

同步代码

```
exports.list = async (ctx, next) => {  
  console.log(ctx.method + ' /students => list, query: ' +  
  try {  
    let students = await Student.getAllAsync();  
  
    await ctx.render('students/index', {  
      students : students  
    })  
  } catch (err) {  
    return ctx.api_error(err);  
  }  
};
```

调试

VSCode

单个文件调试

远程调试

固定文件调试

跨进程调试



调试

变量

本地

- req: IncomingMessage {_read...
- res: ServerResponse {domain...
- this: Server {domain: null,...

脚本

全局

监视

调用堆栈 已于断点 暂停

- (匿名函数) hello_node.js 4
- emitTwo events.js 106
- emit events.js 191
- parserOnIncoming
- parserOnHeadersComplete

断点

- 所有异常
- 未捕获的异常
- hello_node.js /Users/s... 4

```

JS hello_node.js x
1  var http = require('http');
2
3  http.createServer(function (req, res) {
4  |  res.writeHead(200, {'Content-Type': 'text/plain'});
5    res.end('Hello Node.js\n');
6  }).listen(3000, "127.0.0.1");
7
8  console.log('Server running at http://127.0.0.1:3000/');

```

问题 输出 调试控制台 终端

1: curl

```

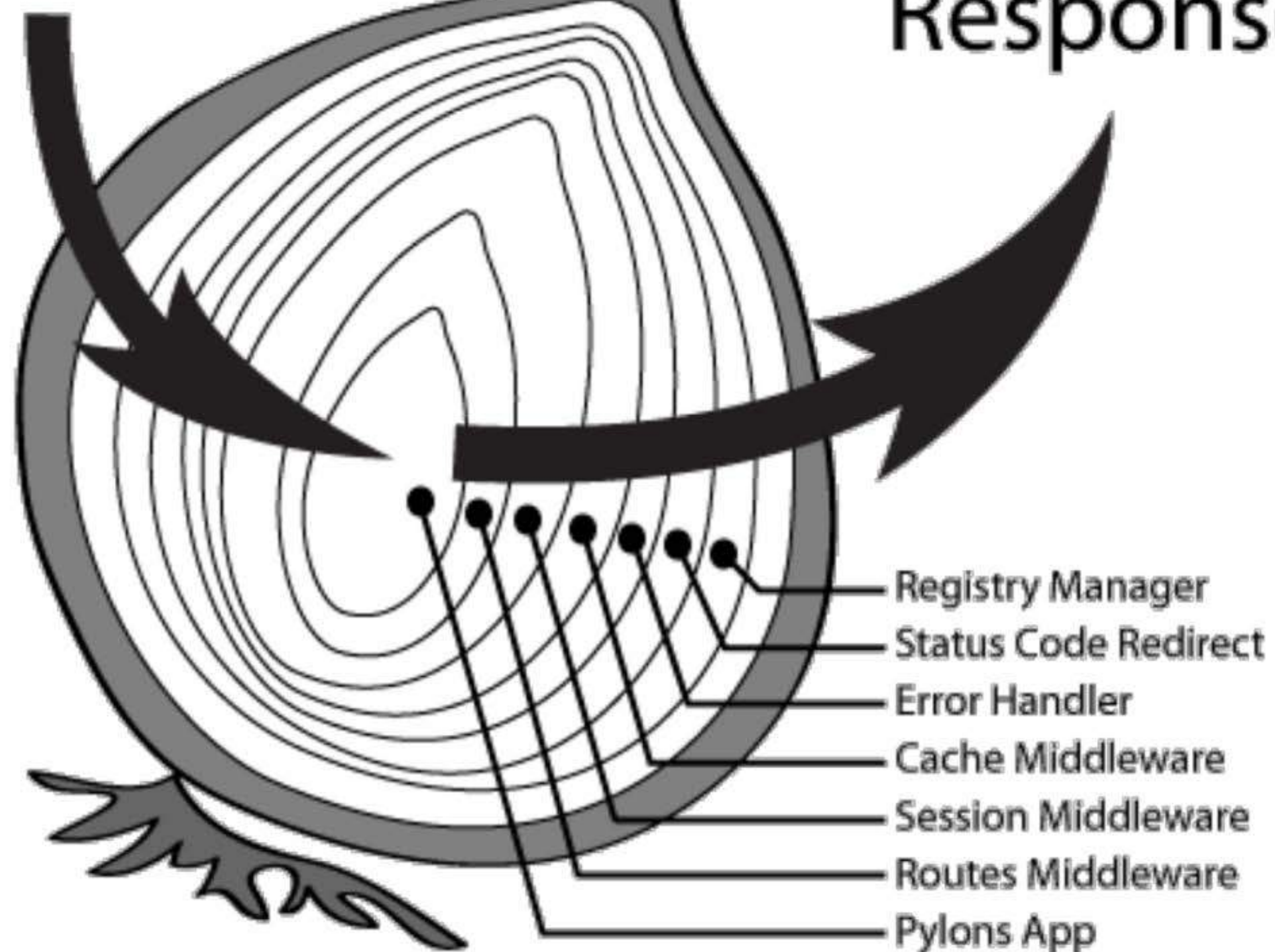
~ curl http://127.0.0.1:3000/
[]

```

框架名称	
Express	简
Derby.js && Meteor	同
Sails、Total	同
MEAN.js	同
Hapi和Restfy	同
ThinkJS	同
Koa	类

Request

Response



点评

看上去更个

的一个标志

蹭了热点

行独立分类

各路，对于

资源管理器

- 打开的编辑器
 - 左侧
 - JS index.js routes
 - JS app.js
 - 右侧
 - JS app.js
 - JS index.js routes
- KOA2-DEMO
 - bin
 - public
 - routes
 - JS index.js
 - JS users.js
 - views
 - JS app.js
 - package.json

```

1  const Koa = require('koa')
2  const app = new Koa()
3  const views = require('koa-views')
4  const json = require('koa-json')
5  const onerror = require('koa-onerror')
6  const bodyparser = require('koa-bodyparser')
7  const logger = require('koa-logger')
8
9  const index = require('./routes/index')
10 const users = require('./routes/users')
11
12 // error handler
13 onerror(app)
14
15 // middlewares
16 app.use(bodyparser({
17   enableTypes: ['json', 'form', 'text']
18 })))
19 app.use(json())
20 app.use(logger())
21 app.use(require('koa-static')(__dirname + '/'))
22
23 app.use(views(__dirname + '/views', {
24   extension: 'pug'
25 })))
26
27 // logger
28 app.use(async (ctx, next) => {
29   const start = new Date()
30   await next()
31   const ms = new Date() - start
32   console.log(`${ctx.method} ${ctx.url} - ${ms}`)

```

```

1  const router = require('koa-router')()
2
3  router.get('/', async (ctx, next) => {
4    await ctx.render('index', {
5      title: 'Hello Koa 2!'
6    })
7  })
8
9  router.get('/string', async (ctx, next) => {
10   ctx.body = 'koa2 string'
11 })
12
13 router.get('/json', async (ctx, next) => {
14   ctx.body = {
15     title: 'koa2 json'
16   }
17 })
18
19 module.exports = router
20

```



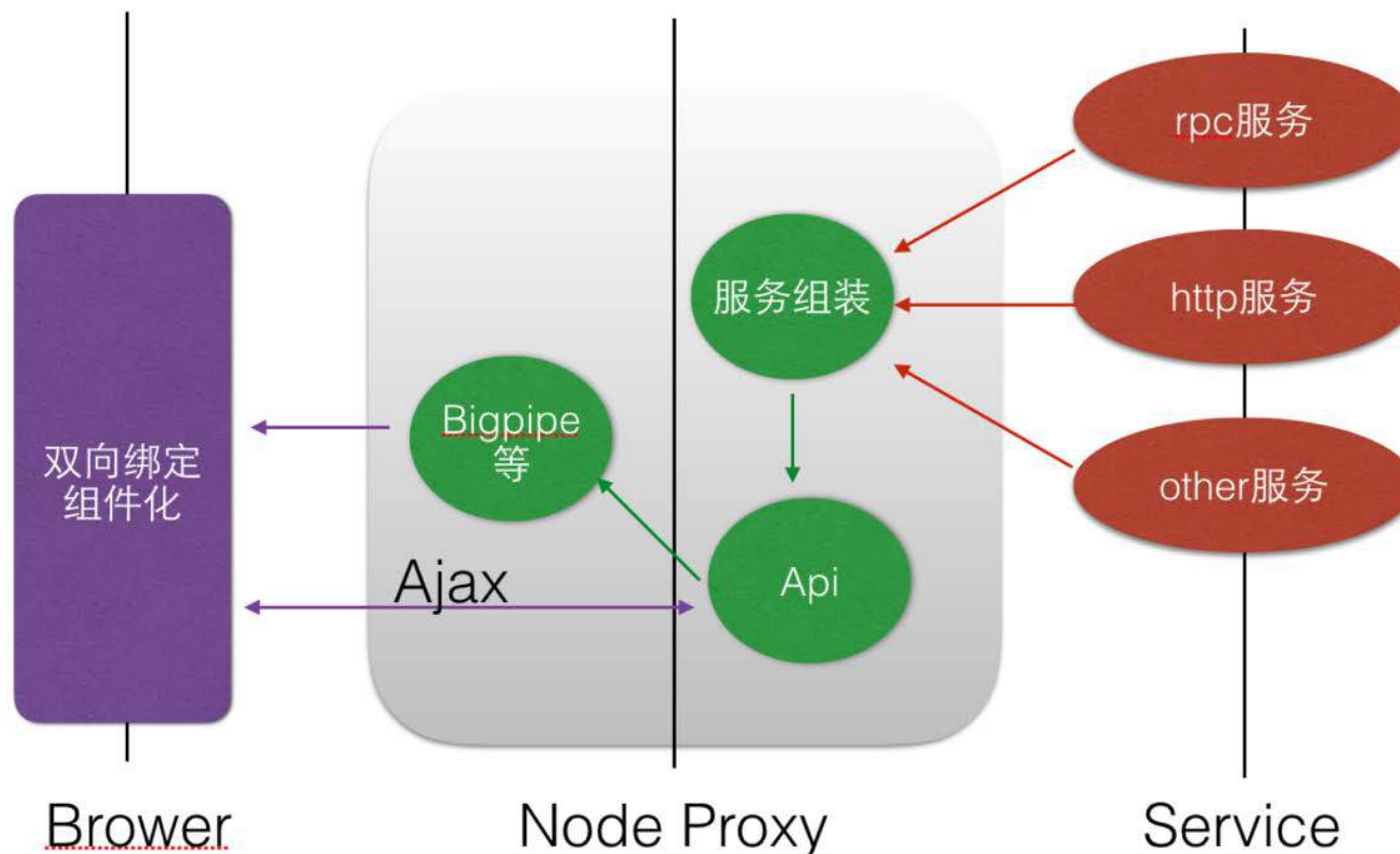
4、我眼中Node全栈



假设前面有2辆车，你必须撞一个，你选哪个？



Api: Proxy层



成为更好的自己，将终身学习进行到底

减少沟通成本

可攻可受

享受快乐



闲时要有吃紧的心思，忙里要有偷闲的乐趣

|| 你能做的可以更多

你会的越多解决越容易





AI时代的移动技术革新

Era of AI: Innovations in Mobile Technologies



APICloud

IT大咖说
知识共享平台

少抱怨，多思考，未来更美好

谢谢观看
THANKS

APICloud