

# NODEJS 运营系统开发实践

演讲者 钱海龙 / [chainhelen@gmail.com](mailto:chainhelen@gmail.com)

# 生命周期

1. 设计
2. 技术选型
3. 编码
4. 测试
5. 部署、监控

# 设计

## 痛点

1. 新接手人员上手慢
2. 产品不同版本之间的特性混乱
3. 不同部门之间的联调低效
4. 项目维护人员替换成本大

# 设计

## 文档化

包含基本交互接口、参数说明、架构图、签名校验等，最好附有实例

# 设计

## 模块化

1. 分割整体，减少复杂性
2. 为每个部分创造了良好的边界

# 设计

## NODEJS —— 模块化

CommonJS/AMD : require / exports

ES6 : import / export

# 技术选型

user code

express

sequelize

...

@xunlei/  
notifier

...

xnpm 迅雷

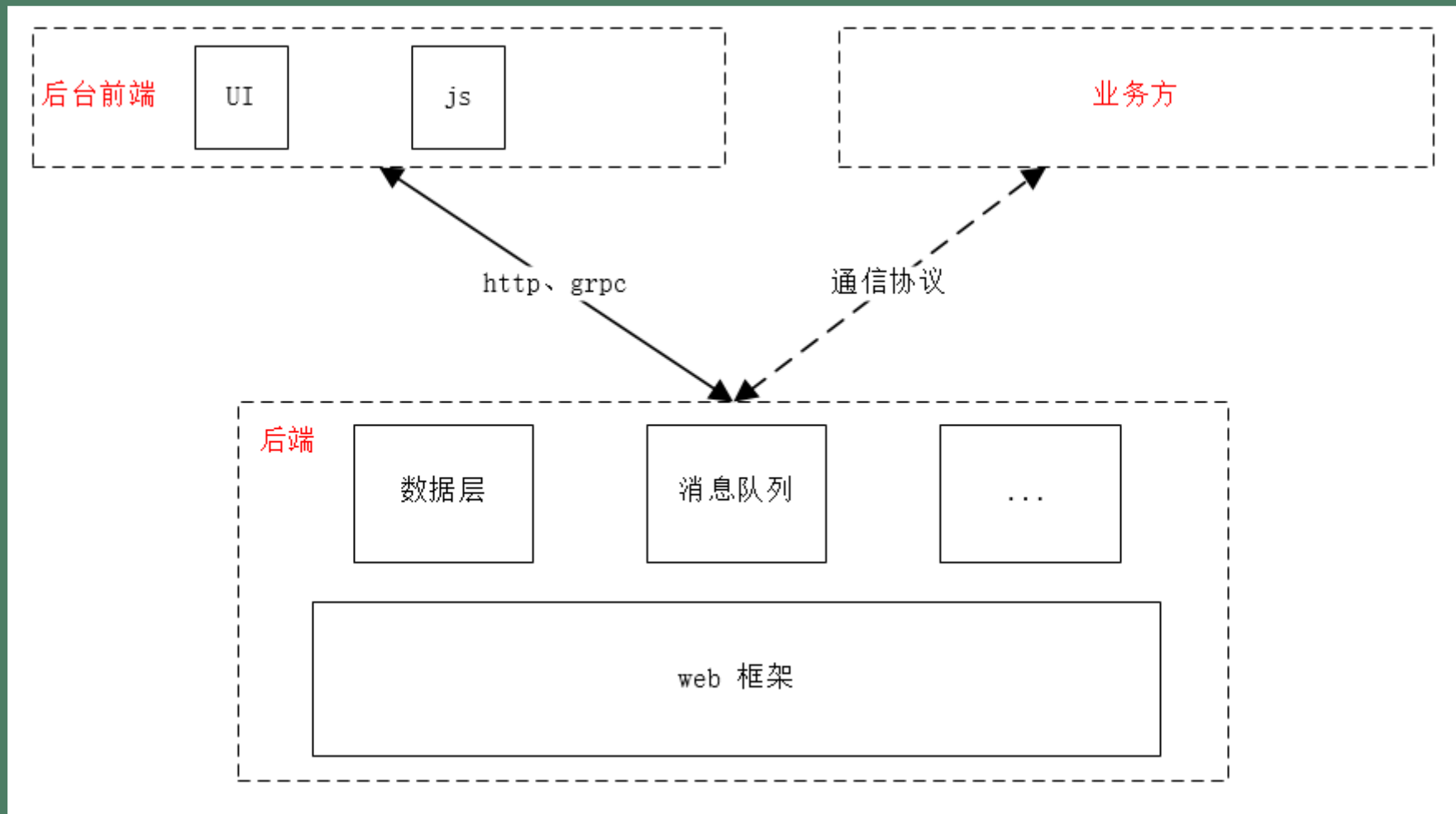
npm 社区

# 技术选型

```
11  "dependencies": {
12    "@xunlei/node-env": "^1.0.0",
13    "@xunlei/node-logger": "^1.0.0",
14    "@xunlei/node-stats": "^1.0.17",
15    "@xunlei/notifier": "^1.0.3",
16    "@xunlei/sso-session": "^1.0.6",
17    "bluebird": "^3.4.6",
18    "body-parser": "^1.15.2",
19    "chai": "4.1.1",
20    "connect-redis": "^3.3.0",
21    "cookie-parser": "^1.4.3",
22    "cross-env": "5.0.5",
23    "debug": "^2.2.0",
24    "element-ui": "^1.4.2",
25    "express": "^4.15.4",
26    "ioredis": "^2.3.0",
27    "ioredis-eventemitter": "^1.0.1",
28    "lru-cache": "^4.1.1",
29    "md5": "^2.2.1",
30    "mocha": "3.5.0",
31    "mongodb": "^2.2.31",
32    "mysql": "2.14.1",
33    "mysql-activerecord": "^0.8.6",
34    "mysql2": "1.4.1",
35    "node-resque": "^3.0.3",
36    "parseurl": "^1.3.1",
37    "path-to-regexp": "^1.5.3",
38    "qiniu": "^6.1.13",
39    "request-promise": "4.2.1",
40    "sequelize": "4.4.9",
41    "serve-favicon": "^2.3.0",
42    "supertest": "3.0.0",
43    "tupu-node-sdk": "^2.0.0"
44  },
```



# 技术选型



# WEB框架 —— 对比

框架名称	特性	点评
express	简单、实用，路由中间件等五脏俱全	最著名的web框架
koa	专注于异步流程改进	下一代的web框架
thinkjs	面向一些新特性	基于koa2，集成了丰富的中间件
fastify	提升性能	http接口性能接近原生

# WEB框架 —— EXPRESS

```
var app = require('express')
app.use(function(req, res, next) {
  var startTime = getCurTime()
  next()
  var endTime = getCurTime()
  console.log('cost time :' + (endTime - startTime))
})
app.listen(3000)
```

# WEB框架 —— EXPRESS

## nodejs 异步问题金字塔

```
app.use(function(req, res, next) {  
  ...  
  UserModel.find(condition, function(err, userInfo) {  
    ...  
    fs.readdir(condition, function(err, files) {  
      ...  
      CountModel.find(condition, function(err, countInfo) {  
        ...  
        next()  
      })  
    })  
  })  
})
```

co 利用了 promise、generator 特性

# WEB框架 —— KOA

## 同步思维写异步

```
app.use(async (ctx, next) => {  
  var userInfo = await UserModel.find(condition)  
  ...  
  var files = await fs.readdir(condition)  
  ...  
  var countInfo = await CountModel.find(condition)  
  ...  
  await next()  
})
```

# 异步流程核心理念

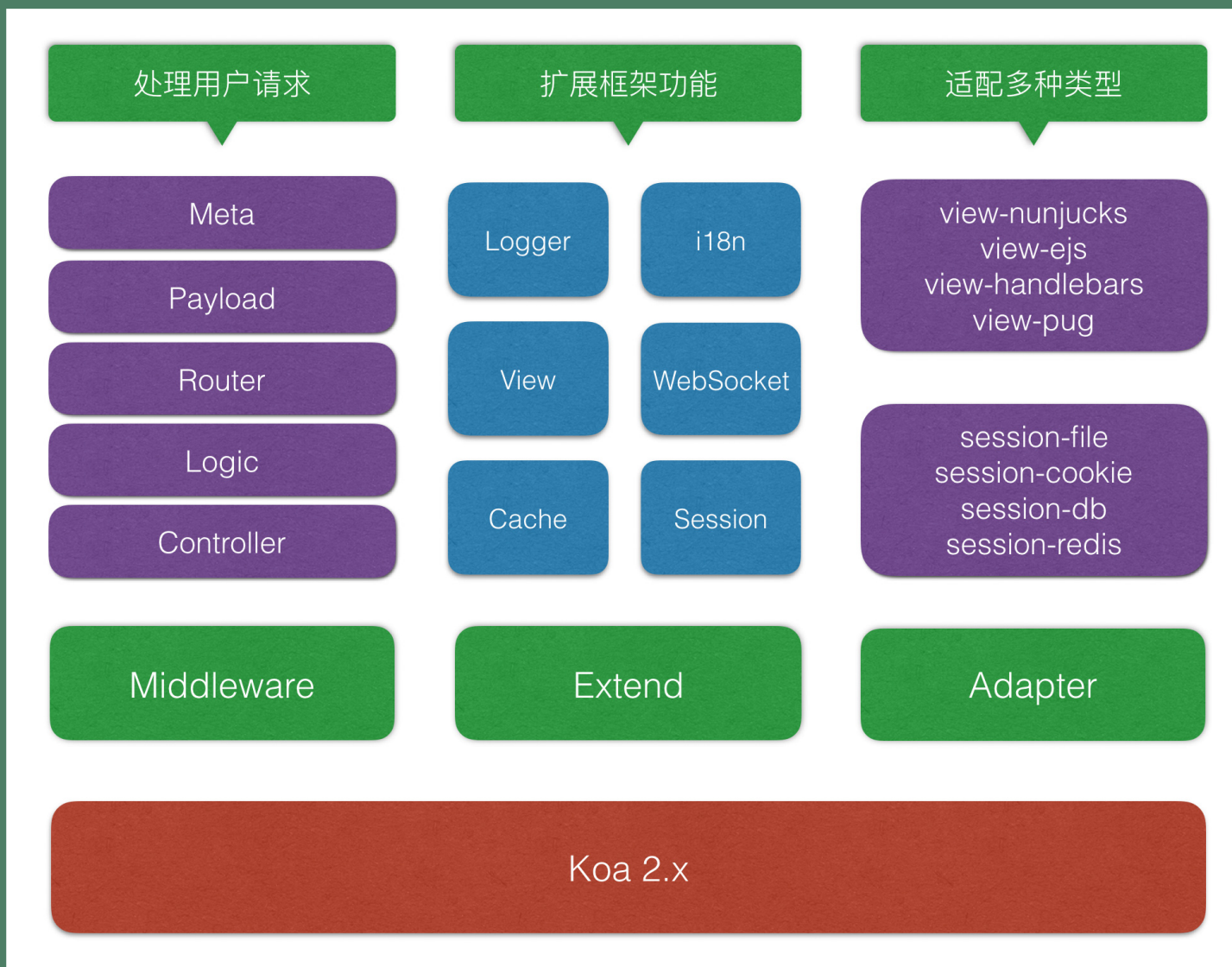
Express: error first callback

Koa: async / await

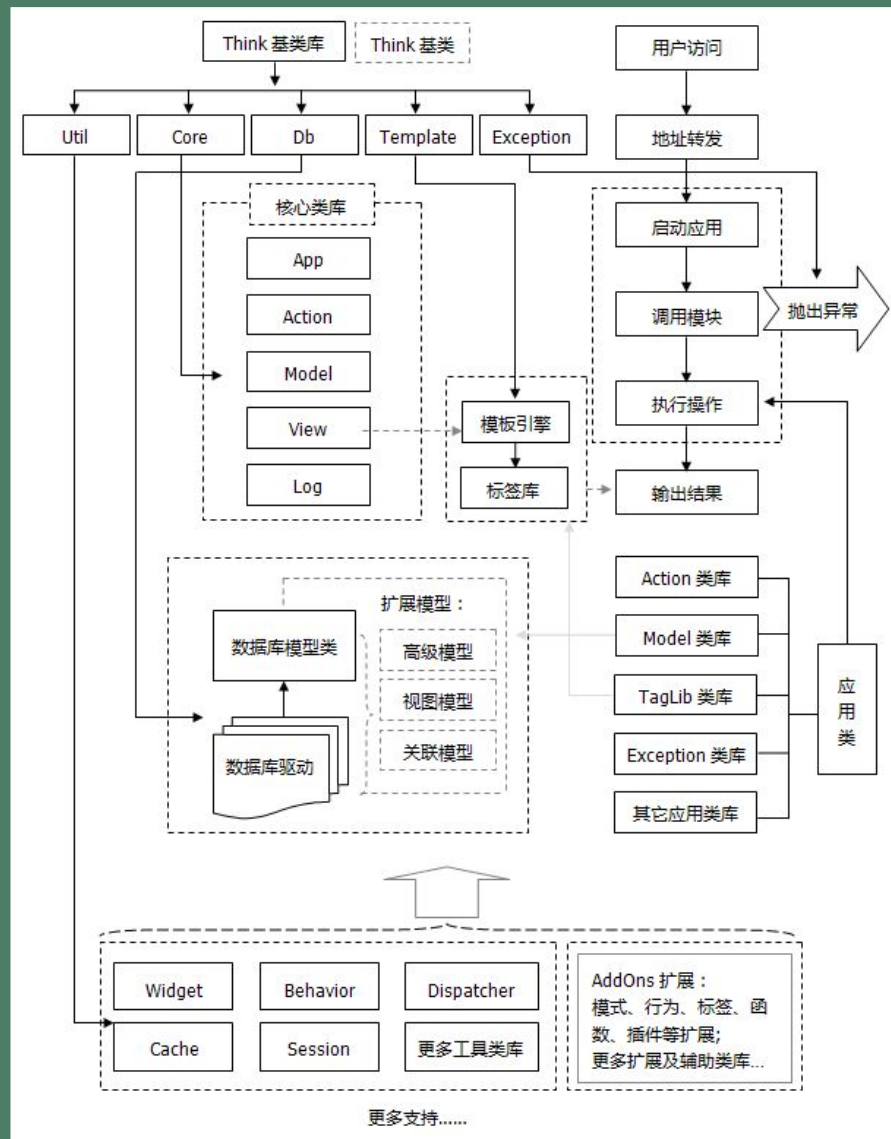
```
// express
app.use(function (req, res, next) {
  someAsyncFunction(function (error) {
    if (error) return next(error)
    // ...
    next()
  })
})

// koa
app.use(async function (ctx, next) {
  await somePromise
  await next()
})
```

# WEB框架 —— THINKJS



# WEB框架 —— THINKJS 对比 THINKPHP





# WEB框架 —— FASTIFY

Framework	Version	Router?	Requests/sec
hapi	16.6.2	✓	5,768
Restify	5.2.0	✓	17,589
Express	4.16.1	✓	20,860
total.js	2.8.0	✓	22,201
Koa ( koa-router )	2.3.0 ( koa-router@7.2.1 )	✓	23,093
Koa	2.3.0	X	25,571
take-five	1.3.4	✓	28,255
micro ( micro-router )	9.0.0 ( micro-router@2.2.3 )	✓	28,700
connect ( router )	3.6.5 ( router@1.3.2 )	✓	33,753
<b>Fastify</b>	<b>0.29.2</b>	<b>✓</b>	<b>34,613</b>
micro	9.0.0	X	36,522
connect	3.6.5	X	37,810
-			
http.Server	8.6.0	X	39,952

# 数据层 —— MYSQLJS

## 基础驱动

问题：sql注入、开发效率低

## 数据层 —— SEQUELIZE

ORM 支持PostgreSQL, MySQL, SQLite and MSSQL

## 数据层 —— MONGOOSE

nosql 文档型数据库，更贴近js普通对象类型

# 消息队列 —— RESQUE

resque lpush rpop 典型的事件驱动 分布式  
错峰、模块解耦、最终一致性

```
const jobs = {
  'subtract': {
    perform: (a, b) => {
      let answer = a - b
      return answer
    }
  }
}

// 消费者
const worker = new NodeResque.Worker({connection: connectionDetails, queues: ['math']}),
await worker.connect()
await worker.workerCleanup() // optional: cleanup any previous improperly shutdown work
worker.start()
// 生产者
const queue = new NodeResque.Queue({connection: connectionDetails}, jobs)
await queue.connect()
await queue.enqueue('math', 'add', [1, 2])
```

## 前端 —— 面临的问题

- 1.需求很频繁
- 2.开发人员少
- 3.界面较复杂

# 前端 —— JS 响应式框架

react

angular

vue

# 前端 —— UI

bootstrap

element-UI

ant design



# 编码

## 编辑器/IDE

名称	是否收费	断点调试	功能
Webstorm	收费	支持	IDE, 在代码提示、重构方面非常强大, 支持大量语言
Sumblime	收费	不支持	作为编辑器非常友好, 许多前端都很熟悉
Emace/Vim	免费	不支持	号称神的编辑器/编辑器之神, 上古神器
Atom	免费	支持	github出品, 功能强大, 缺点卡顿
Vscode	免费	支持	微软出品, 跨平台, 速度快, 各方面友好

# 编码

## 规范

js 出身在互联网莽荒时代，野蛮生成

nodejs 是 js 的后端版本

项目可维护性 防止低级错误 代码的可阅读性

# 编码

## 规范选择

1. Google Coding-style for JavaScript
2. airbnbjs
3. standardjs

# 白盒检查

## eslint —— 开箱即用

终端

```
→ home.xl9.xunlei.com git:(master) npm run lint

> server@1.10.3 lint /Users/benzhao/Sites/home.xl9.xunlei.com
> eslint --ext .js .

/Users/benzhao/Sites/home.xl9.xunlei.com/middlewares/betacontrol.js
 180:3  error  'a' is not defined  no-undef

* 1 problem (1 error, 0 warnings)
```

集成IDE



The screenshot shows a code editor with a JavaScript file. The code is as follows:

```
173     ...     encode: encode({ max: 2048, maxAge: 3 * 60 * 1000 }, // 3 分钟)
174     ...     type: 'sync'
175     ...   }),
176     ...   'host': options.host,
177     ...   [eslint] 'a' is not defined. (no-undef)
178     ...
179     ...   正在加载...
180     ...   a
181     ...
182     ...   return function (req, res, next) {
183     ...     debug('enter betacontroller');
184     ...     if (res.locals.betacontrol || (req.query && req.query.ignorebeta
185     ...       return next();
186     ...     } else {
```

The tooltip displays the ESLint error message: "[eslint] 'a' is not defined. (no-undef)". Below the error message, it says "正在加载..." (Loading...). The cursor is positioned on line 180, where the variable 'a' is used.

# 测试

## 特性测试

框架: mocha

断言: should

# 特性测试


```
44
45 it('lack para "resources" should return RestError', (done) => {
46   request(server)
47     .post('/api/resource/add' + util.createTestReqParaStr())
48     .type('json')
49     .send({
50       callbackurl: callbackurl
51     })
52     .expect(200)
53     .expect('Content-Type', /application\/json/)
54     .end((err, res) => {
55       if (err) throw err
56       expect(res.body).haveOwnProperty('code').equal(RestError.ERROR_PARAM_INVALID)
57       done()
58     })
59 })
```


## CI —— 自动化测试部分

持续集成的目的，就是让产品可以快速迭代，同时还能保持高质量




# CI —— 自动化测试部分


## express pull request



**Some checks were not successful**[Hide all checks](#)

2 failing and 1 successful checks

×		continuous-integration/appveyor/pr — AppVeyor build failed	<a href="#">Details</a>
×		continuous-integration/travis-ci/pr — The Travis CI build failed	<a href="#">Details</a>
✓		coverage/coveralls — Coverage remained the same at 100.0%	<a href="#">Details</a>

**This branch has conflicts that must be resolved**

Only those with [write access](#) to this repository can merge pull requests.

---

**Conflicting files**

- test/req.param.js
- test/res.attachment.js
- test/res.format.js
- test/res.send.js



# CI —— 自动化测试部分

## express pull request

```
2249 without an api key
2250 ✓ should respond with 400 bad request
2251 with an invalid api key
2252 ✓ should respond with 401 unauthorized
2253 with a valid api key
2254 ✓ should respond user repos json
2255 ✓ should 404 with unknown user
2256 when requesting an invalid route
2257 ✓ should respond with 404 json
2258
2259
2260 833 passing (4s)
2261
2262 =====
2263 Writing coverage object [/home/travis/build/expressjs/express/coverage/coverage.json]
2264 Writing coverage reports at [/home/travis/build/expressjs/express/coverage]
2265 =====
2266
2267 ===== Coverage summary =====
2268 Statements : 98.76% ( 1910/1934 ), 37 ignored
2269 Branches : 94.44% ( 748/792 ), 21 ignored
2270 Functions : 100% ( 267/267 )
2271 Lines : 99.95% ( 1866/1867 )
2272 =====
2273
2274
2275 The command "npm run test-ci" exited with 0.
2276 $ npm run lint
2277
2278 > express@4.16.2 lint /home/travis/build/expressjs/express
2279 > eslint .
2280
2281
2282
```

# CI —— 自动化测试部分

## gitlab —— CI configuration

```
28 # 代码检查
29 lint:
30   stage: test
31   script: npm run lint
32
33 # 单元测试
34 unit:
35   stage: test
36   script: npm run unit
37
38 # 部署测试服务器
39 deploy_dev:
40   stage: deploy
41   tags:
42     - 10.10.34.91-dev
43   only:
44     - develop
45   script:
46     - rsync -av --delete-after --exclude-from=/data/shell/home.xl9.xunlei_exclude.list . $DEV_RSYNC_PATH
47     - chmod -R 755 $DEV_RSYNC_PATH
48     - chown -R nobody:nobody $DEV_RSYNC_PATH
49     - find $DEV_RSYNC_PATH -type f -exec chmod 644 {} \;
50     - cd $DEV_RSYNC_PATH
51     - npm install
52
53 # 部署测试服务器
54 deploy_pro:
55   stage: deploy
56   only:
57     - master
58   script:
59     - rsync -av --delete-after --exclude-from=/data/shell/home.xl9.xunlei_exclude.list . $DEV_RSYNC_PATH
60     - chmod -R 755 $DEV_RSYNC_PATH
61     - chown -R nobody:nobody $DEV_RSYNC_PATH
62     - find $DEV_RSYNC_PATH -type f -exec chmod 644 {} \;
63     - cd $DEV_RSYNC_PATH
64     - npm install
```

# 压力测试

## 单进程

### 1. apache ab

```
ab -c 500 -n 50000 http://hacking.pub:3000/
```

```
Hostname:      hacking.pub
Server Port:   3000
Document Path: /
Document Length: 6 bytes
Concurrency Level: 500
Time taken for tests: 12.093 seconds
Complete requests: 50000
Failed requests: 0
Write errors: 0
Total transferred: 4050000 bytes
HTML transferred: 300000 bytes
Requests per second: 4134.65 [#/sec] (mean)
Time per request: 120.929 [ms] (mean)
Time per request: 0.242 [ms] (mean, across all concurrent requests)
Transfer rate: 327.06 [Kbytes/sec] received
```

# 压力测试

## 2. loadrunner

- a. 高规模适应性的自动负载测试工具
- b. 预测系统行为，优化性能
- c. 模拟实际用户的操作行为和实行实时性能监控

# 部署、监控

# 发布大致流程

1. 登录发版机，执行发布脚本
2. 发布脚本拉取指定分支代码，默认master，触发npm安装依赖包
3. 发布脚本过滤敏感、不必要文件，如打包构建文件、git相关文件
4. 发布脚本触发rsync服务，将代码同步至线上机器
5. 线上机器守护进程watch代码变动，自动无缝重启

实际情况还要复杂一点

# 进程生命周期管理

pm2: 近期代码merge; 23k star

forever: 一年以上时间没有变化; 10k star

# PM2-LIST

unitech@t450: ~

>>> pm2 ls

App name	id	mode	pid	status	restart	uptime	cpu	mem	watching
api	0	cluster	27287	online	0	2m	0%	35.5 MB	disabled
api	5	cluster	27308	online	0	2m	0%	32.4 MB	disabled
api	6	cluster	27348	online	0	2m	0%	33.7 MB	disabled
front	3	fork	27303	online	0	2m	0%	26.5 MB	enabled
healthcheck	2	fork	0	stopped	0	0	0%	0 B	disabled
mailer	4	fork	27309	online	0	2m	0%	26.1 MB	disabled
worker	1	fork	27292	online	0	2m	0%	24.9 MB	disabled

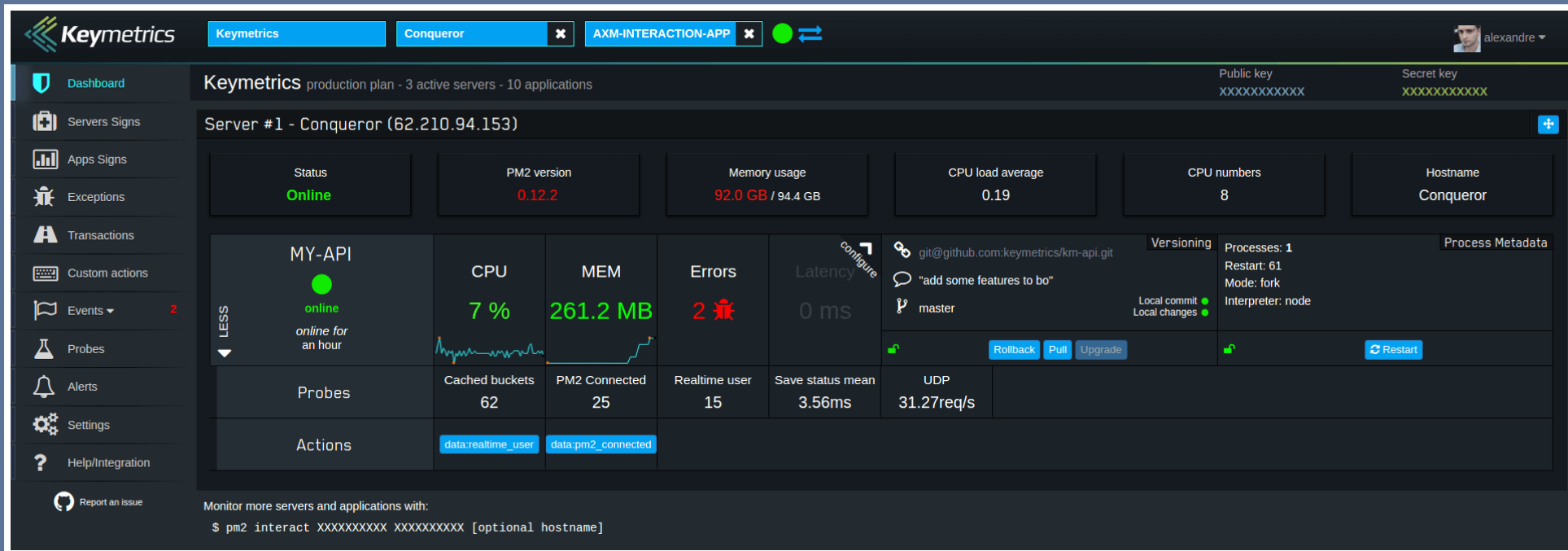
Use `pm2 show <id/name>` to get more details about an app



**为什么是无缝重启，不是热更新？**

cluster模式，依次重启，当新instance重启成功才把旧worker杀死

# 监控 -- KEYMETRICS

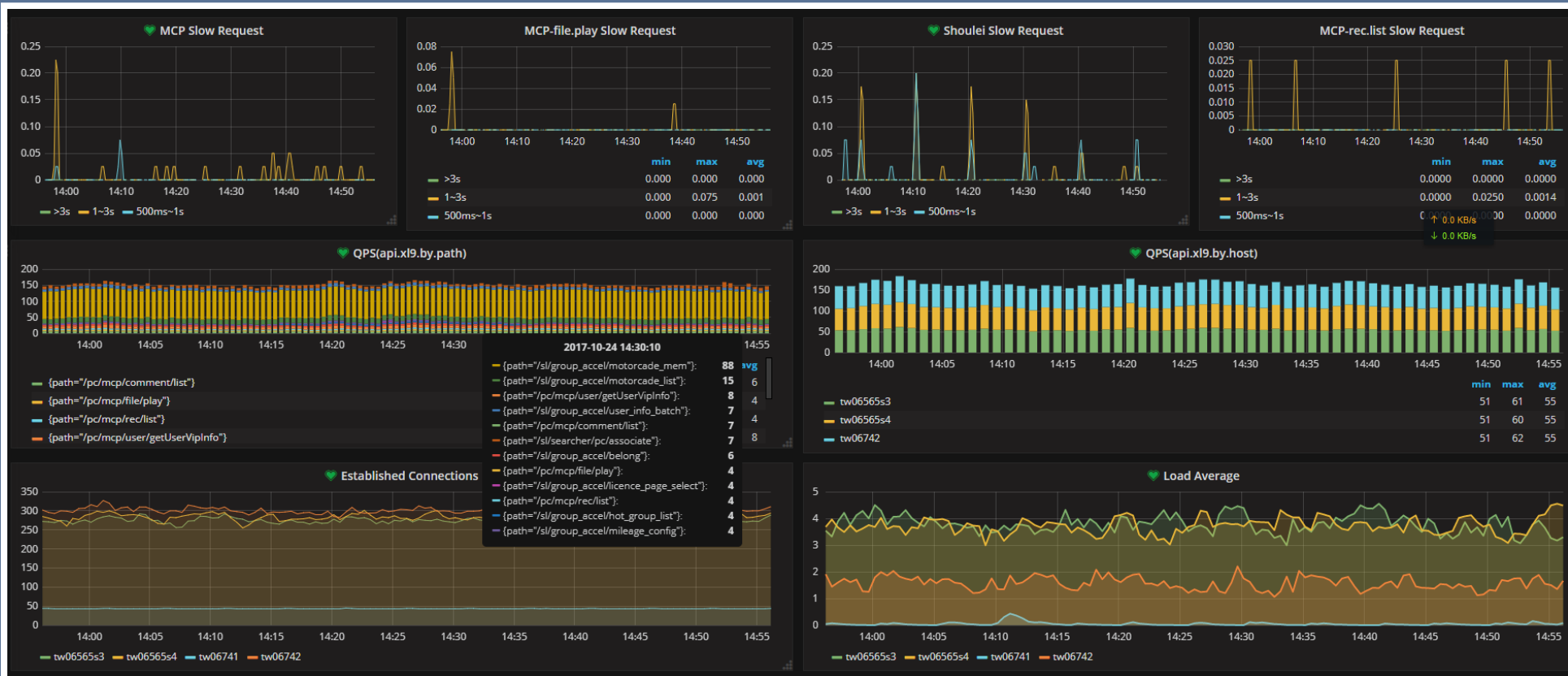


免费版 适合小型个人网站，创业公司

# 监控 -- PROMETHEUS

Prometheus是一套开源的监控 & 告警 & 时间序列数据库的组合

# 监控 -- PROMETHEUS + GRAFANA



# 监控 -- PROMETHEUS

```
// 统计上报
var Monitor = require('@xunlei/node-stats')
app.use(Monitor.middleware({
  'name': 'audit.yl',
  'type': 'textfile', //textfile|push|pull
  'interval': 1000, //textfile|pushgateway ms
  'path': path.join(os.tmpdir(), '/node_exporter/') // 'http://localhost:9092/'
}, function (err) {
  logger.error(err)
}))
```

**THANKS**