Atitit 基于http的接口attilax大总结

[1. 目前基于http的通讯协议rest soap等 1](#_Toc21500)

[1.1. Rest这个目前最流行的了，确实比较好 1](#_Toc17439)

[1.2. Soup 、webservice 1](#_Toc25761)

[2. 协议内容格式(json推荐，xml不推荐了） 1](#_Toc4558)

[3. 参数传递 推荐dsl模式 vs http param模式 1](#_Toc31196)

[4. http接口异常处理 2](#_Toc24543)

# 目前基于http的通讯协议rest soap等

## Rest这个目前最流行的了，确实比较好

## Soup 、webservice

# 协议内容格式(json推荐，xml不推荐了）

至于其他二进制虽然性能好，单不推荐了，除非对方要求高性能

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ）

汉字名：艾提拉（艾龙），   EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 参数传递 推荐dsl模式 vs http param模式

这个不错，但是我提个建议啊，参数传递采用dsl模式。

（http param模式）<http://www.123.com?p1=123&p2=abc>

（dsl模式） [http://www.123.com?d](http://www.123.com?p1=123&p2=abc)sl=new com.attilax.pkg1.class1().method1(123,”abc”)

直接把简化版本的java语句作为dsl传入进去。后边使用反射，实例化 om.attilax.pkg1.class1这个类，然后调用method1(param1,param2)接口即可。。这样的dsl最接近我们平常写的调用方法代码啦。。

这里只使用了一个普通方法表达式dsl，还可以扩展为多条语句，不过就比较复杂了。后台就是拆分多条语句，使用反射操作即可，效果更好哈。

# http接口异常处理

异常处理推荐使用exception异常模式，其次返回码模式。。如果可以俩个都提供最好了，由调用方选择使用哪种模式。。Jdk内部许多即可就是即提供异常模式，也提供返回码模式方便if判断，一般是异常优先，返回码的接口也是调用异常接口，把异常转换为返回码处理。。

异常模式的优点很多了。返回码的优势就是数据量少，对性能在很大通讯量的时候回有所提高。。

可以把后台语言，比如java的异常序列化为json抛出来，然后前台使用js接受，转换为js的异常（即把这个json作为error对象参数throw即可），抛出到上级处理。。前端js也可能分层次，尽可能的吧异常抛出上层，由调用方进行处理即可。。

参考资料

基于http的通信协议――SHCP - hunkcai的专栏 - 博客频道 - CSDN.NET.html

Atitit.异常机制的设计原理 - attilax的专栏 - 博客频道 - CSDN.NET.html

Atititjs javascript异常处理机制与java异常的转换.js exception process - attilax的专栏 - 博客频道 - CSDN.NET.html

atiend