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 13 - InterSystems IRIS Native API for Java

本文档解释了如何使用 InterSystems IRIS Native 功能从 Java 应用程序中访问 InterSystems IRIS®数据平台的 globals。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 Java 应用程序中执行。
为了让您体验 IRIS Native，而又不陷入细节困境，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。请确保不要把这种对 IRIS Native 的探索与真实的情况相混淆! 本文档末尾提供的参考资料将使您对在生产中使用 IRIS Native 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 Globals 简介
Globals 提供了一种易于使用的方式，来存储持久化多维数组中的数据。Global 是存储在 InterSystems IRIS 物理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procedure described for your instance（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 为什么 IRIS Native 很重要？
IRIS Native 是建立在 InterSystems IRIS JDBC 功能之上的一项功能，它允许您执行核心类 ObjectScript 命令的有限子集，并使用 globals 访问 InterSystems IRIS 数据，类似于在 InterSystems 终端（InterSystems Terminal）中使用的方式。该功能利用 JDBC 连接（JDBC connection）在 Java 应用程序中暴露（expose）核心 ObjectScript 功能。重要的是，由于 IRIS Native 使用与 JDBC 相同的连接，因此 InterSystems IRIS 数据通过 JDBC 作为关系表暴露（expose）给 Java 应用程序，通过 IRIS Native 作为 globals 暴露（expose）给 Java 应用程序。

探索 IRIS Native

InterSystems IRIS
提供了一组独特的功能，可以使用相同的物理连接和事务环境来使用多个范例（本机、关系和面向对象）操作数据。

3 探索 IRIS Native
以下向您简短演示如何在 Java 应用程序中使用 IRIS Native。(想试试 InterSystems IRIS Java 开发和互操作性功能的在线视频演示吗？请看 Java QuickStart（Java快速入门）！)

3.1 用前须知
要使用该程序，您需要在一个系统上工作，安装 JDK 1.8 版本和您选择的 Java IDE，并连接一个正在运行的 InterSystems IRIS 实例。您对 InterSystems IRIS 的选择包括多种已授权的和免费的评估实例；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）。关于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 Java IDE 中的信息，将 IDE 连接到您的 InterSystems IRIS 实例。
您还需要将 InterSystems IRIS JDBC 驱动程序 intersystems-jdbc-3.0.0.jar 添加到您的本地 CLASSPATH。您可以从https://github.com/intersystems/quickstarts-java/tree/master/lib 下载这个文件。如果您已经在本地系统或您能访问的另一个系统上安装了 InterSystems IRIS，您可以在 install-dir\dev\java\lib\JDK18 中找到该文件，其中 install-dir 是 InterSystems IRIS 的安装目录。
3.2 连接 InterSystems IRIS
InterSystems IRIS 连接字符串的语法是：
jdbc:IRIS://hostIP:superserverPort/namespace,username,password
其中的变量表示在 InterSystems IRIS Basics:Connecting an IDE（《InterSystems IRIS 基础：连接一个 IDE》）中connection settings described for your instance（为您的实例描述的连接设置）——这与您将 IDE 连接到实例时使用的信息相同。将命名空间设置为预定义的命名空间 USER，如下面的代码所示，或者设置为您在安装的实例中创建的另一个命名空间（只要您更新代码）。
如果您连接到本地 Windows 系统上的实例（使用主机名 localhost 或 IP 地址 127.0.0.1），该连接会自动使用一种特殊的、高性能的本地连接，称为shared memory connection（共享内存连接），它为 IRIS Native 提供了更好的性能。有关更多信息，请参见 First Look:JDBC and InterSystems IRIS（技术概要: JDBC 和 InterSystems IRIS）。
3.3 使用 IRIS Native
现在，您已经准备好试验 IRIS Native 了。在您已连接的 IDE 中（请参见“ 用前须知 ”），创建一个名为 IRISNative 的新 Java 项目，并粘贴以下代码。确保编辑变量superserverPort（超级服务端口）、username（用户名）、name space（命名空间）和password（密码），以反映您实例的正确值。

```
import java.sql.DriverManager;
import com.intersystems.jdbc.IRISConnection; import com.intersystems.jdbc.IRIS;
import com.intersystems.jdbc.IRISIterator;

public class IRISNative {
protected static int superserverPort = 00000; // YOUR PORT HERE protected static String namespace = "USER";

探索 IRIS Native

protected static String username = "SYSTEM"; protected static String password = "SYS";
public static void main(String[] args) { try {
// open connection to InterSystems IRIS instance using connection string IRISConnection conn = (IRISConnection)
DriverManager.getConnection
("jdbc:IRIS://localhost:"+superserverPort+"/"+namespace,username,password);
// create IRIS Native object
IRIS iris = IRIS.createIRIS(conn);

System.out.println("[1. Setting and getting a global]");
// setting and getting a global
// ObjectScript equivalent: set ^testglobal("1") = 8888 iris.set(8888,"^testglobal","1");
// ObjectScript equivalent: set globalValue = $get(^testglobal("1")) Integer globalValue =
iris.getInteger("^testglobal","1");
System.out.println("The value of ^testglobal(1) is " + globalValue); System.out.println();

System.out.println("[2. Iterating over a global]");
// modify global to iterate over
// ObjectScript equivalent: set ^testglobal("1") = 8888
// ObjectScript equivalent: set ^testglobal("2") = 9999 iris.set(8888,"^testglobal","1");
iris.set(9999,"^testglobal","2");
// iterate over all nodes forwards
IRISIterator subscriptIter = iris.getIRISIterator("^testglobal"); System.out.println("walk forwards");
while (subscriptIter.hasNext()) {
String subscript = subscriptIter.next(); System.out.println("subscript="+subscript+",
value="+subscriptIter.getValue());
}
```

```
System.out.println();
System.out.println("[3. Calling a class method]");
// calling a class method
// ObjectScript equivalent: set returnValue = ##class(%Library.Utility).Date(5) String returnValue =
iris.classMethodString("%Library.Utility","Date",5); System.out.println(returnValue);
System.out.println();
// close connection and IRIS object iris.close();
conn.close();
} catch (Exception ex) { System.out.println(ex.getMessage());
}
}
}
```

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 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（subnode），类似于 $ORDER ObjectScript 函数。
3. 第三部分展示了如何使用 IRIS Native 从您的 Java 应用程序调用 ObjectScript 类方法（class method）。

如果示例执行成功，您应该会看到带有示例代码结果的输出：

了解有关 IRIS Native 的更多信息

```
[1.Setting and getting a global] The value of ^testglobal(1) is 8888
[2.Iterating over a global] walk forwards
subscript=1, value=8888 subscript=2, value=9999
[3.Calling a class method] Jan 30, 2018
```

3.4 确认管理门户（Management Portal）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URL described for your instance（为您的实例描述的 URL），在浏览器中打开您实例的管理门户（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)。您应该会看到在示例代码中创建的 testglobal。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。

4 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：
First Look:JDBC and InterSystems IRIS（《技术概要：JDBC 和 InterSystems IRIS》）
Using Globals（《使用 Globals》）
Using the Native API for Java（《使用 Native API for Java》）

#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-13-intersystems-iris-native-api-java