# W-IO版本更新文档

### 版本 1.4

1. 解决分布式系统断线重连后访问异常的问题
2. 添加全链路日志监控，并命名为狼眼
3. 将分布式系统核心代码封装，将数据库连接池加入，采用HiKariCP连接池
4. 解决其他bug
5. 前端定为LayerUI

### 版本 1.3

1. 添加并完善对websocket的支持，websocket由入库把持，业务服务器不对websocket进行长连，由逻辑连接来做，支持业务服务器推送消息。
2. 前端封装websocket发送，仿照jquery的ajax封装，采用message和close回调函数处理消息
3. http接口，server接口，websocket分离，非同类接口不能访问，例如http请求不能访问server接口，但接口内部还是可通过server接口进行访问，本次分离后，http服务，server服务，websocket服务可以分开部署（如有需要），分开部署后，业务服务器需要额外添加OtherServer类及里面的handle，复制一套换个端口就行
4. 添加静态资源文件缓存，提高静态文件访问效率。
5. 添加入口缓存支持，当前版本缓存由业务服务器主推，缓存采用netty的堆外内存工具类进行保存，可大幅度提高数据传输效率，当前测试http缓存数据后性能可提高5倍以上，server缓存后性能最低7倍（个人pc虚拟机测试），当判断路径拥有缓存后将不在执行到后台，当然，这只是初版，后续将完善更多缓存功能，目前只有http和server接口支持缓存，websocket由于是后端主推，所以不考虑支持缓存

### 版本 1.2

对比版本1.1版本完善了入口，优化了注册中心代码

1. 优化了注册中心代码
2. 添加http文件上传下载支持，
3. 添加服务调用文件传输和返回文件的支持
4. 添加对cookie的支持
5. 建议一次不要添加太多文件，视文件大小和内存来定

### 版本 1.1

对比版本1.0，1.1版本完善了入口，业务服务器，事务调度器的配置文件，将更多内容加入到配置文件中，并修复了一些bug

1. 完善了配置文件
2. 添加路径负载均衡，可以对controller方法的最大请求数进行配置，并将路径负载均衡和全局负载均衡分离，路径负载数的加减并不会影响全局负载数，全局负载数也不会影响路径负载数，可针对不同的情况配置不同的策略
3. 锁配置建议，controller方法上配置的锁请不要有同名，否则容易引发锁被冲掉的问题
4. 事务配置建议，建议在事务的起始方法，也就是第一个含有事务的方法上加锁，避免引发并发下等待事务调度导致的死锁问题，因为业务线程池会被耗尽，或者在设计的时候注意一下，或者加大业务线程数，但是在并发请求数大于业务线程池的线程数时，仍然建议在起始方法加锁
5. 往后的版本更新，说明将在本文档内，起始文档建议看w-io文档，更新文档建议看本文档