文章
Nicky Zhu · 一月 11, 2021 阅读大约需 5 分钟

Open Exchange

ObjectScript包管理器中的单元测试和测试覆盖率

本文将描述通过ObjectScript包管理器（见https://openexchange.intersystems.com/package/ObjectScript-Package-Manag...）运行单元测试的过程，包括测试覆盖率测量（见https://openexchange.intersystems.com/package/Test-Coverage-Tool）。

ObjectScript中的单元测试

https://docs.intersystems.com/irislatest/csp/docbook/Doc.View.cls?KEY=TU...

最好的做法是将单元测试代码单独放在源代码树中，无论它只是“ /tests ”还是其他名字。在InterSystems中，我们最终使用/internal/testing/unittests/作为我们事实上的标准，这是有意义的，因为测试是内部/非发布的，而且除了单元测试还有其他类型的测试，但这对于简单的开源项目来说可能有点复杂。您可以在我们的一些GitHub仓库中看到这种结构。

从工作流的角度来看，这在VSCode中非常简单，您只需创建目录并将类放在里面。对于较老的以服务器为中心的源代码控制方法（Studio中使用的方法），您需要正确地地映射这个包，使用的方法会因源代码控制程序而异。

从单元测试类命名的角度来看，我个人的偏好（以及我的团队的最佳实践）是：

UnitTest.<待测试的包/类>[.<待测试的方法/特性>]
例如，如果在类MyApplication.SomeClass 中对方法Foo进行单元测试，单元测试类将被命名为UnitTest.MyApplication.SomeClass.Foo；如果测试是针对整个类的，那么名字就是UnitTest.MyApplication.SomeClass。

ObjectScript 包管理器中的单元测试

让Object
Script包管理器知
道您的单元测试，很简单！只需按
如下所示向module.xml中添加一行代码（来自https://github.com/timleavitt/ObjectScript-Math/blob/master/module.xml - 这是Open Exchange上的@Peter
Steiwer的出色数学扩展包，我以它作为简单的正面例子）：

```
<Module>
  ...
  <UnitTest Name="tests" Package="UnitTest.Math" Phase="test"/>
</Module>
```

这些代码的意思是：

- 单元测试位于模块根目录下的“ tests ”目录中。
- 单元测试在“ UnitTest.Math ”包中。这样很直观，因为被测试的类就在“ Math ”包中。

- 单元测试在包生命周期的“ 测试 ”阶段运行。（当然还有一个可以运行它们的“ 验证 ”阶段，这里不赘述。）

运行单元测试

对于上述定义的单元测试，包管理器提供了一些实用工具来运行它们。您仍然可以像平常使用%UnitTest.Manager那样设置^UnitTestRoot等，但下面的方法可能更简单，尤其在同一个环境下做几个项目的时候。

您可以克隆上述objectscript-math仓库，然后用 zpm "load /path/to/cloned/repo/"加载，或者在您自己的包上使用包名（和测试名）替换“ objectscript-math ”来尝试所有这些方法。

重新加载模块，然后运行所有单元测试：
zpm "objectscript-math test"

只运行单元测试（不重新加载）：
zpm "objectscript-math test -only"

只运行单元测试（不重新加载）并提供详细输出：
zpm "objectscript-math test -only -verbose"

运行一个特定的测试套件（指一个测试目录 -
在本例中，是UnitTest/Math/Utils中的所有测试）而不重新加载，并提供详细输出：
zpm "objectscript-math test -only -verbose -DUnitTest.Suite=UnitTest.Math.Utils"

运行一个特定的测试用例（在本例中，是UnitTest.Math.Utils.TestValidateRange）而不重新加载，并提供详细输出：
zpm "objectscript-math test -only -verbose -DUnitTest.Case=UnitTest.Math.Utils.TestValidateRange"

如果您只是想解决单个测试方法中的小问题：
zpm "objectscript-math test -only -verbose -DUnitTest.Case=UnitTest.Math.Utils.TestValidateRange
-DUnitTest.Method=TestpValueNull"

通过ObjectScript包管理器进行测试覆盖率测量

怎样评估单元测试的质量？测量测试覆盖率虽然并不全面，但至少有参考意义。早在2018年的全球峰会上，我就展示过。 见 - https://youtu.be/nUSeGHwN5pc。

首先需要安装“ 测试覆盖率 ”包：
zpm "install testcoverage"

注意，并不需要ObjectScript包管理器才能安装/运行；可以在Open Exchange上了解更多信息：https://openexchange.intersystems.com/package/Test-Coverage-Tool

不过如果您已经在用ObjectScript包管理器，那么您可以更好地利用这个“ 测试覆盖率 ”工具。

运行测试前，需要指定测试所覆盖的类/routine宏。这一点很重要，因为在非常大的代码库中（例如，HealthShare），测试和收集项目中所有文件的测试覆盖率所需要的内存可能超出您的系统内存。（提一句，如果您感兴趣，可以使用逐行监视器的gmheap。）

文件列表在您的单元测试根目录下的coverage.list文件中；单元测试的不同子目录（套件）可以拥有它们自己的副本，以覆盖在测试套件运行时将跟踪的类/例程。

有关objectscript-math的简单示例，见：https://github.com/timleavitt/ObjectScript-Math/blob/master/tests/UnitTe...；测试覆盖率工具用户指南有更详细的介绍。

要在启用测试覆盖率测量的情况下运行单元测试，只需再向命令添加一个参数，指定应使用TestCoverage.Manager而非%UnitTest.Manager 来运行测试：
zpm "objectscript-math test -only -DUnitTest.ManagerClass=TestCoverage.Manager"

输出（即使是非详细模式）将包括一个URL，供您查看您的类/routine(宏)的哪些行被单元测试覆盖了，以及一些汇总统计信息。

接下来的步骤

这些能不能在CI中自动化？能不能报告单元测试的结果和覆盖率分数/差异？答案是：能！您可以使用Docker，Travis
CI和codecov.io来试一下这个简单示例，见https://github.com/timleavitt/ObjectScript-Math
；我打算以后写篇文章来详细讲讲，介绍几种不同的方法。

#ObjectScript #InterSystems Package Manager (IPM) #持续集成 #InterSystems IRIS #InterSystems IRIS for Health
在 InterSystems Open Exchange 上检查相关应用程序

源URL:
https://cn.community.intersystems.com/post/objectscript%E5%8C%85%E7%AE%A1%E7%90%86%E5%99%A8%E4%B8%AD%E7%9A%84%E5%8D%95%E5%85%83%E6%B5%8B%E8%AF%95%E5%92%8C%E6%B5%8B%E8%AF%95%E8%A6%86%E7%9B%96%E7%8E%87