文章
jieliang liu · 一月 8, 2021 阅读大约需 3 分钟

Open Exchange

qewd-conduit：展示QEWD对Node.js/JavaScript和IRIS的独特集成

关注开发者社区全栈竞赛的朋友会知
道，我提交了一个名为qewd-conduit
的参赛作品。我想总结一下，为什么我认为您应该花点时间来看看这个作品。

qewd-conduit 使用基于 Node.js 的 QEWD 框架和 IRIS，可以为 RealWorld
Conduit 应用程序实现后端 REST API：

https://github.com/gothinkster/realworld

这个方案很酷，它提供了一个平台，让很多人可以为特定应用程序的后端和前端，实现不同的技术解决方案。而qewd-conduit只是符合相同REST
API
后端规范的众多解决方案之一。同样，
要实现完全相同的UI/UX
，您可以尝试使用众多不同前端客
户端中的任何一个，并通过REST与包括QEWD-Conduit在内的任何Conduit后端集成。

这样就可以用各种不同的框架和技术，去执行完全相同的任务，从而对不同的框架和技术进行比较和对比。

RealWorld
应用程序很好地平衡了相关性和非琐碎性（
不仅仅是ToDo
应用程序！），并且也不是太复杂，无论是在UI/UX还是在后端API
方面。因此便于展示和说明如何使用特定技术来实现RealWorld的指定功能。

那么对于qewd-conduit，我将展示如何用QEWD作为Node.js框架来完全用JavaScript
实现一组REST API（在IRIS中维护RealWorld Conduit应用程序的数据）。

QEWD
真正有趣和与众不同的
是，您不仅可以通过一个极高性能的进程内连
接从Node.js/JavaScript 内部访问IRIS，而且，您关注的IRIS
数据库被抽象为持久JavaScript 对象。什么意思呢？可以理解成JavaScript版的
ObjectScript
上箭头，能够无缝、透明地区分内
存中和磁盘上的数组。对于QEWD的抽象（称为QEWD-JSdb
），您实际上不仅拥有普通内存中JavaScript
对象，而且还拥有物理上恰好驻留在IRIS
数据库磁盘上的对象。因此，当您操作这些QEWD-JSdb
JavaScript对象时，实际上是直接在IRIS数据库中访问和操作它们的属性！

您可以观看我今年一月给伦敦Node.js用户组做的QEWD-JSdb的演示视频：

https://www.youtube.com/watch?v=1TlAKTw167s&list=PLam80-FY3vSPW9apMaczTN4dtke9_GYM

我之所以开发qewd-conduit，主要动机之一是想展示针对一款知名且流行的应用程序和用例（即RealWorld Conduit REST API及其相关数据集）的QEWD-JSdb抽象。

然后，使用QEWD，您可以完全用JavaScript来维护和访问 IRIS数据库中的数据：qewd-conduit不包含ObjectScript代码（QEWD堆栈的任何部分也不包含ObjectScript代码），您所需要了解的关于IRIS的所有信息就是如何在JavaScript中使用QEWD-JSdb抽象来建模和操作数据。Node.js和IRIS之间的高速进程内连接，由开源mg-dbx接口（由我的同事 Chris Munt创建和维护）提供：从而在JavaScript和IRIS数据库之间创建了所需的紧密关系，其性能水平接近原生ObjectScript。

https://github.com/robtweed/qewd-conduit/blob/master/QEWD-JSdb.md

如果您有兴趣详细了解我是如何使用QEWD-JSdb对支持RealWorld Conduit应用程序后端的REST API所需的数据进行建模的，我在qewd-conduit仓库中提供了一份文档：

如果您想更深入地了解，可以在这里找到有关 QEWD-JSdb 的背景信息（包括详细教程）：

https://github.com/robtweed/qewd-jsdb

关于mg-dbx接口的更多信息，见：

https://github.com/chrisemunt/mg-dbx

最后一件事：qewd-conduit比RealWorld Conduit规范更进了一步，它还将那些相同的RealWorld Conduit REST API实现为相应的WebSocket消息--WebSocket支持无缝集成到QEWD中，而且，将处理REST API的相同后端逻辑作为WebSocket消息暴露也比较合理。这样您就有了一个难得的机会，让REST和WebSockets执行完全相同的任务，从而比较和对比他们的使用和相对性能。我要感谢Ward De Backer为基于Vue.js的RealWorld 客户端提供了一个支持WebSocket的版本，有关更多信息，请参阅主要的qewd-conduit文档。（剧透提示：您应该会发现WebSocket消息比REST API消息快得多！）

以上就是关于qewd-conduit的一些额外背景知识。正如您期望的那样，这只是一个非常有趣的技术冰山的一角！

那么，如果您觉得这个有趣和有用，请在全栈竞赛中为qewd-conduit投票！

#JavaScript #Node.js #REST API #InterSystems IRIS
在 InterSystems Open Exchange 上检查相关应用程序

源URL:https://cn.community.intersystems.com/post/qewd-conduit%EF%BC%9A%E5%B1%95%E7%A4%BAqewd%E5%AF%B9nodejsjavascript%E5%92%8Ciris%E7%9A%84%E7%8B%AC%E7%89%B9%E9%9B%86%E6%88%90