

B/S 系统的接口测试设计研究

侯犀丽

(公安部第三研究所 928实验室, 上海 201210)

摘要: 接口测试是系统组件间接口的一种测试, 是软件开发测试过程中的一个重要环节。主要用于测试系统与外部其他系统之间的接口, 以及系统内部各个子模块之间的接口。测试的重点是要检查接口参数传递的正确性, 接口功能实现的正确性, 输出结果的正确性, 以及对各种异常情况的容错处理的完整性和合理性。一个具备可操作性的接口测试可有效验证软件功能与需求说明、概要设计之间的一致性, 从而确保软件功能的有效性。本文针对基于浏览器Web应用的接口测试进行研究。

关键词: 接口测试; 系统接口; 接口依赖

0 引言

Web接口是Web服务的重要组成部分, W3C将Web服务定义为一个由URI标识的软件系统。随着Web业务日趋复杂, 系统愈加庞大, 版本的发布周期变短, 使得Web 产品测试的难度极大地提高, 工作量也大幅增加^[1]。

1 Web接口

随着微服务技术与前后端分离设计在互联网技术中的应用, 越来越多的系统通过后端接口的方式提供服务, 大大增加了系统的灵活性和快速迭代, 满足了人们对互联网的信息实时更新的需求。Web接口是前端页面与客户端交互的方式, 即浏览器或者其他客户端工具与web服务UI层交互的协议。常见的有两大类, 一是浏览器与服务器交互的 HTTP或HTTPS协议的接口。另一类web service接口如soap, rmi, rpc等。把Web服务分为无状态Web服务和状态Web服务两类, 对两种类型的服务, 都应通过接口测试确保服务的正确性, 一致性与完整性^[2]。

随着越来越多的系统采用前后端分离的设计, 前端开发与后端服务程序同时进行开发, 在增加开发效率的同时, 也加大了质量控制的环节和难度。因此, 针对Web接口的相关功能和接口契约在开发过程中进行多轮测试也是十分必要的。随着技术发展, Web前端的接口测试已经不依赖于服务端代码的完成, 同时, 后端服务代码的测试也有相应的API管理工具供B/S系统前后端开发人员查看, 目的都是为了保证前后端接口调用的一致性^[3]。

2 接口测试的目标

web 接口测试即在web服务程序UI层之上进行测试的一种手段, 是从用户的角度测试web服务程序业务逻辑的正确性。测试的重点是围绕web服务暴露的接口, 检查接口数据的正确性, 这个过程是将web服务程序作为黑盒, 通过接口测试技术提高测试执行效率降低测试的成本。

测试的主要对象是接口, 接口测试主要用于检测外部系统与所测系统之间以及内部各系统之间的交互。在软件开发的设计阶段, 针对外部及内部接口的格式来校验应有约定, 其后的前后端开发均遵循接口设计契约进行。接口测试的重点是检查数据交互、传递和控制管理过程以及系统间的相互依赖关系等。在前期开发阶段, 通过建立接口调用的关系网络, 测试人员可以发现整个系统的关键接口并对其重点测试, 确保关键接口的可用性与稳定性。接口调用关系还可用于接口依赖测试, 其目的是从较大的测试粒度下验证接口的返回, 因此, 接口依赖测试的粒度高于单元测试, 低于系统集成测试^[4]。

接口测试还应当包括接口性能和幂等等测试内容。随着越来越多的大数据应用出现, 单次接口的反应时间也应当尽可能地纳入测试范围, 而不仅仅属于性能测试的范畴。随着后台数据量的增长, 即使接口的请求参数不变, 接口的返回内容也有可能发生较大的变化。因此, 在多次相同的接口请求中, 接口返回是否能随着数据量的增长作出相应的响应也应被全盘考虑。

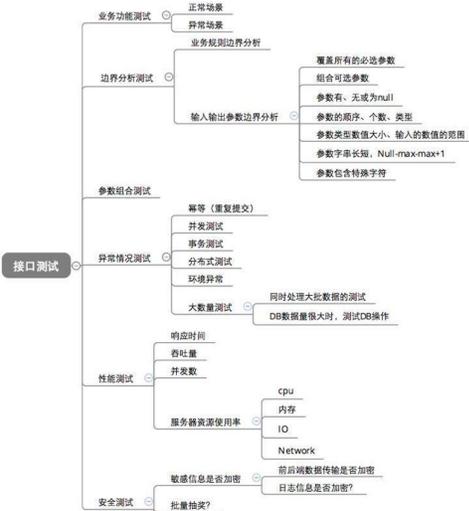


图1 接口测试的组成部分

在实际测试工作中, 除了将接口视为黑盒以外, 如果条件允许, 也有必要对接口可能发生的各种异常情形进行模拟。此时, 需要对接口后端的实现进行模拟返回, 达到测试异常情况的目的。对异常返回的模拟设计接口后端代码的查看, 不能将接口完全视作黑盒 (见图1)。

接口测试还应当包括接口性能和幂等等测试内容, 相同的接口调用在不同重复次数的情况下应当有相同的效果。

随着越来越多的大数据应用出现, 单次接口的反应时间也应当尽可能地纳入测试范围, 而不仅仅属于性能测试的范畴。随着后台数据量的增长, 即使接口的请求参数不变, 接口的返回内容也有可能发生较大的变化。因此, 在多次相同的接口请求中, 接口的响应是否能随着数据量的增长而正确地返回也应当被全盘考虑。

3 接口测试的设计

在整个系统设计中, 由于涉及到多层业务逻辑, 要使各层之间调用透明化, 可以按照接口做具体的事情, 从而融合到整个系统中。在计算机中, 接口是计算机系统两个独立的部件进行信息交换的共享边界。这种交换可以发生在计算机软、硬

件、外部设备或进行操作的人之间,也可以是它们的结合。

从业务逻辑的角度,接口按照不同交互方式划分为不同的种类。电脑等信息机器硬件组件之间的接口叫硬件接口,信息机器软件组件之间的接口叫软件接口。我们的测试对象为软件接口。以常见的REST API为例,针对不同类型的请求设计相应的测试用例。

3.1 GET请求测试的设计

GET请求的本质是发送一个请求来获得服务器上的某个资源,在实际开发中,GET请求的使用方式并没有硬约束,例如可通过GET请求传递的参数更新后台数据,这种做法并不会在软件层面被禁止。因为GET请求在开发过程中的灵活性,测试时应特别注意请求的实际目的。资源通过一组HTTP头和数据体(如HTML文本,或者图片或者视频等)返回给客户端。对GET请求的返回进行以下检查,目前的接口测试工具如PostMan等都支持测试脚本的编写,可以对返回结果做进一步的验证操作并输出报告。

(1) 格式检查:GET请求的返回为某个资源,在Web应用中资源以数据按照一定的格式呈现,因此,返回数据应满足格式要求。常见的格式验证包括是否包含指定的数据项,数据项是否包含指定的子项。若数据项为数组,验证数据项数组是否支持空数组。

(2) 数量检查:在使用GET请求返回数据列表的情景下,还应注意接口是否返回数据总数,应当检验数据总数是否与返回列表的数量一致。在分页查询场景下,请求中可能包含查询页数及每页数量,应对返回结果的数据量进行验证,同时,与后台数据库中的分页查询结果进行比对。但目前分页查询的数据比对更多依赖人工进行SQL查询实现,较难实现自动化,这也是未来进行自动化接口测试时需要改进的方向。

3.2 POST请求测试的设计

POST请求的本质是发送一个请求来获得服务器上的某个资源,待更新的资源一般以数据体的形式被提交至后台进行处理。发送POST请求时,后台数据会根据请求的URL及数据体的组合检查进行相应的操作。因此在测试设计中应注意不同的数据体与URL参数的配合测试。在具体应用场景中,不同的使用场合会对数据质量提出不同的需求,测试设计应该根据场景进行调整。

(1) 登录和注册:登录注册时的POST请求可能包含校验码,可考虑增加校验码的测试。(2) 文件上传:文件上传的测试可以增加不同扩展名文件的测试,验证系统对上传文件的限制。除此以外,文件大小限制,文件名长度限制,图片尺寸和分辨率等也可以作为测试项目。(3) 新建数据:一般地,若后台数据库操作为添加新建数据,前端使用POST请求。在进行新建数据的测试时,对后台数据结构及其依赖关系应有一定的了解,从而设计出符合与不符合后台数据规则的用例,验证系统的可靠性。

3.3 PUT请求测试设计

PUT请求一般用于更新后台指定的资源。HTML表单也不支持这个。本质上来讲,PUT和POST极为相似,都是向服务器发送数据,但它们之间有一个重要区别,PUT通常指定了

资源的存放位置,而POST则没有,POST的数据存放位置由服务器自己决定。客户端向服务器传送的数据取代指定文档的内容。由于资源存放位置的特殊性,在进行PUT请求测试时,测试用例应该覆盖合法与不合法的资源位置,防止错误的请求错误地进行数据覆盖。

3.4 DELETE请求测试设计

DELETE请求删除某一个资源,使用频率较少。服务器根据删除请求里的url所标识的资源进行删除操作。由于数据删除操作十分敏感,测试用例设计时可以特别加入SQL注入安全测试,防止通过DELETE请求进行非法数据操作,造成严重后果。

4 接口集成测试

除了对单个接口分别进行测试之外,将接口的调用关系作为整体进行集成测试也是重要的测试内容。集成测试有助于发现单个接口约定的不合理之处,可以为软件开发的快速迭代提供更加完整且坚实的基础。

接口集成有不同的方法,如果被依赖接口已经完成则通过用Annotation添加注解的方法完成。如果被依赖接口还没有完成,则可以通过Mock的方式完成目标接口测试,进而完成整个业务场景的测试。Mock测试对保证软件开发的前期质量控制有很重要的价值,可以延长软件测试周期,及时发现前期设计中可能存在的问题^[9]。

4.1 Mock测试概念

Mock测试就是在测试过程中,对于某些不容易构造(如HttpServletRequest 必须在Servlet 容器中才能构造出来)或者不容易获取的比较复杂对象(如 JDBC 中的ResultSet 对象),用一个虚拟的对象(Mock对象)来创建以便目标对象测试的测试方法。

4.2 Mock测试工具

通常Mock工具通过简单的方法对于给定的接口生成Mock对象的类库。它提供对接口的模拟,能够通过录制、回放、检查三步来完成大体的测试过程,可以验证方法的调用种类、次数、顺序,可以令Mock对象返回指定的值或抛出指定异常。通过这些Mock工具我们可以方便地构造Mock对象从而使单元测试顺利进行,能够应用于更加复杂的测试场景(见表1)。

表1 Mock工具列表

工具提供者	工具入口网址
GoogleMock	http://code.google.com/p/googlemock/
EasyMock	http://easymock.org/
XXL-API - V2EX	https://github.com/xuxueli/xxl-api
RAP	https://github.com/thx/RAP/wiki/home_cn
NEI	https://nei.netease.com/

4.3 Mock测试的步骤

以EasyMock为例主要通过以下五个步骤完成被依赖接口的输出行为场景对应参数的模拟。

使用EasyMock生成Mock对象;设定Mock对象的预期行为和输出;将Mock对象切换到Replay状态;调用Mock对象方法进行单元测试;对Mock对象的行为进行验证。

(下转第236页)

时发现异常情况并报告给运维技术人员。发射监控系统在不同时间段的检测结果具有不同的意义。计算机技术能够对发射机的运行状态进行模拟,选择最佳发射状态,并完成信息数据汇总与汇报的工作。使用计算机技术也可以实现对发射系统的实时监测,并将运算结果应用于异常情况报警。全天候的实时监测极大地解放了检测人员与运维人员的工作压力,也能够更加准确地对广播电视发射状态进行评估,优化监控系统的运行^[6]。

3.2 自动诊断并处理设备故障

广播电视信号监控是广播电视发射的重要环节。由于信号监控的特殊性,信号监控需要在现场对设备进行监管,完成检测信号等具体任务,使广播电视系统良好运行。依赖于计算机技术的数据存储与处理能力,可以通过计算机分析设备是否正常运行,并记录设备运行中的数据。如果设备出现故障,还能够导出故障设备的历史数据,将故障数据与正常运转的数据进行对比,从而发现设备故障的原因。相关工作人员可以使用存储的历史数据辅助诊断设备的问题所在。更加智能的监控设备甚至能够自动完成对监控设备的诊断,定期的对设备进行检查,及时解决细微问题,避免不可挽回事故的发生。在各项数据正常的情况下删除不必要的信息,提高运转与问题排查的效率,同时避免安全事故的发生,维持广播电视系统的正常运转。

3.3 实现远程监控

广播电视发射监控主要是为了保障发射系统设备状态,降低故障率,报站电视内容的可靠传输。但是传统人工实施监控时,整体工作环境噪声较高,而且机器散热也会使得工作温度升高,长时间记录数据十分容易导致工作人员的疲劳,影响记录的准确性与真实性,导致工作效率降低。人力监控对于设备问题的发现与处理并不能保证完全理性客观,经验不足或者对设备了解不到位会导致应对不当或无法发现问题。而将计算机技术应用于发射监控能够较好地避免该类问题的出现。使用

计算机对信号进行检测或对开关进行控制,能够让工作人员在更舒适的环境对发射设备进行操作。同时,设备运行中如果主信号源发生故障,也可以通过计算机远程操控的方式更换信号源保障节目信号能够稳定传输^[7]。

4 结论

随着广播电视发射技术的不断发展,信息传输的准确性、时效性已经成为维护广播电视正常运转的基本要求。在计算机技术的不断推动下,在广播电视发射监控中融入计算机技术以提高监控准确率与时效性,从而提升整体运行效率是大势所趋。计算机技术在广播电视发射监控中的应用方向包括监测广播电视发射系统工作状态、自动诊断并处理设备故障、实现远程监控等。需要提出的是,计算机技术的应用并不能解决广播电视发射监控中的所有问题,仍然需要相关工作人员熟悉计算机技术在监控系统中的应用方式并熟悉相关操作与原理,提升综合素养,更加科学高效地使用计算机技术完成广播电视发射监控工作,推动广播电视行业的快速发展。

参考文献

- [1] 温布仁. 信息化时代网络技术在广播电视工程技术中的运用与实践[J]. 电子元器件与信息技术, 2021,5(01):49-50.
- [2] 刘鹤. 信息化时代网络技术在广播电视工程技术中的运用[J]. 电子世界, 2021(01):27-28.
- [3] 岳鹏飞. 计算机技术在广播电视发射监控中的运用之研究[J]. 科学技术创新, 2020(19):93-94.
- [4] 倪杰. 计算机自动监控技术在广播电视高山无线发射台中的应用探究[J]. 西部广播电视, 2020(01):230-231.
- [5] 纪少丹. 浅谈计算机技术在广播电视后期制作及广播电视发射监控中的应用[J]. 数字通信世界, 2020(01):202.
- [6] 郑哨军. 计算机自动监控技术在广播电视高山无线发射台中的应用[J]. 西部广播电视, 2019(06):213+215.
- [7] 尹明艳. 视频转码技术在广播电视中的应用研究[J]. 缔客世界, 2020,6(1):43.

(上接第233页)

4.4 接口回归测试

项目需求的瞬息万变引起迭代频率大幅增加,其相对的测试工作量也成倍的增长。为了保证项目迭代的效率和质量,就需要对各种业务场景做回归测试。由于页面改动频繁导致页面自动化测试的维护成本大大增加,然而接口的轻便和接口测试脚本维护成本低。接口自动化测试就应运而生^[5]。由于自动化测试要求测试场景具有稳定性,不会频繁更改,因此常被应用于回归测试。

接口回归测试是在接口没有重大改动的情况下,验证接口协议是否因软件版本迭代而出现变化。在接口测试中,没有功能改动的接口应被纳入回归测试,确保所有接口的可用性。回归测试脚本基本固定,若某个接口发生迭代升级,回归测试脚本也应进行迭代。

5 结论

随着前后端分离设计模式在Web应用中的广泛使用,接口作为数据传递的唯一,对界面起到的作用越来越重要。对接口的全覆盖测试可以确保前后端整合质量。

随着敏捷测试的盛行,我们都知道测试工作要尽早地介

入到项目开发周期中,因为越早的发现 bug,修复的成本就越低。由于功能测试一般都要等到系统提供可测试的 UI 界面后才能进行,单元测试又要求较高的专业性和人力成本,所以选择接口测试来更早的介入测试。接口测试可以在功能界面未开发出来之前对系统的接口进行测试,从而更早的发现并修复问题。

在接口测试的设计中应充分考虑接口的功能特性,着重测试接口的特殊功能需求,如安全性测试,压力测试等等。最后,接口测试的自动化测试较为成熟,相比于UI界面与性能测试,接口测试的功能覆盖面更广,实现难度也较低,是实践自动化测试的较好切入点。

参考文献

- [1] 王会青,冯秀芳. Web 应用软件测试方法的研究[J]. 太原理工大学学报, 2007,38(4):304-306,319.
- [2] 叶波,孙俊若,林洁. 软件接口自动化测试技术研究与实现[J]. 信息化研究, 2010,36(4):47-51.
- [3] 杨硕. 基于 Web 服务组合的企业应用集成研究[D]. 宁夏大学, 2013.
- [4] 李隽. 浅析基于 B/S 模式信息系统数据库优化[J]. 电子元器件与信息技术, 2018(07):86-88.
- [5] 杨清玉,李金丽,陈吉兰,等. HTTP 接口自动化测试方法研究[J]. 微型机与应用, 2016,35(18):22-25.