

单元测试计划

1452792 施峰 1452789 李尚霖
1452733 孔瑜 1452750 梁英杰

目录

- 1 引言 2
 - 1.1 目的 2
 - 1.2 测试策略 2
 - 1.3 范围 2
 - 1.4 参考文献 2
- 2 测试项目 2
 - 2.1 异步 http 接收模块模块..... 2
 - 2.2 httpRespons 数据处理模块..... 4
 - 2.3http 请求处理模块..... 9
 - 2.4 URL 路径处理模块..... 11
 - 2.4 日志打印模块 12
- 3.测试方法 12
- 4. 测试通过/失败标准 13
- 5.测试挂起/恢复的条件..... 13
- 6.单元测试交付物..... 13
- 7.单元测试任务 14
- 8.环境需求 14
 - 8.1 硬件需求..... 14
 - 8.2 软件需求..... 14
 - 8.3 测试工具..... 15
- 9.角色和职责..... 15
- 10. 单元测试进度 15
- 11.风险和应急计划 16
- 12.审批 17

1 引言

1.1 目的

本文档为安卓上异步收发 http 请求的 Java 库的单元测试提供范围方法资源和进度方面的指导。

本文档的主要读者是测试经理、测试人员和开发人员。

1.2 测试策略

以类为单元，采用独立的单元测试策略，通过设计相应的驱动和桩的方法用来测试类中的方法。在选择类的被测方法时，根据方法的规模和复杂度进行判定。非空非注释代码行数 LOC 大于 20，或复杂度 VG>3 的方法进行单元测试。其他方法不进行单元测试。

对于子类的测试采用独立测试与自下而上的策略，对于类的变化部分设计新的测试用例，与父类相同的部分则重用父类的测试用例。

执行单元测试的次序是根据《软件详细设计说明书》中的软件层次架构图，从图中最小依赖关系的类开始测试，在逐步扩大到依赖关系较强的类，直到所有的类测试完毕。

1.3 范围

单元测试主要包含了计划阶段、设计阶段、实现阶段和执行阶段 4 个阶段。本单元测试计划是整个软件开发项目中的一部分，起始于详细设计阶段，直到单元测试阶段结束后终止。该计划主要处理安卓上异步收发 http 请求的 Java 库单元测试有关的任务安排、资源需求、人力需求、风险管理、进度安排等内容。

1.4 参考文献

《软件需求规格说明书》

《软件详细设计说明书》

2 测试项目

以下为此次对安卓上异步收发 http 请求的 Java 库进行所有函数分析之后，进行筛选出来的 51 个可被测函数以及其他不可测函数，所有的筛选都遵循 1.2 测试策略中的规定，对非空非注释代码行数 LOC 大于 20，或复杂度 VG>3 的方法进行单元测试，其他方法不进行单元测试。

对不满足测试策略中的选取被测试方法将不进行单元测试，但这些方法必须经过严格代码检视，以保证不会出现一些低级性的错误，并且在集成测试阶段统一验证其接口功能的正确性。不被测函数如下表所示。

2.1 异步 http 接收模块模块

设计类标识：AHC (AsyncHttpClient)设计类

测试函数：

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
AHC_FUNC_1	AsyncHttpClient(Scheme Registry schemeRegistry)	74	
AHC_FUNC_2	getDefaultSchemeRegistry(boolean fixNoHttpResponseExce	24	

	ption, int httpPort, int httpsPort)		
AHC_FUNC_3	getUrlWithQueryString(boolean shouldEncodeUrl, String url, RequestParams params)	22	
AHC_FUNC_4	endEntityViaReflection(HttpEntity entity)	24	
AHC_FUNC_5	cancelRequests(final Context context, final boolean mayInterruptIfRunning)	20	
AHC_FUNC_6	sendRequest(DefaultHttpClient client, HttpContext httpContext, HttpRequest uriRequest, String contentType, ResponseHandlerInterface responseHandler, Context context)	42	
AHC_FUNC_7	isInputStreamGZIPCompressed(final PushbackInputStream inputStream)	17	4

不被测函数：

方法标识符	方法名	代码行（LOC）	复杂度（VG）
AHC_FUNC_8	silentCloseInputStream()	8	2
AHC_FUNC_9	silentCloseOutputStream()	8	2
AHC_FUNC_10	cancelAllRequests()	9	2
AHC_FUNC_11	cancelRequestsByTAG()	12	3
AHC_FUNC_12	paramsToEntity()	15	2

设计类标识：**AHR(AsyncHttpRequest)**设计类

测试函数：

方法标识符	方法名	代码行（LOC）	复杂度（VG）
AHR_FUNC_1	run()	36	
AHR_FUNC_2	makeRequest()	24	

AHR_FUNC_3	makeRequestWithRetries()	36	
------------	--------------------------	----	--

不测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
AHR_FUNC_4	cancel()	5	2
AHR_FUNC_5	isCancelled()	6	2

设计类标识: **AHRH(AsyncHttpResponseHandler)**设计类

测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
AHRH_FUNC_1	setUseSynchronousMode(boolean sync)	11	4
AHRH_FUNC_2	handleMessage(Message message)	45	
AHRH_FUNC_3	sendResponseMessage(HttpResponse response)	17	5
AHRH_FUNC_4	getResponseData(HttpEntity entity)	34	

不测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
AHRH_FUNC_5	PostRunnable()	11	3
AHRH_FUNC_6	sendMessage(Message msg)	9	2

2.2 httpRespons 数据处理模块

设计类标识: **B64 (Base64)**设计类

测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
B64_FUNC_1	encode(byte[] input, int offset, int len, int flags)	30	
B64_FUNC_2	process(byte[] input, int offset, int len, boolean finish)	130	

B64_FUNC_3	process(byte[] input, int offset, int len, boolean finish)	102	
------------	--	-----	--

不测试函数：

方法标识符	方法名	代码行（LOC）	复杂度（VG）
B64_FUNC_3	decode()	13	3
B64_FUNC_4	encodeToString()	8	2

设计类标识：**BJHRH(BaseJsonHttpResponseHandler)**设计类

测试函数：

方法标识符	方法名	代 码 行 （LOC）	复杂度（VG）
BJHRH_FUNC_1	onSuccess(final int statusCode, final Header[] headers, final String responseString)	36	
BJHRH_FUNC_2	onFailure (final int statusCode, final Header[] headers, final String responseString, final Throwable throwable)	36	

不测试函数：

方法标识符	方法名	代 码 行 （LOC）	复杂度（VG）
BJHRH_FUNC_3	OnFailure()	3	1
BJHRH_FUNC_4	onSuccess()	3	1

设计类标识：**BHRH(BinaryHttpResponseHandler)**设计类

测试函数：

方法标识符	方法名	代 码 行 （LOC）	复杂度（VG）
BHRH_FUNC_1	sendResponseMessage(HttpResponse response)	25	

不测试函数：

方法标识符	方法名	代 码 行 （LOC）	复杂度（VG）
BHRH_FUNC_2	BinaryHttpResponseHand	8	2

	ler(String[] allowedContentTypes)		
BHRH_FUNC_3	BinaryHttpResponseHand ler(String[] allowedContentTypes, Looper looper)	8	2

设计类标识: **DAHRH(DataAsyncHttpResponseHandler)**设计类

测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
DAHRH_FUNC_1	handleMessage(Message message)	17	4
DAHRH_FUNC_2	getResponseData(HttpEntity entity)	36	
DAHRH_FUNC_3	copyOfRange()	13	4

不测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
DAHRH_FUNC_4	onProgressData(byte[] responseBody)	3	1

设计类标识: **FAHRH(FileAsyncHttpResponseHandler)**设计类

测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
FAHRH_FUNC_1	getTargetFileByParsingURL()	23	
FAHRH_FUNC_2	getResponseData(HttpEntity entity)	24	
FAHRH_FUNC_3	FileAsyncHttpResponseHan dler()	13	4

不测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
-------	-----	----------------	----------

FAHRH_FUNC_4	getTargetFile()	6	2
FAHRH_FUNC_5	getTemporaryFile(Context context)	8	1

设计类标识: **JHRH(JsonHttpResponseHandler)**设计类

测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
JHRH_FUNC_1	onSuccess(final int statusCode, final Header[] headers, final byte[] responseBytes)	48	
JHRH_FUNC_2	onFailure(final int statusCode, final Header[] headers, final byte[] responseBytes, final Throwable throwable)	45	
JHRH_FUNC_3	parseResponse(byte[] responseBody)	26	

不测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
JHRH_FUNC_4	JsonHttpResponseHandler()	6	1

设计类标识: **JSE(JsonStreamerEntity)**设计类

测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
JSE_FUNC_1	escape(String string)	58	
JSE_FUNC_2	writeTo(final OutputStream out)	110	
JSE_FUNC_3	writeToFromStream(OutputStream os, RequestParams.Stream)	26	

	Wrapper entry)		
JSE_FUNC_4	writeToFromFile(OutputStream os, RequestParams.FileWrapper wrapper)	31	

不测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
JSE_FUNC_5	writeMetaData()	11	1
JSE_FUNC_6	JsonStreamerEntity()	6	1

设计类标识：RFAHRH(RangeFileAsyncHttpResponseHandler)设计类

测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
RFAHRH_FUNC_1	sendResponseMessage(HttpResponse response)	24	
RFAHRH_FUNC_2	getResponseData(HttpEntity entity)	24	

不测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
RFAHRH_FUNC_3	updateRequestHeaders(HttpUriRequest uriRequest)	7	3

设计类标识：REH(RetryHandler)设计类

测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
REH_FUNC_1	retryRequest(IOException exception, int)	30	

	executionCount, HttpContext context)		
--	---	--	--

不测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
REH_FUNC_2	isInList(HashSet<Class<?>> list, Throwable error)	7	3

设计类标识：SAHRH(SaxAsyncHttpResponseHandler)设计类

测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
SAHRH_FUNC_1	getResponseData(HttpEntity entity)	33	

不测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
SAHRH_FUNC_2	SaxAsyncHttpResponseHandler(T t)	6	2

2.3http 请求处理模块

设计类标识：RH(RequestHandle)设计类

测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
RH_FUNC_1	cancel(final boolean mayInterruptIfRunning)	13	4

不测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
RP_FUNC_2	shouldBeGarbageCollected()	5	2

RP_FUNC_3	setTag(Object tag)	5	2
-----------	--------------------	---	---

设计类标识: **RP(RequestParams)**设计类

测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
RP_FUNC_1	toString()	35	
RP_FUNC_2	createJsonStreamerEntity(ResponseHandlerInterface progressHandler)	20	
RP_FUNC_3	createMultipartEntity(ResponseHandlerInterface progressHandler)	28	
RP_FUNC_4	getParamsList(String key, Object value)	42	
RP_FUNC_5	add(String key, String value)	13	4

不测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
RP_FUNC_6	RequestParams(Object... keysAndValues)	8	3
RP_FUNC_7	setContentEncoding(final String encoding)	6	2
RP_FUNC_8	put(String key, File files[], String contentType, String customFileName)	10	3
RP_FUNC_9	getParamsList()	6	2

设计类标识: **PCS(PersistentCookieStore)**设计类

测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
PCS_FUNC_1	PersistentCookieStore(Context context)	16	4
PCS_FUNC_2	addCookie(Cookie cookie)	13	4

PCS_FUNC_3	clearExpired(Date date)	11	4
PCS_FUNC_4	byteArrayToHexString(byte[] bytes)	9	4

不测试函数：

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
PCS_FUNC_5	clear()	7	2
PCS_FUNC_6	setTag(Object tag)	5	2
PCS_FUNC_7	deleteCookie(Cookie cookie)	6	1
PCS_FUNC_8	encodeCookie(SerializableCookie cookie)	11	2
PCS_FUNC_9	hexStringToByteArray(String hexString)	6	2

2.4 URL 路径处理模块

设计类标识：MRH(MyRedirectHandler)设计类

测试函数：

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
MRH_FUNC_1	getLocationURI(final HttpResponse response, final HttpContext context)	63	
MRH_FUNC_2	isRedirectRequested (final HttpResponse response, final HttpContext context)	17	5

不测试函数：

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
MRH_FUNC_3	MyRedirectHandler(final boolean allowRedirects)	4	1

设计类标识: **MSSF(MySSLSocketFactory)**设计类

测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
MSSF_FUNC_1	getKeystoreOfCA()	33	

不测试函数:

方法标识符	方法名	代 码 行 (LOC)	复杂度 (VG)
MSSF_FUNC_2	MySSLSocketFactory(K eyStore truststore)	10	1
MSSF_FUNC_3	getKeystore()	9	1
MSSF_FUNC_4	getFixedSocketFactory()	10	1
	getNewHttpClient(KeySt ore keyStore)	13	1

2.4 日志打印模块

设计类标识: **LH(LogHandler)**设计类

测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
LH_FUNC_1	logWithThrowable(int logLevel, String tag, String msg, Throwable t)	26	

不测试函数:

方法标识符	方法名	代码行 (LOC)	复杂度 (VG)
-------	-----	-----------	----------

3.测试方法

根据类规约和操作规约构建测试用例, 合理利用面向对象测试、单元测试等知识, 黑盒测试、白盒测试和静态测试方法对项目中每个方法进行完备测试。

计划的测试执行顺序:

①黑盒测试 (UseCase 法、等价类划分法、错误推测法等)

②静态测试，人工代码走查，依赖于经验和智慧（代码缺陷检查表/网上查阅的依据/PMD 工具的使用）

③白盒测试（基路径覆盖测试）

对具有特殊需求的类辅以下面两种方法设计测试用例：

（1）根据状态转换图构建测试用例。该方法根据被测试的类的对象所处的状态以及状态之间的转移来构造测试用例，对状态之间和状态内部的每一转换及可能发生的异常转换，转换的监护条件等进行全面测试。

（2）基于实现构建测试用例。该方法利用传统逻辑覆盖法、数据流分析法等白盒测试技术对程序的逻辑结构和数据流进行测试，以达到一定的代码覆盖率。

4. 测试通过/失败标准

测试通过的标准如下：

- （1）所有单元测试的用例都被执行并通过；
- （2）所有发现的缺陷都被修正并通过回归测试；
- （3）所有被测对象的前置条件和后置条件组合覆盖率达到 100%或能明确给出不需要达到的理由；
- （4）单元测试报告被授权人批准。

测试失败标准表述如下：

- （1）严重缺陷密度大于 15 个/KLOC；
- （2）发现软件结构有重大设计问题，其修改会导致 20%以上的接口、功能、数量的变化，进一步测试已经无意义；
- （3）发现关键功能未被设计，该功能的设计会导致 20%以上的接口、功能、数量的变化，进一步测试已经无意义；

测试结果审批过程：

- （1）测试被挂起的条件已经被解决。
- （2）需要恢复测试的对象达到单元测试入口条件，在这里要求这些被测对象已经通过代码走读和语法检查。

5.测试挂起/恢复的条件

测试挂起的条件有：

- （1）当某个类在单元测试执行过程中发现有阻塞用例的时候，该类的单元测试被挂起。
- （2）当有 20%以上的被测试类都遇到有阻塞用例的时候，所有类的单元测试被挂起。
- （3）当出现有新增需求的时候，与该需求相关的所有类的单元测试被挂起。
- （4）当开发人员提出要进行设计变更时，相关类的单元测试将被挂起。

测试恢复的条件有：

- （1）测试被挂起的条件已解决。
- （2）需要恢复测试的对象达到单元测试入口条件，在这里要求这些被测试对象已经通过代码走读和语法检查。

6.单元测试交付物

单元测试计划、单元测试设计规格、单元测试用例规格、单元测试用例脚本、单元测试驱动和桩代码、单元测试执行日志、单元测试报告。

7.单元测试任务

单元测试任务表如表:1 所示。

表 1： 单元测试任务表

任务标识	任务描述	责任人	优先级	依赖关系
UT-TASK-001	单元测试计划制定	测试经理	高	
UT-TASK-003	单元测试计划评审	SQA	中	UT-TASK-001
UT-TASK-005	单元测试计划修改	测试经理	中	UT-TASK-003
UT-TASK-007	单元测试设计规格制定	开发或测试人员	中	UT-TASK-003
UT-TASK-009	单元测试设计规格评审	SQA	中	UT-TASK-007
UT-TASK-011	单元测试设计规格修改	开发或测试人员	中	UT-TASK-009
UT-TASK-013	单元测试用例规格设计	开发或测试人员	高	UT-TASK-009
UT-TASK-015	单元测试用例规格评审	SQA	中	UT-TASK-013
UT-TASK-017	单元测试用例规格修改	开发或测试人员	中	UT-TASK-015
UT-TASK-019	单元测试驱动、桩、用例脚本代码实现	开发或测试人员	中	UT-TASK-015
UT-TASK-021	驱动、桩、脚本代码走查	SQA	低	UT-TASK-019
UT-TASK-023	驱动、桩、脚本代码修改	开发或测试人员	低	UT-TASK-021
T-TASK-025	单元测试执行及回归	开发或测试人员	高	UT-TASK-023
UT-TASK-027	单元测试报告	测试经理	高	UT-TASK-025
UT-TASK-029	单元测试报告审批	开发或测试人员	高	UT-TASK-027

8.环境需求

8.1 硬件需求

装载 window 7 或以上的计算机 ， 2G 以上内存

8.2 软件需求

安装 JDK 1.5 以上的 Java 运行环境，以及 Java IDE 如 eclipse 与 intellij，缺陷管理工具如 bugzilla 等

8.3 测试工具

JUnit 4.0

9.角色和职责

单元测试角色和职责如表 2 所示。

表 2：单元测试角色和职责对应表

角色	职责
产品经理	解决资源需求，对单元测试结果进行监督
开发经理	协助制定单元测试计划，安排单元测试任务
测试经理	制定单元测试计划，安排单元测试任务，参与单元测试结果验收
SQA	对单元测试过程进行监控
开发或测试人员	完成单元测试需要的输入，并完成单元测试设计规格，单元测试用例规格、单元测试规程的制定，执行单元测试，记录发现问题，修改问题，并负责问题的回归测试。与此同时，负责定位问题和解决问题

10.单元测试进度

单元测试进度如表 3 所示：

表 3：单元测试进度安排表

任务标识	任务描述	起始日期	周期/天
UT-TASK-001	单元测试计划制定	系统设计结束后 7 天内	
UT-TASK-003	单元测试计划评审	单元测试计划完成后 3 天内	
UT-TASK-005	单元测试计划修改	单元测试计划评审完成后 3 天内	
UT-TASK-007	单元测试设计规格制定	单元测试计划评审完成后 7 天内	
UT-TASK-009	单元测试设计规格评审	单元测试设计规格完成后 3 天内	
UT-TASK-011	单元测试设计规格修改	单元测试设计规格评审完成后 3 天内	

UT-TASK-013	单元测试用例规格设计	单元测试设计规格评审完成后 7 天内	
UT-TASK-015	单元测试用例规格评审	单元测试用例规格完成后 3 天内	
UT-TASK-017	单元测试用例规格修改	单元测试用例规格评审完成后 3 天内	
UT-TASK-019	单元测试驱动、桩、用例脚本代码实现	单元测试用例规格评审完成后 7 天内，并且编码阶段已经开始	
UT-TASK-021	驱动、桩、脚本代码走查	单元测试驱动，桩，用例脚本代码完成后 3 天内	
UT-TASK-023	驱动、桩、脚本代码修改	单元测试驱动，桩，用例脚本代码走读后 3 天内	
T-TASK-025	单元测试执行及回归	单元测试驱动，桩，用例脚本代码走读后 7 天，并且编码阶段已经结束	
UT-TASK-027	单元测试报告	单元测试执行及回归完成后 7 天内	
UT-TASK-029	单元测试报告审批	单元测试报告完成后 3 天内	
	风险预留时间	单元测试阶段工作中任意时候	

11.风险和应急计划

风险和应急计划安排如表 4 所示

表 4： 风险和应急计划

ID	风险描述	责任人	优先级	规避措施	应急计划
1	人员无法及时到位	开发或测试经理	高	在产品的预算中体现这部分需求。定期催促人力资源部进行资源协调。从可能空闲的产品部中物色人员	推迟计划进度。进行招聘。考虑工作外包。
2	人员技能不符合要求	开发或测试经理	中	在人力预算中给出人员技能要求。对提供的人员进行技能面试。从其他部门协调人	提高培训难度，加强培训效果监控，对工作输出加强检视

				员。	
3	测试工具无法到位	开发或测试人员	低	在产品预算中体现这部分需求。 联系其他产品，看是否有空闲的 License。 尽早联系工具代理商，洽谈采购事宜	采用人工分析程序规模。 允许使用规模估计值自行开发测试工具。

12.审批

计划提交人签字： 孔瑜、梁英杰

日期：2017 / 4 / 29

开发经理签字： 施峰

日期：2017 / 4 / 30

产品经理签字： 李尚霖

日期：2017 / 4 / 30