文章
Hao Ma · 十一月 2, 2021 阅读大约需 7 分钟

IRIS 2021 技术文档 First Look 33 - InterSystems IRIS Native API for Node.js

本文档解释了如何使用 Native API 从 Node.js 应用程序访问 InterSystems IRIS® globals。Native API 还允许您运行 ObjectScript 方法、函数和例程。在本文中，您将首先连接到 InterSystems IRIS。然后您将在 InterSystems IRIS 中设置和检索一个 global 节点的值，并在另一个 global 节点上进行迭代。您还将调用 InterSystems IRIS 类方法。所有这些活动都将在 Node.js 应用程序中执行。
为了让您体验 Native API，而又不陷入细节的困境，本次探索特意设计得很简单。这些活动被设计成只使用默认设置和功能，这样您就可以熟悉功能的基本原理，而不必处理那些离题或过于复杂的细节。当您把 IRIS Native 引入您的生产系统时，您可能需要做一些不同的事情。本文档末尾提供的参考资料将使您对在生产中使 Native API 的情况有一个很好的了解。
要浏览所有的技术概要（First Look），包括可以在 InterSystems IRIS 免费的评估实例上执行的那些，请参见 InterSystems First Looks（《InterSystems 技术概要》）。

1 InterSystems IRIS 存储结构简介
InterSystems IRIS 提供了一种易于使用的方法来将数据存储在持久化多维数组中。global 是存储在物理 InterSystems IRIS 数据库中的命名多维数组。在应用程序中，globals 到物理数据库的映射基于当前命名空间，命名空间提供一个或多个物理数据库的逻辑统一视图。例如，要使用一个名为^Settings 的 global 将值 "Red"与键 "Color"关联起来，请使用 InterSystems IRIS Basics：Connecting an IDE（《InterSystems IRIS 基础: 连接一个 IDE》）中的为您的实例描述的程序打开 InterSystems 终端（InterSystems Terminal），并输入以下代码：

```
set ^Settings("Color")="Red"
```

您可以利用 globals 的多维特性来定义一个更复杂的结构：

```
set ^Settings("Auto1","Properties","Color") = "Red" set ^Settings("Auto1","Properties","Model") = "SUV" set ^Settings("Auto2","Owner") = "Mo"
set ^Settings("Auto2","Properties","Color") = "Green"
```

有关 globals 的更多信息，请参见 Using Globals（《使用 globals》）。

2 探索 Native API for Node.js
此时，您已准备好试验 Native API 了。这个简短的演示将向您演示如何在一个简单的 Node.js 应用程序中使用 Native API。
想试试 InterSystems Native API for Node.js 的在线视频演示吗？请查看 Node.js QuickStart（Node.js 快速入门）！

探索 Native API for Node.js

2.1 用前须知
要使用该程序，您需要一个安装了您最喜欢的 Node.js IDE 的系统来运行，并需要一个运行中的 InterSystems IRIS 实例来连接。您对 InterSystems IRIS 的选择包括多种类型的已授权的和免费的评估实例；该实例不需要由您正在工作的系统托管（尽管它们必须相互具有网络访问权限）。有关如何部署每种类型的实例的信息（如果您还没有可使用的实例），请参见 InterSystems IRIS Basics: Connecting an IDE（《InterSystems IRIS 基础：连接一个 IDE》）中的 Deploying InterSystems IRIS（部署 InterSystems IRIS）。使用同一文档中的 InterSystems IRIS Connection Information（InterSystems IRIS 连接信息）和 Node.js IDE 中的信息，将 IDE 连接到您的 InterSystems IRIS 实例。

2.2 设置您的项目
接下来，您需要使用 npm 来设置您的项目，npm 是和 Node.js 一起发布的 JavaScript 包管理器。作为这个过程的一部分，您将安装 Native API 模块 intersystems-iris-native。
使用这个程序：

1. 在命令提示符下，为演示创建一个名为 IRISNative 的新文件夹，并切换到该目录。 例如：

mkdir IRISNative cd IRISNative

2. 使用 npm 包管理器初始化新项目：

npm init

3. 按照提示，创建 package.json 文件。
4. 要安装 IRIS Native API 模块：
- 如果您的 InterSystems IRIS 实例安装在您正在工作的系统上，请输入命令

npm install --save <install-dir>\dev\nodejs\intersystems-iris-native

其中 <install-dir> 是安装 InterSystems IRIS 的目录， 例如 C:\InterSystems\myIRIS。
- 如果您的实例没有安装在您正在工作的系统上，请按照以下步骤操作：

a. 下载或克隆以下 repo 到您的 IDE 中：
https://github.com/intersystems/quickstarts-nodejs
b. 如果您已经下载了 repo，请解压。
c. 切换到您刚刚克隆或下载的 repo 的 Solutions 目录，例如：

cd quickstarts-nodejs/Solutions

d. 要安装 IRIS Native API 模块，请输入：

npm install --save intersystems-iris-native

2.3 Native API 应用程序
现在您已经创建了项目，接下来您将创建一个小的应用程序，演示 Native API 的一些功能。
1. 在您的 IDE 中，在 IRISNative 目录中创建一个新的源文件，将该文件保存为 IRISNative.js。

探索 Native API for Node.js

2. 将以下代码粘贴到 IRISNative.js 中， 将 InterSystems IRIS 实例的连接信息 替换为 connectionInfo 中的值。您可以指定所示的 USER 命名空间，也可以选择在实例上创建的另一个命名空间：

```
const irisnative = require('intersystems-iris-native')
// Modify connection info based on environment let connectionInfo = {
host: '127.0.0.1',
port: 1972, ns: 'USER',
user: 'SYSTEM', pwd: 'SYS'
};
// create database connection
const connection = irisnative.createConnection(connectionInfo);
//create IRIS instance
const dbnative = connection.createIris();
console.log('[1] Setting and getting a global');
// setting and getting a global
// ObjectScript equivalent: set ^testglobal("1") = 8888 dbnative.set(8888, 'testglobal', '1');
// ObjectScript equivalent: set globalValue = $get(^testglobal("1")) let globalValue = dbnative.get('testglobal','1');
console.log('The value of testglobal is ' + globalValue); console.log();
console.log('[2] Iterating over a global');
// modify global to iterate over dbnative.set(7777, 'testglobal', '1');
dbnative.set(8888, 'testglobal', '2');
dbnative.set(9999, 'testglobal', '3');
let subscriptIter = dbnative.iterator('testglobal'); console.log('walk forwards');
for ([key,value] of subscriptIter) { console.log('subscript='+ key +', value=' + value);
};
console.log();
```

```
console.log('Iterate backwards a different way');
let revIter = dbnative.iterator('testglobal').reversed(); let node = revIter.next();
while (!node.done) {
console.log('subscript='+ node.value[0] +', value='+ node.value[1]); node = revIter.next();
}
console.log();
console.log('[3] Calling a class method');
// calling a class method
// ObjectScript equivalent: set returnValue = ##class(%Library.Utility).Date(5) let returnValue =
dbnative.classMethodValue("%Library.Utility", "Date", 5); console.log(returnValue);
// close connection connection.close();
```

示例代码分为三个部分：
1. 第一部分展示了如何设置一个 global 的值以及稍后如何检索它。这部分里面执行的命令等同于 ObjectScript 的 SET 和 GET 命令。
2. 第二部分展示了如何迭代 global 的子节点，类似于 $ORDER ObjectScript 函数。
3. 第三部分展示了如何使用 Native API 从您的 Node.js 应用程序调用 ObjectScript 类方法。

注意： ObjectScript中的 Globals 以插入符号（^）开头。在使用 Native API 的 Node.js 应用程序中，这不是必需的。

了解有关 IRIS Native 的更多信息

2.4 运行练习
现在您已经准备好运行演示应用程序了。如果示例执行成功，您应该会看到带有示例代码结果的打印输出：
[1] Setting and getting a global The value of testglobal is 8888
[2] Iterating over a global walk forwards
subscript=1, value=7777 subscript=2, value=8888 subscript=3, value=9999
Iterate backwards a different way subscript=3, value=9999 subscript=2, value=8888 subscript=1, value=7777

[3] Calling a class method Apr 11, 2019

2.5 确认管理门户（Management Portal）中的变更
接下来，使用以下程序在管理门户（Management Portal）中确认您的结果：
1. 使用 InterSystems IRIS Basics:Connecting an IDE（《InterSystems IRIS 基础：连接一个 IDE》）中为您的实例描述的 URL，在浏览器中打开您的实例的管理门户（Management Portal）。
2. 如果您不在代码指定的命名空间中，请切换到该命名空间。
3. 导航到 Globals 页面(System Explorer（系统资源管理器） > Globals)。您应该会看到示例代码中创建的 testglobal global。点击 View（查看）来查看其内容。 您应该会看到 global 的两个节点：^testglobal(1) = 8888 和 ^testglobal(2) = 9999。

3 了解有关 IRIS Native 的更多信息
有关 IRIS Native、globals 和 InterSystems IRIS 的更多信息，请参见：
- Using the Native API for Node.js（《使用 Native API for Node.js》）
- Using Globals（《使用 Globals》）

#InterSystems IRIS for Health

源 URL:
https://cn.community.intersystems.com/post/iris-2021-%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3-first-look-33-intersystems-iris-native-api-nodejs