

远程团队时间管理与研发管理

刘昊臻
Eolinker CEO

Eolinker 是做什么的？

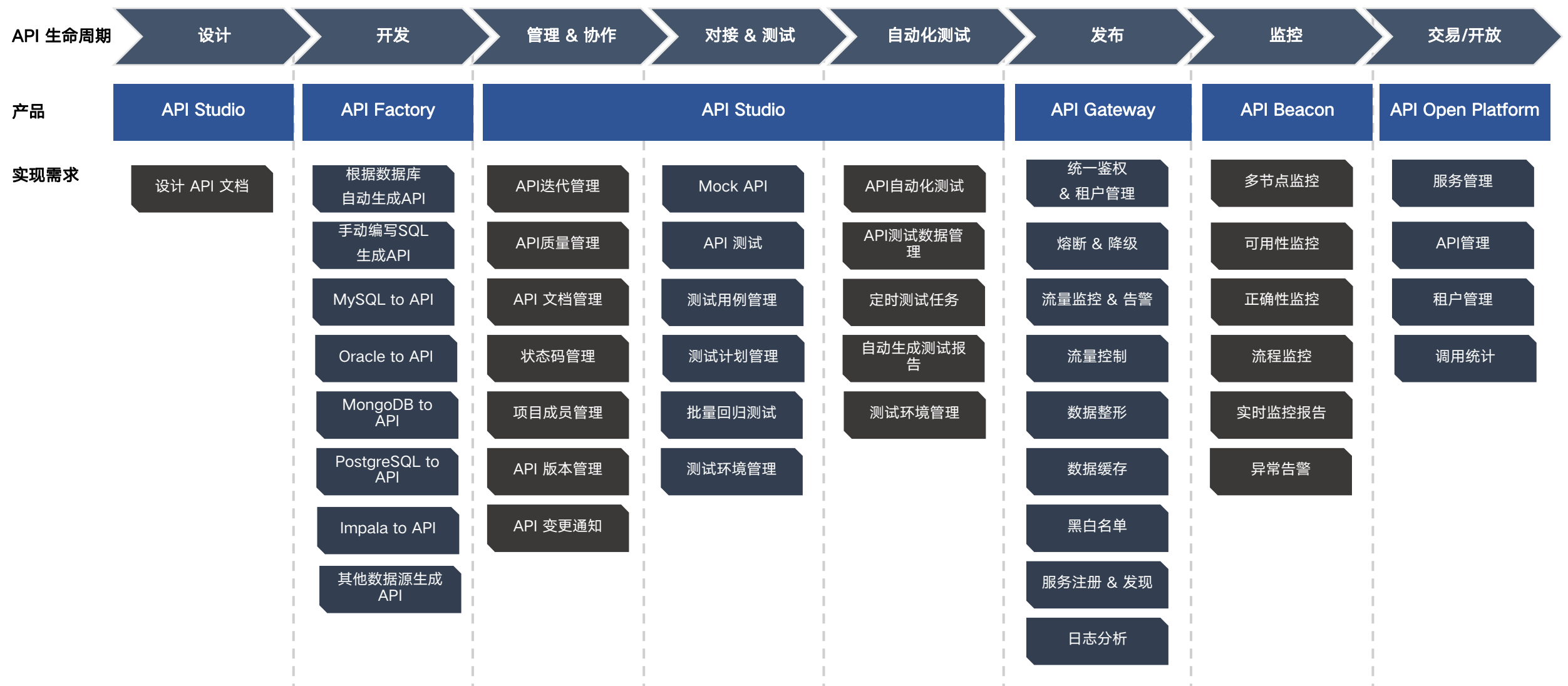
Eolinker 是一家专注提供 API 生命周期管理产品的公司，旗下产品线覆盖API的生成、API研发管理、API自动化测试、API监控、API微服务网关、API开放平台等应用场景。

API 生成 | API 研发管理 | API 自动化测试 | API 监控 | API 微服务网关 | API 开放平台

- 首家为 ITSS（中国电子工业标准化技术协会）制定API研发管理行业规范的企业
- 提供SaaS以及本地化部署产品
- 目前已为超过3000家企业提供专业的API研发管理服务
- Eolinker 累计托管超过700万在线API服务，每日执行上千万次API测试及自动化测试请求

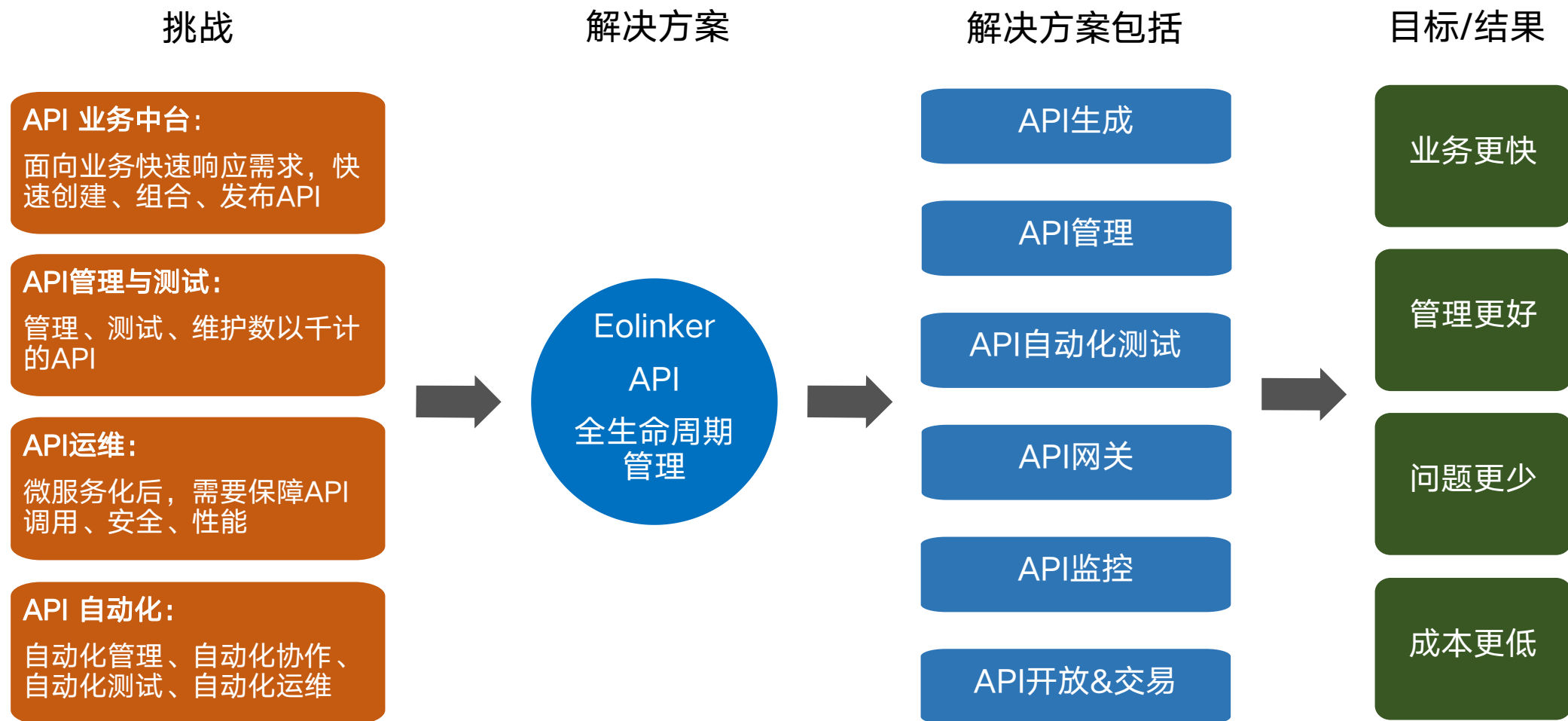


Eolinker API 全生命周期管理产品线



由业务增长、需求快速变化带来的挑战

需求总是在变、业务总感觉慢、运维成本总在增加，即使996依然无法满足？



构建企业内外部API生态，获取API经济价值

Eolinker 为企业打造一站式、自动化、高性能、高稳定性的 API 服务生产、管理、测试、发布、运维平台



目录

1. 如何打造“高效且自由”的研发团队
2. 如何协调团队与个人、工作和生活的时间
3. 话锋一转，聊聊 API 研发管理中的效率损耗
4. 通过 API 管理和自动化测试实现“文档驱动开发（DDD）”、“测试驱动开发（TDD）”？
5. 总结

打造高效且自由的研发团队

1. 研发工作本质是“智力劳动”，码农并不只是复制粘贴的搬运工。长期使用“体力劳动”的规则管理研发团队，会使得积极性降低，从而影响效率和稳定性。
2. 传统办公场景是基于“监督”，而远程办公是基于“信任”，“体力劳动”的规则不再适用于远程办公。
3. 高效有几个条件（类比如玩游戏）：
 - 稳定的工作环境（机器好、网不卡）
 - 清晰的工作目标（打啥怪、有啥等级和装备要求）
 - 经常获取有价值的反馈（打赢有奖励，打输会懊恼）
4. 自由：
 - 高效是自由的前提，只要能够按时按量完成工作目标并且不影响公司文化氛围，怎么工作只是员工的“隐私问题”。

我们的做法

1. 研发团队完全远程，偶尔有事回办公室，减少路途的时间和精力损耗（花更多时间写代码不香吗，为什么要花时间开车/挤地铁？）
2. 按需进行早会、晚会、临时会
3. 尽量少但够用的工具（减少工具之间切换的时间和操作成本）：
 - 即时通讯：企业微信、QQ
 - 研发管理：Gitlab（代码）、Eolinker（API、测试、进度管理、开发协作等）、其他
 - 其他：群晖NAS（公司用的私有云盘，上面有各种日程管理、办公套件、共享资源等等）、滴答清单（自我管理）
4. 打造全功能团队：
 - 适当地身兼多职可以激发积极性：研发抽空兼职客服，了解客户需求才能更好开发代码，还能锻炼说话能力
 - 高度的信息透明可以减少沟通成本：知道自己在做什么，知道别人在做什么，知道自己怎么帮助别人
 - 能自动化的不手动：减少重复劳动，减少人为出错
 - 能看/写文档的不说话：减少沟通成本，方便回顾历史

如何协调团队与个人、工作和生活的时间

远程办公由于每个人所处办公环境不统一，导致工作时段和时长很难统一。因此通过制度规定劳动者的时间，不如帮助他们了解工作的目标、时限，以及如何达成目标。团队成员会自己调整工作时间。

毕竟，规定工作时间却无法按时完成目标 = 没规定时间

因此我会在认为“信息出现不透明”，或出现效率低下的“趋势”时，问成员以下问题：

1. 完成目前的工作预计需要多久？
2. 这么多项工作，哪一项更重要？为什么？产出是什么？
3. 近期是否有其他的个人安排可能导致协作效率降低？（生娃/带娃/生病/失恋/其他不可抗力）
4. 举行会议保证其他相关成员也能了解以上信息并调整自己的工作时间安排

话锋一转，聊聊API研发管理中的效率损耗

研发团队用在沟通、测试、管理 API 中的时间 和 用在开发项目代码上的时间 是差不多的。

以下和API相关的问题广泛出现：

- API 文档不清晰而不知道怎么对接和测试，需要反复和后端沟通？
- 遇到历史遗留的API，没有文档只能看代码？
- 前端和测试需要等待API开发完成才能继续进行工作？
- API 变更了不及时通知相关人员跟进？改动了哪里也不清楚？
- API 文档和测试是两套系统，来回切换并且无法同步数据？
- API 测试用例无法复用，API一改就需要重新测试，导致测试时间不足，测试范围不够？
- API 自动化测试需要写脚本，学习成本、时间成本、维护成本高？
- ...

API研发的文档驱动和测试驱动

文档驱动开发 指的是在开发之前先把文档写好，明确功能需求、入参出参定义、异常情况处理等之后再行开发。这就好比我们在做题之前需要先了解清楚题目要求，否则不审题就下笔很容易导致最后返工。

测试驱动开发 指的是在开发之前先把测试方案/用例写好，只开发能够顺利通过测试的功能，如果测试不通过则持续进行改进。这就好比我们考试前会先了解考试通过的标准，没有标准乱答一通肯定没有好结果。

简单地说：

1. 用标准文档代替口头约定和笔记文档，让开发、测试、运维、协作有迹可循；
2. 快速用测试结果推动开发进度，让团队沟通更充分、管理有事实依据，实现敏捷开发。



文档驱动：解决协作中目标不清晰的问题

文档驱动的目的：用标准文档代替口头约定和笔记文档，让开发、测试、运维、协作有迹可循；

缺少文档的情况下，沟通会类似这样：

- A：之前你说这个功能要这样做的，怎么现在不对了？
- B：什么时候的事情？
- A：就上周二呀，你在钉钉说这个不能这样改，所以跟我远程语音了一番。
- B：我查一下聊天记录。。。但是真的不记得为啥要这样做了。最近熬夜有点头疼。。。
- A：所以现在到底是怎么样？
- B：你等一下，我问一下C
- C：嗯？？？

基于文档的情况下，沟通会类似这样：

- A：之前你说这个功能要这样做的，怎么现在不对了？
- B：看一下文档，后面我和C也沟通过了。
- A：噢的确是，我记错了，应该是文档写的这个才对。
- C：所以养成以文档为准的习惯就行~

API文档驱动的例子

场景：前端开发已经对接完API，将当前API状态改为待测试，并且通知相关测试人员进行测试。

POC

POC_EOLINKER

API 管理与测试 / 示例项目 / AMS DEMO PROJECT / API

在 API 管理与测试 内搜索

下载桌面端

帮助

返回项目列表

项目

项目概况

API

API 测试

API 测试用例

API 迭代计划

状态码文档

项目文档

公共资源管理

项目环境

数据结构

通用函数

管理

权限管理

项目设置

日志

返回API列表

编辑

复制

删除

API

测试

测试用例

Mock API

未设置环境

HTTP

GET

对接

已关注

编辑

变更通知

编辑历史

评论

测试

/user/login

用户登录

分组: 用户系统

负责人: 无

创建者: POC_EOLINKER

最后编辑人: POC_EOLINKER

更新时间: 2020-02-05 02:39:09

请求头部

生成预览数据

标签	必填	内容	全部展开
Content-Type	是	application/x-www-form-urlencoded	

REST参数

参数名	类型	必填	说明	全部展开
user_token	[string]	是		展开

Body 请求参数

JSON

最外层结构为: Object

生成预览数据

参数名	类型	必填	说明	全部展开
user_info	[object]	是	用户信息	
user_name	[string]	是	用户名	展开
user_psw	[string]	是	用户密码	展开

返回参数

JSON

最外层结构为: Object

生成预览数据

参数名	类型	必含	说明	全部展开
-----	----	----	----	------

测试驱动：解决协作中结果反馈的问题

文档驱动的目的：快速用测试结果推动开发进度，让团队沟通更充分、管理有事实依据，实现敏捷开发。

缺少测试驱动的情况下，沟通会类似这样：

- A：我这边开发完了，一会你有空的时候测一下。
- B：你先自己测一下，我这边还有个需求在测试着。
- A：但我这边还有新的功能要开发呀，明晚要上线了啊啊啊。。。。而且我测的结果又不一定准。
- B：好吧，我先忙完这个。

第二天

- A：测试完了么？
- B：。。。我还没搞完我这边，需求一改就得重新测。

基于测试驱动的情况下，沟通会类似这样：

- A：我这边开发完了，然后用你之前写的测试用例跑了一遍没有问题。
- B：ok，我这边还有个新的需求在写测试用例，一会我检查一下你的结果。

API测试驱动的例子

场景：后端已经开发完成API，自行使用测试人员写好的测试用例对API进行批量测试，排查错误。

POC EOLINKER

API 管理与测试 / 示例项目 / AMS DEMO PROJECT / API

在 API 管理与测试 内搜索

下载桌面端

帮助

返回项目列表

项目

项目概况

API

API 测试

API 测试用例

API 迭代计划

状态码文档

项目文档

公共资源管理

项目环境

数据结构

通用函数

管理

权限管理

项目设置

日志

返回API列表

编辑

复制

删除

API

测试

测试用例

Mock API

HTTP

GET

测试

已关注

编辑

变更通知

编辑历史

评论

测试

/user/login

用户登录

分组: 用户系统

负责人: 无

创建者: POC_EOLINKER

最后编辑人: POC_EOLINKER

更新时间: 2020-02-13 19:48:34

请求头部

生成预览数据

标签	必填	内容	全部展开
Content-Type	是	application/x-www-form-urlencoded	

REST参数

参数名	类型	必填	说明	全部展开
user_token	[string]	是		展开

Body 请求参数

JSON

最外层结构为: Object

生成预览数据

参数名	类型	必填	说明	全部展开
user_info	[object]	是	用户信息	
user_name	[string]	是	用户名	展开
user_psw	[string]	是	用户密码	展开

返回参数

JSON

最外层结构为: Object

生成预览数据

参数名	类型	必填	说明	全部展开
-----	----	----	----	------

经过文档驱动和测试驱动后的研发流程



总结 (You can take away)

- 传统办公场景是基于“监督”，而远程办公是基于“信任”，“体力劳动”的规则不再适用于远程办公。研发团队需要向着“高效且自由”的方向发展。
- 高效是自由的前提，只要能够按时按量完成工作目标并且不影响公司文化氛围，怎么工作只是员工的“隐私问题”。
- 通过制度规定劳动者的时间，不如帮助他们了解工作的目标、时限，以及如何达成目标。成员会自行调整工作时间安排。
- 研发团队用在沟通、测试、管理 API 中的时间 和 用在开发项目代码上的时间 是差不多的。进行 API 管理和自动化测试可以减少大量成本，并且对实现远程办公有巨大帮助。
- 最佳实践：通过API管理和自动化测试来实现研发团队的文档驱动和测试驱动。

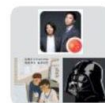
谢谢



刘昊臻-Jack liu
广东 广州



扫一扫上面的二维码图案，加我微信



EOLINKER



该二维码7天内(2月19日前)有效，重新进入将更新



Eolinker

API 全生命周期管理解决方案

eolinker.com

在线注册即可免费使用 SaaS 服务

vip.eolinker.com

申请即可免费试用本地部署产品



400-616-0330

欢迎来电咨询 (9:30 - 18:00)



Eolinker

API 全生命周期管理解决方案

鲲鹏说



Geekbang

极客邦科技

T00 鲲鹏會

THANKS

—— 高端技术人员学习型社交平台 ——



TGO 鲲鹏会 × 极客时间 LIVE

疫情当前，互联网公司 如何做好技术战略规划

沈剑 快狗打车 CTO、“架构师之路”公众号作者

分享时间：2月14日 20:00–21:00



TGO 鲲鹏会

汇聚全球科技领导者的高端社群

🏢 12 个城市成立分会

👤 会员超 1000 人

使命
Mission

助力会员成为卓越的
科技领导者

愿景
Vision

构建全球化的有技术背景的
优秀人才同侪学习成长平台



扫描二维码，了解更多内容