一次HTTP请求响应过程

GET /schema HTTP/1.0 → http://127.0.0.1:3000/schema

```
cat.middleware → 注册 ctx.cat
                                                                          app-proto 内置中间件
request-transaction → 记录当前HTTP请求Transaction
logger → 注册 ctx.$logger, 与Cat集成 (error\event 会同步到cat.dp)
mta.koaMiddleware → 统计本次请求network信息(次数 & 耗时)
                                                                           $global.js 全局中间件
sayHey → console.log('sayHey', 'hello world')
                                                                           路由中间件
schema-logger \rightarrow console.log(`\{ctx.method\} \{ctx.url\} - \{ms\}ms`)
controller: (ctx) => {} → initStateData + ctx.render() → ctx.body = tpl.toHtml() -
                                                                             Controller 中间件
```

筛搜页举例: https://channel-zhenguo.meituan.com/filter-search

```
cat.middleware → 注册 ctx.cat
request-transaction → 记录当前HTTP请求Transaction
logger → 注册 ctx.$logger, 与Cat集成 (error\event 会同步到cat.dp)
mta.koaMiddleware → 统计本次请求network信息(次数 & 耗时)
 'method-control' → 过滤掉非 POST、GET 请求
 'sentry-error' → 请求异常捕获 & 上报 Sentry
 'error-stack' → 供调试使用,打印异常堆栈
 'body-parser' → koajs@bodyparser
 'compress' → koajs@compress
```