

从刀耕火种，铁犁牛耕，迈向机械化生产

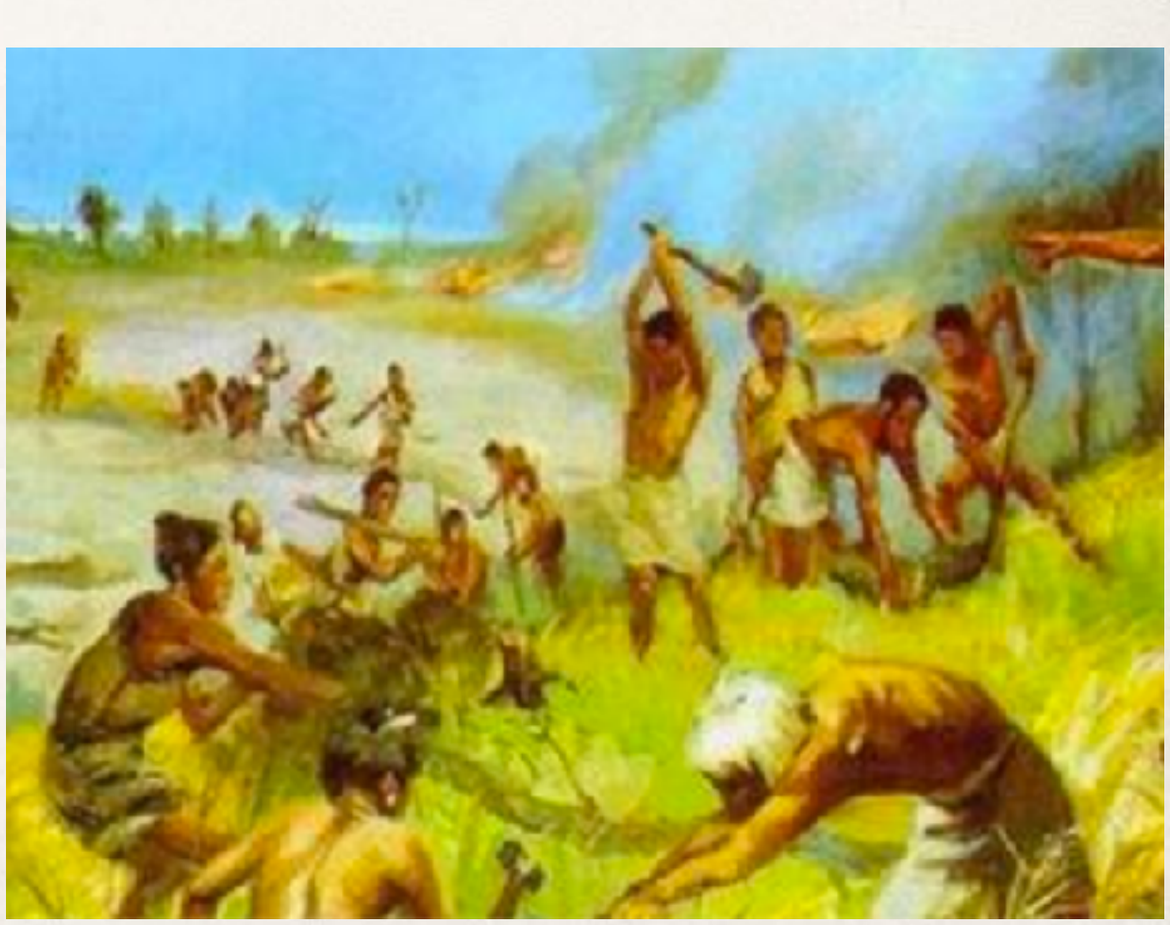
Egg.js 在阿里巴巴集团的的实践运用 egg

庄少/ngot

关于我

- 庄少/ngot
- Egg.js 核心开发者；fibjs 核心开发者；
- Follow me at :
  - ngot
  - ngot
  - Ngot_ftd

刀耕火种



裸用社区框架

- 裸用 koa, express 等
- 各团队独自野蛮发展，开发规范混乱

问题

- 重复建设
- 无法复用
- 跨团队合作异常困难
- 中间件对接困难

铁犁牛耕

BU级别定制框架出现

- 少数 BU 出现了基于 Koa 的定制框架
- 开发了 cnpm，并且搭建了内部私有 npm
- 围绕各个框架，生态开始繁荣

问题

- 从公司整体看，无法形成合力
- 基础组件复用度低
- 跨 BU 合作依然困难重重

迈向机械化生产

天下归一

- 联合整个公司制定 Node.js 企业级的开发标准
- 基于标准规范开发对应的 Web 框架(Egg.js)
- 开源回馈社区

Egg.js

为企业级框架和应用而生

express vs koa

- 基于 Generator / Async 的中间件(同步写法更适合企业应用)
- 核心精简（req/res/ctx），易于扩展
- 掌握核心科技(苏千，死马 koa 核心开发)

换位思考

- 业务开发
  - 易用的框架，丰富的库，能够快速搞定业务才是王道
- 架构师
  - 降低团队成员学习门槛，基础组件沉淀复用，公司内部生态建设，打通开源社区

设计原则

- 追求规范和共建
- 微内核，可扩展
- 约定优于配置

Egg.js 项目一览

```
showcase
├── app
|   ├── controller (控制器)
|   |   └── home.js
|   ├── service (业务逻辑)
|   |   └── github.js
|   ├── view (模板)
|   |   └── index.tpl
|   ├── public (静态资源)
|   |   └── main.css
|   └── router.js (路由)
├── config (配置)
|   ├── config.default.js
|   ├── config.prod.js
|   └── plugin.js
├── test (单元测试)
├── README.md
└── package.json
```

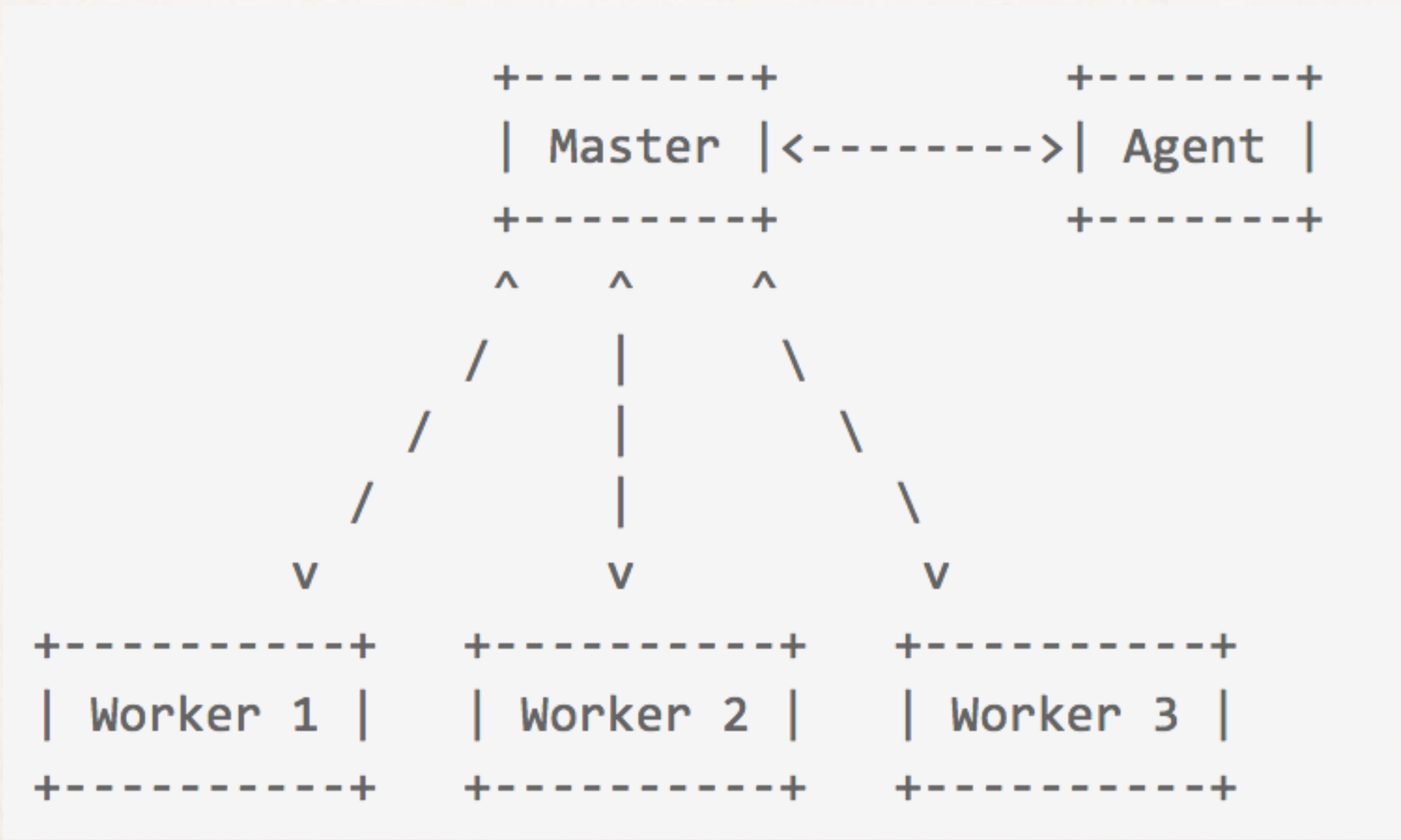
Demo

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

```
npm i egg-init -g
```

```
$ egg-init dest
? Please select a boilerplate type (Use arrow keys)
❯ simple - Simple egg app
  plugin - egg plugin
```

进程模型



渐进式开发

- 最初是状态
- 插件的雏形
- 抽成独立插件
- 沉淀到框架

Demo

```
mkdir -p lib/plugin

cd lib/plugin

egg-init --type plugin  egg-nodeparty-hello
```

Async Function

- Node.js >= 7.6

```
// app/controller/post.js
module.exports = app => {
  class PostController extends app.Controller {
    // 从 * create() 换成 async create()
    async create() {
      const { ctx, service } = this;
      const createRule = {
        title: { type: 'string' },
        content: { type: 'string' },
      };
      // 校验参数
      ctx.validate(createRule);
      // 组装参数
      const author = ctx.session.userId;
      const req = Object.assign(ctx.request.body, { author });
      // 调用 service 进行业务处理
      // 从 yield 换成 await
      const res = await service.post.create(req);
      // 设置响应内容和响应状态码
      ctx.body = { id: res.id };
      ctx.status = 201;
    }
  }
  return PostController;
}
```

定制属于你自己的框架

- 规范自己团队的开发体系
- 开箱易用

如何定制?

- 扩展
- 暴露
- 文档

Demo

```
egg-init --type framework  nodeparty
```

```
cnpm i egg-doctools --save-dev
```

npm scripts 添加：

```
"doc-server": "doctools server --external https://github.com.cnpmjs.org/eggjs/egg.git"
```

工具链

- egg-bin
- egg-mock
- egg-scripts
- egg-doctools

阿里内部落地

一线开发者

基于上层框架开发的应用

团队架构师

适合特定团队业务场景的上层框架 Framework

蚂蚁 Chair | UC Nut | 集团 Begg | ...

Plugin
插件生态

Tool
工具链

社区生态

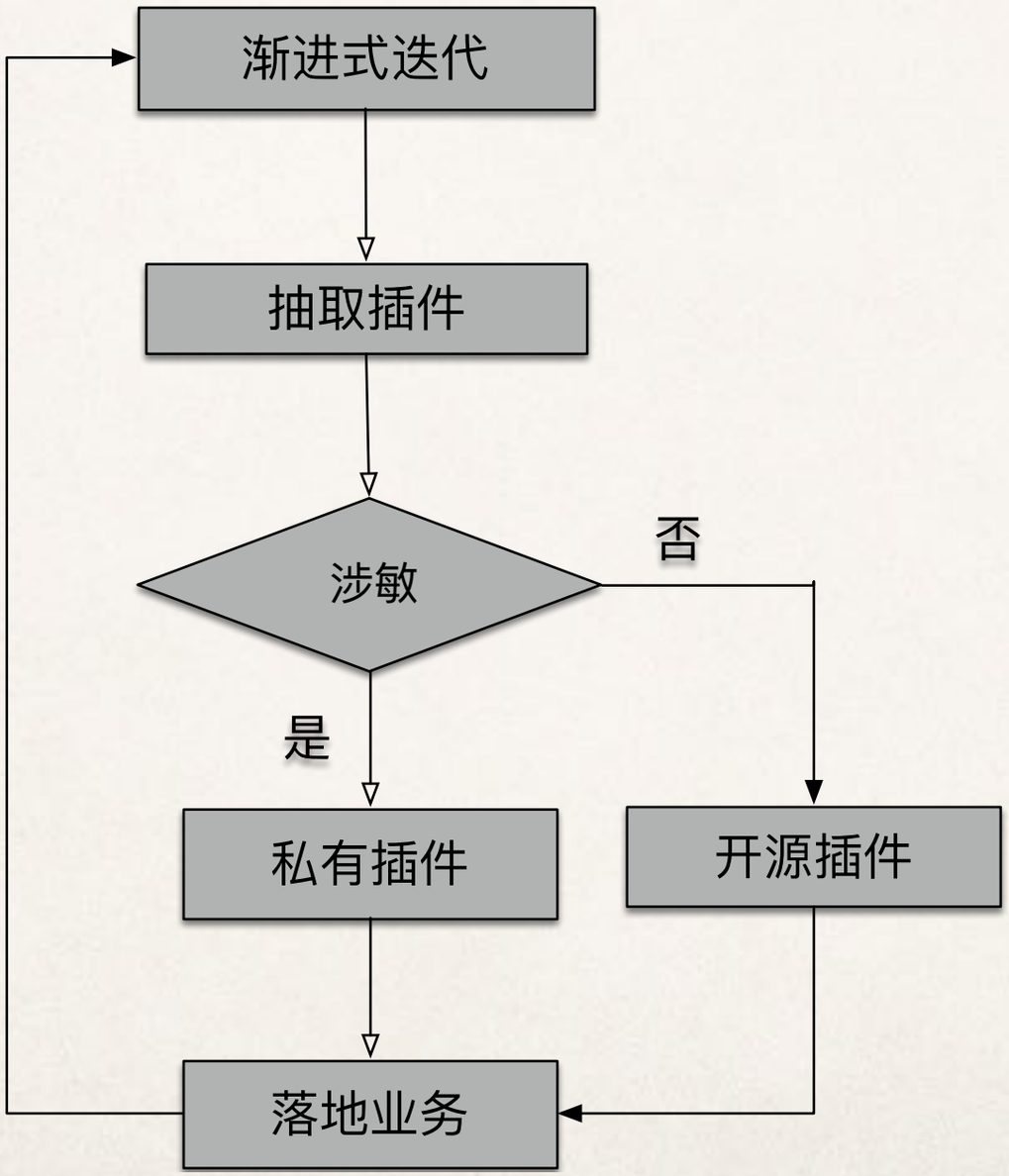
Egg

Specification 一套规范和约定

Koa

Node.js

与社区联动



hengfei.zhf@alibaba-inc.com

谢谢！