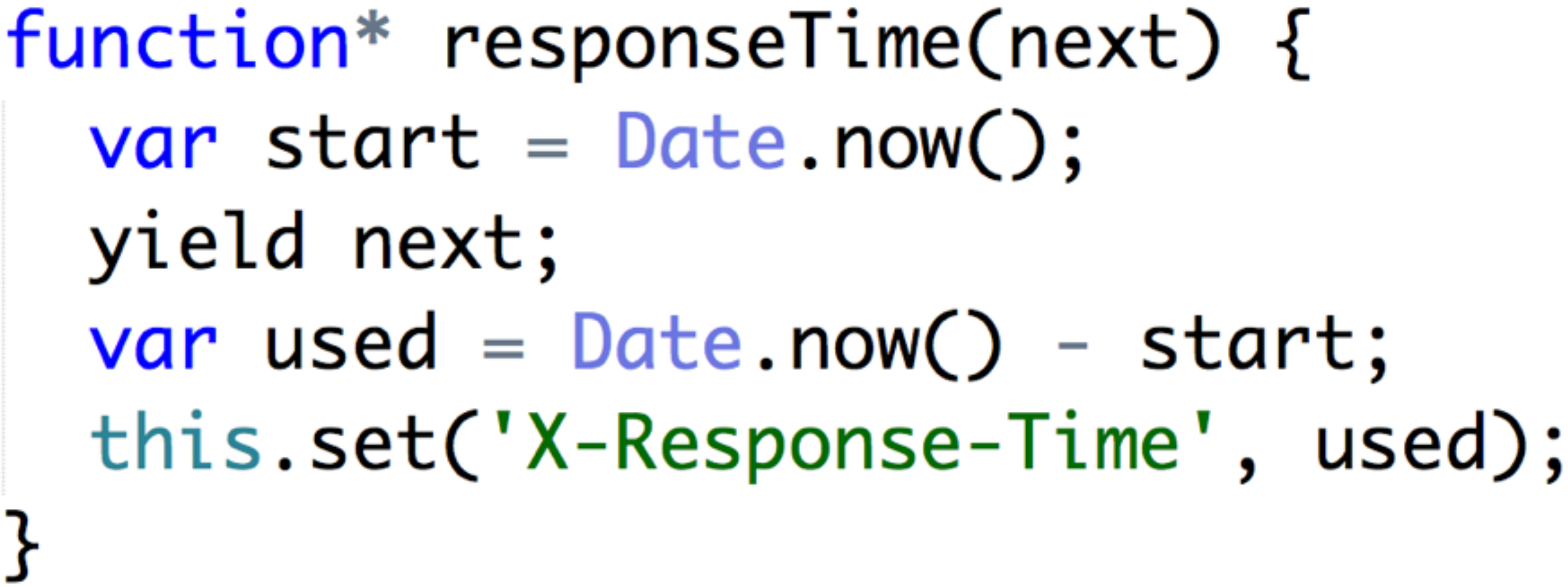
stream in koa



```
app.use(function* () {
  this.body = fs.createReadStream('filename.txt');
  this.body = this.body.pipe(zlib.createGzip());
});
```

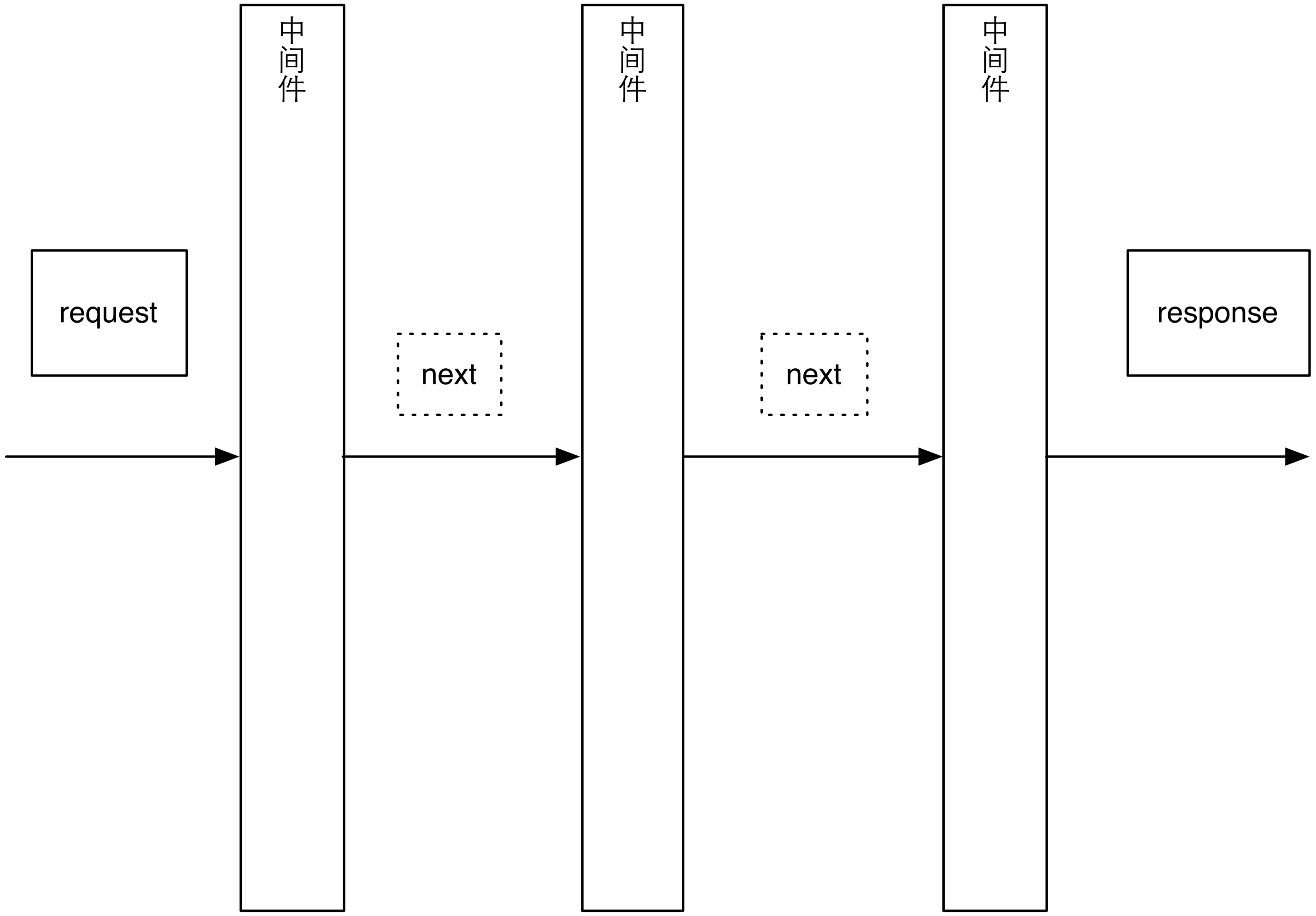
- generator 带来更自然的异常处理
- 抽象的 context + 神奇的 stter 简化代码

Middleware

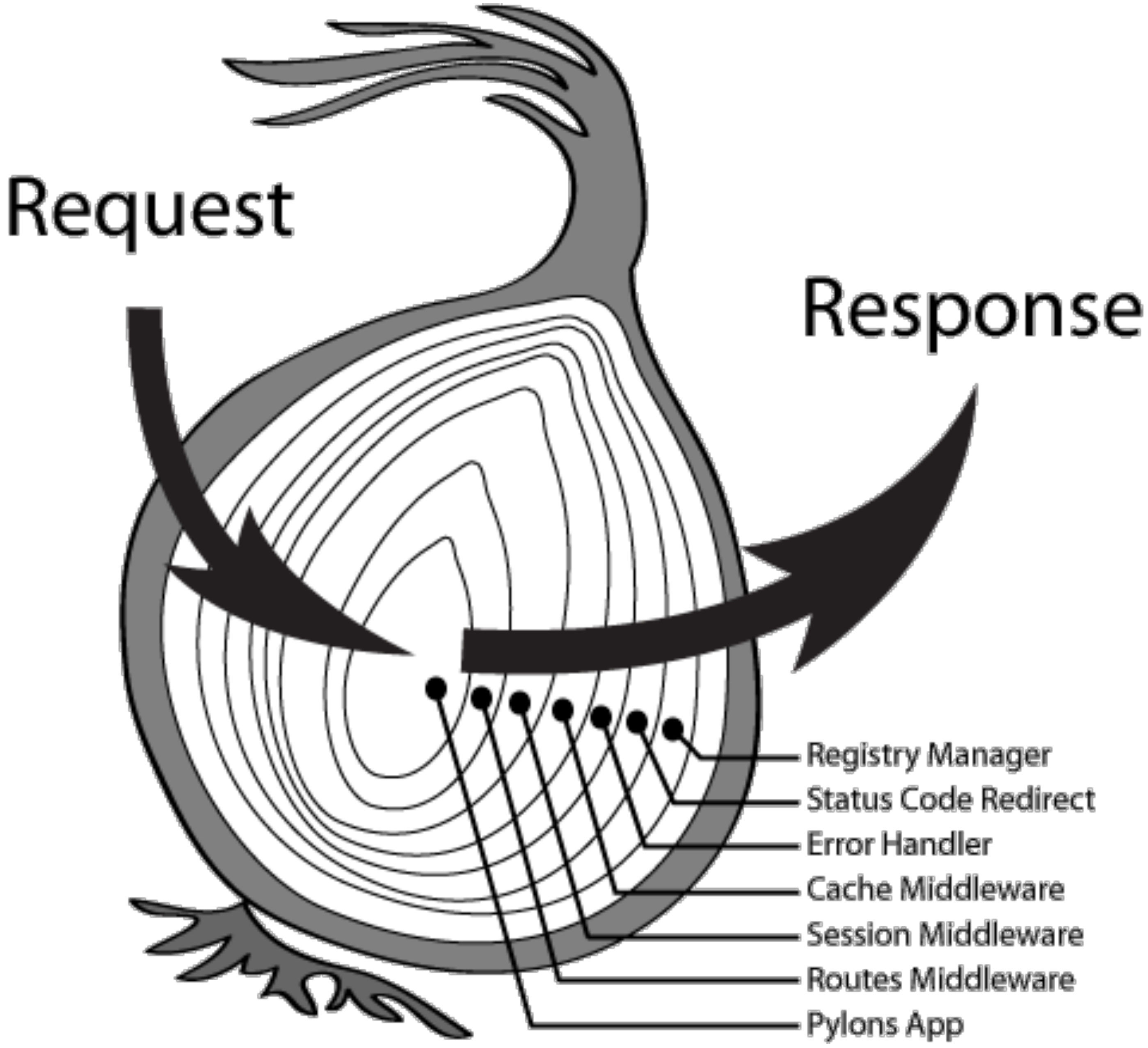


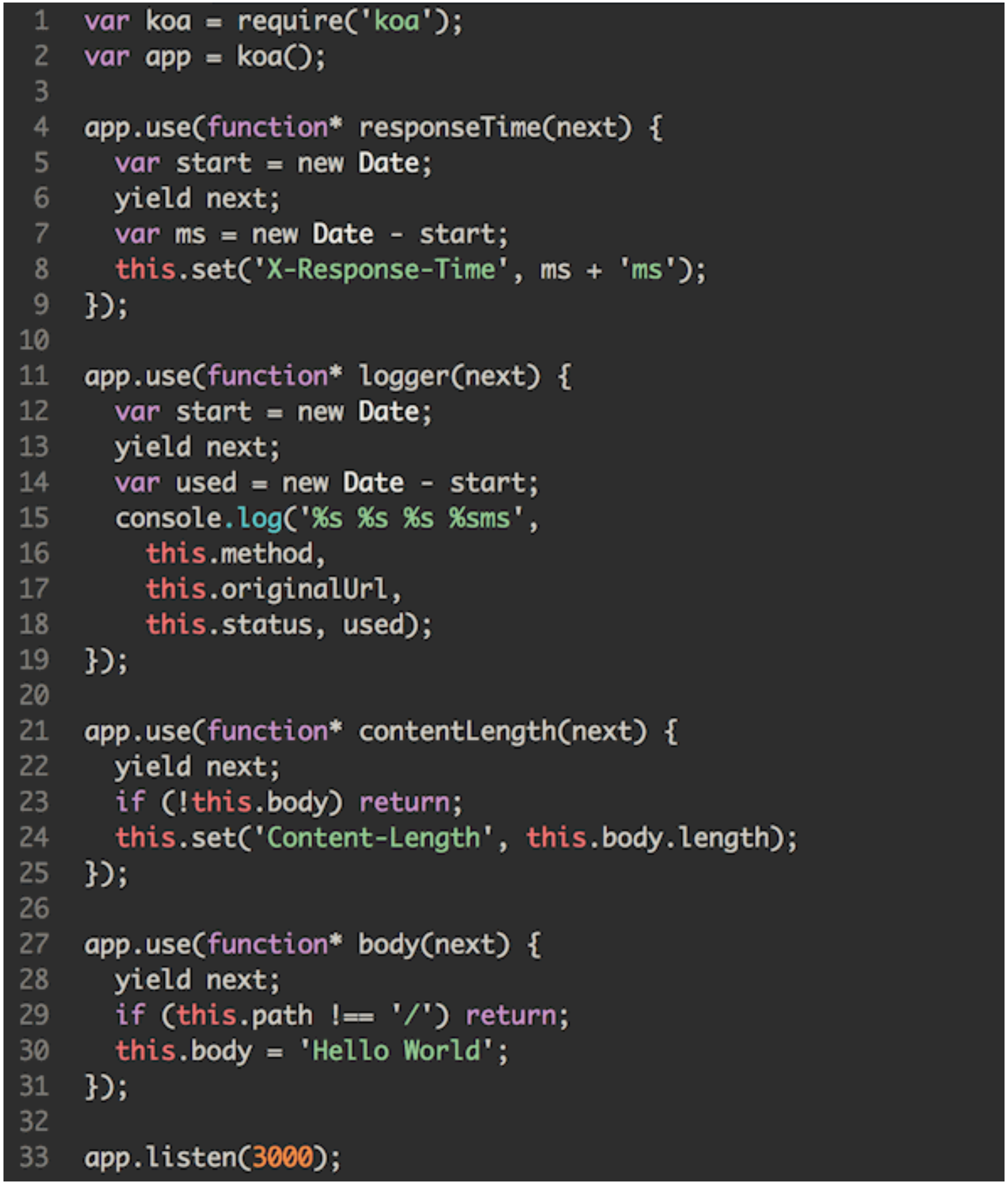
```
function* responseTime(next) {
  var start = Date.now();
  yield next;
  var used = Date.now() - start;
  this.set('X-Response-Time', used);
}
```

express



koa


Request
Response
Registry Manager
Status Code Redirect
Error Handler
Cache Middleware
Session Middleware
Routes Middleware
Pylons App



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

app.use(function* responseTime(next) {
  var start = new Date;
  yield next;
  var ms = new Date - start;
  this.set('X-Response-Time', ms + 'ms');
});

app.use(function* logger(next) {
  var start = new Date;
  yield next;
  var used = new Date - start;
  console.log('%s %s %s %sms',
    this.method,
    this.originalUrl,
    this.status, used);
});

app.use(function* contentLength(next) {
  yield next;
  if (!this.body) return;
  this.set('Content-Length', this.body.length);
});

app.use(function* body(next) {
  yield next;
  if (this.path !== '/') return;
  this.body = 'Hello World';
});

app.listen(3000);
```

express-compression

VS

koa-compress

> “Talk is cheap, show me the code.”
>
> – Linus Torvalds

https://github.com/dead-horse/koa-step-by-step

更多学习资料

- koajs/koa: 源码是最好的教程
- koa/examples: 各种示例
- koajs/kick-off-koa: 交互式学习教程
- koajs/workshop: TDD 式学习教程
- tj/co: 基于 generator 的异步流程控制解决方案
- jshttp: web 框架的基石

Q&A

Thanks