# 数据传输压缩-GZIP研究

# 参考资料

iOS 客户端：

Gzip压缩方法相关参考（提供成扩展文件，可直接使用）：

<https://github.com/nicklockwood/GZIP/tree/master/GZIP>

<https://github.com/levinXiao/LFCGzipUtility>

客户端调用GZIP 方法参考（普通网络请求设置Gzip）：

<http://stackoverflow.com/questions/2682483/nsurlconnection-nsurlrequest-gzip-support>

PHP 服务器端：

参考资料：

<https://www.oschina.net/question/2265205_181108?fromerr=AmqdvDKg>

Apache 开启gzip 参考配置（我本机安装了一个集成环境,直接支持Gzip模块）

<http://jingyan.baidu.com/article/db55b609a7bc234ba20a2f7e.html>

Android 参考资料（我并未实现android 的网络请求）：

<http://blog.csdn.net/kepoon/article/details/7482096>

<http://www.linuxidc.com/Linux/2012-04/58954.htm>

<http://blog.sina.com.cn/s/blog_4adc4b090102vwbi.html>

# 逻辑部分

完成的Demo 实现：

刚开始实现时是使用了 Base64+Gzip 配合使用

简单实现逻辑：

1.1普通字符串参数类型转换为NSData数据类型

1.2 NSData 数据类型进行压缩

1.3 iOS 发送网络请求,常规设置(默认支持Gzip 请求)

1.4 响应data 数据类型转换为普通字符串

1.5.1 服务器返回的响应数据也进行了Gzip压缩（进行区分，和客户端对应）

普通字符串进行Base64 解码-🡪再进行gzip 数据解压

（服务器与客户端逻辑对应（base64过程和Gzip压缩过程））

单独Gzip 压缩实现：（基本同上，把Base64部分去掉即可）

# 代码实现部分

PHP实现代码：

$isGzip = $\_REQUEST['gzipFlag']; //0 表示数据进行了压缩 1表示未压缩

$str = $\_REQUEST['str']; //进行Gzip或未进行压缩的请求参数

/\* //GZIP 和 BASE64 结合使用

if($isGzip=='0'){

//如果进行了压缩,自定义一个压缩的响应数据（客户端进行Base64解码及Gzip解压）

$res = 'responseData:cuihaohaoTest54786';

$res = gzencode($res);

$res = base64\_encode($res);

echo $res;

}else{

//如果请求没有进行压缩,拿到请求的明文,进行Base64，并进行Gzip 压缩

$str3 = base64\_decode($str);

echo gzdecode($str3);

}

\*/

if($isGzip=='0'){

//如果进行了压缩,自定义一个压缩的响应数据（客户端进行Gzip解压）

$res = 'responseData:cuihaohaoTest54786';

$res = gzencode($res);

echo $res;

}else{

//如果请求没有进行压缩,拿到请求的明文直接返回

echo $str;

}

iOS 客户端实现代码：

配合Base64部分：

- (void)testGzipReqWithBase64 {

NSString\* param = @"testGzip1235874";

NSData\* paramData = [param dataUsingEncoding:NSUTF8StringEncoding];

paramData = [paramData gzippedData];

NSString\* base = [[NSString alloc] initWithData:[GTMBase64 encodeData:paramData] encoding:NSUTF8StringEncoding];

BOOL isGzip = true;

NSString\* gzipFlag = @"1"; //0 压缩 1未压缩

if(isGzip){

gzipFlag = @"0";

}else{

gzipFlag = @"1";

}

NSURL \*url = [NSURL URLWithString:[NSString stringWithFormat:@"%@?str=%@&gzipFlag=%@",urlStr,base,gzipFlag]];

NSString\* showMsg = [NSString stringWithFormat:@"请求的地址是：%@ \n 发送给服务器的请求数据是Gzip压缩数据：%@",url,base];

UIAlertView\* alert = [[UIAlertView alloc]initWithTitle:@"提示" message:showMsg delegate:nil cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];

[alert show];

NSLog(@"请求的地址是：%@ 发送给服务器的请求数据是Gzip压缩数据：%@",url,base);

// 3.创建网络请求

NSURLRequest \*request = [NSURLRequest requestWithURL:url cachePolicy:NSURLRequestReloadIgnoringLocalCacheData timeoutInterval:10];

// 创建同步链接

NSURLResponse \*response = nil;

NSError \*error = nil;

NSData \*data = [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];

NSString\* resp = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];

if(isGzip){

NSData\* unData = [GTMBase64 decodeString:resp];

NSData\* resData = [unData gunzippedData];

NSString\* res = [[NSString alloc] initWithData:resData encoding:NSUTF8StringEncoding];

NSString\* resMsg = [NSString stringWithFormat:@"服务器返回的是Gzip压缩并且客户端解压后的响应数据：%@",res];

UIAlertView\* alert2 = [[UIAlertView alloc]initWithTitle:@"提示" message:resMsg delegate:nil cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];

[alert2 show];

NSLog(@"服务器返回的是Gzip压缩并且客户端解压后的响应数据：%@",res);

}else{

NSString\* resMsg = [NSString stringWithFormat:@"服务器返回的是没有Gzip压缩的响应数据(和没有压缩的请求数据一样)：%@",resp];

UIAlertView\* alert2 = [[UIAlertView alloc]initWithTitle:@"提示" message:resMsg delegate:nil cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];

[alert2 show];

NSLog(@"服务器返回未压缩的响应数据(和没有压缩的请求数据一样)：%@",resp);

}

}

单独进行Gzip 代码部分：

-(void)testGziReq{

NSString\* param = @"testGzip1235874";

BOOL isGzip = false;

if(isGzip){

[self requestWithGzip:param];

return;

}else{

[self requestNormalWithParam:param];

return;

}

}

-(void)requestNormalWithParam:(NSString\*)param{

NSURL \*url = [NSURL URLWithString:[NSString stringWithFormat:@"%@?str=%@&gzipFlag=1",urlStr,param]];

NSString\* showMsg = [NSString stringWithFormat:@"请求的地址是：%@ \n 发送给服务器的请求数据是Gzip压缩数据：%@",url,param];

UIAlertView\* alert = [[UIAlertView alloc]initWithTitle:@"提示" message:showMsg delegate:nil cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];

[alert show];

NSLog(@"请求的地址是：%@ 发送给服务器的请求数据是Gzip压缩数据：%@",url,param);

// 3.创建网络请求

NSURLRequest \*request = [NSURLRequest requestWithURL:url cachePolicy:NSURLRequestReloadIgnoringLocalCacheData timeoutInterval:10];

// 创建同步链接

NSURLResponse \*response = nil;

NSError \*error = nil;

NSData \*data = [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];

NSString\* resp = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];

NSString\* resMsg = [NSString stringWithFormat:@"服务器返回的是没有Gzip压缩的响应数据(和没有压缩的请求数据一样)：%@",resp]; UIAlertView\* alert2 = [[UIAlertView alloc]initWithTitle:@"提示" message:resMsg delegate:nil cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];

[alert2 show];

NSLog(@"服务器返回未压缩的响应数据(和没有压缩的请求数据一样)：%@",resp);

}

-(void)requestWithGzip:(NSString\*)param{

param = [NSString stringWithFormat:@"str=%@",param];

NSData\* paramData = [param dataUsingEncoding:NSUTF8StringEncoding];

paramData = [paramData gzippedData];

NSString\* reqURL = [NSString stringWithFormat:@"%@?gzipFlag=0",urlStr];

// NSString\* reqURL = [urlStr stringByAppendingString:@"?gzipFlag=0"];

NSMutableURLRequest \*request = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:reqURL]cachePolicy:NSURLRequestReloadIgnoringLocalAndRemoteCacheData timeoutInterval:60.0f];

NSString \*postLength = [NSString stringWithFormat:@"%lu", [paramData length]];

NSLog(@"压缩后的数据大小是======%@",postLength);

[request setValue:postLength forHTTPHeaderField:@"Content-Length"];

[request setValue:@"application/json" forHTTPHeaderField:@"Accept"];

[request setValue:@"application/json" forHTTPHeaderField:@"Content-Type"];

[request setHTTPMethod:@"POST"];//POST请求

[request setValue:@"gzip" forHTTPHeaderField:@"Accept-Encoding"];

[request setHTTPBody:paramData];//body 数据

NSURLResponse \*response = nil;

NSError \*error = nil;

NSData \*data = [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];

NSData\* resData = [data gunzippedData];

NSString\* res = [[NSString alloc] initWithData:resData encoding:NSUTF8StringEncoding];

NSString\* resMsg = [NSString stringWithFormat:@"服务器返回的是Gzip压缩并且客户端解压后的响应数据：%@",res];

UIAlertView\* alert2 = [[UIAlertView alloc]initWithTitle:@"提示" message:resMsg delegate:nil cancelButtonTitle:@"确定" otherButtonTitles:nil, nil];

[alert2 show];

NSLog(@"服务器返回的是Gzip压缩并且客户端解压后的响应数据：%@",res);

}