

文章
Jeff Liu · 九月 22, 2021  阅读大约需分钟

IRIS 2021 拨文档 First Look 14-- JDBC 和 InterSystems 数据库

本文档介绍了如何使用 InterSystems JDBC 驱动程序连接到 InterSystems IRIS® 数据平台实例，这样您就可以在 InterSystems IRIS 中使用 Java。

要浏览所有的拨 (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 拨)。

JDBC：如何在 InterSystems IRIS 中使用它

InterSystems 提供了完全兼容的 (JDBC 4.2)、纯 Java、type 4 JDBC 驱动程序，它是一个独立的 JAR 文件，没有任何依赖性。如果您已经熟悉 JDBC，并且安装了 JDK 1.8，那么您需做的就是将 JDBC 驱动程序添加到本地 CLASSPATH 中 (请参见 [JDBC: Exploring It \[\(JDBC: 探索它\)\]](#))。JDBC URL (连接字符串) 是：

```
jdbc:IRIS://ipAddress:superserverPort/namespace
```

其中的变量表示 InterSystems IRIS 实例主机的 IP 地址、实例的超级服务器端口和实例上的命名空间。

如果您连接到本地计算机上的一个实例 (使用主机名 localhost 或 IP 地址 127.0.0.1)，该连接可以使用一种特殊的、高速本地连接，称为 [共享内存连接 \(shared memory connection\)](#)。有关共享内存连接 (shared memory connection) 的更多信息，请参见 "[JDBC: What's Unique about Shared Memory Connections \[\(JDBC: 共享内存连接独特之处\)\]](#)"。

本文档的重点是让您了解在 InterSystems IRIS 中使用 JDBC，而不是让您陷入细节困境，所以我们做了简单的探索。但是，当您把 InterSystems IRIS 引入您的生产系统时，您需做很多不同的事情，例如 (但不限于) 安全性面。所以请确保要把这种对 InterSystems IRIS 的探索与真实情况相混淆！本文档末尾提供的参考资料将使您对在生产中使用 JDBC 与 InterSystems IRIS 的情况有一个很好的了解。

JDBC：InterSystems IRIS Java 连接选项的一部分

InterSystems IRIS JDBC 驱动程序是 InterSystems IRIS 的核心 Java 组件，支持传统的关系 (SQL) 访问。它还使用 InterSystems IRIS Native API for Java 的 Java 调用提供连接机制，这些调用可以访问本机存储格式中的数据。对于基于对象的 Java 集成 InterSystems IRIS 还提供一个单独功能-----InterSystems IRIS XEP 组件 (InterSystems IRIS XEP component)。

综上所述，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 [native]、关系 [relational] 和面向对象 [object-oriented]) 操作数据。对于更复杂的应用程序，InterSystems 完全支持 Hibernate。支持所有这些连接形式的-----InterSystems IRIS XEP、Hibernate 以及 InterSystems IRIS Spark 连接器 (Connector)-----是 InterSystems IRIS JDBC 驱动程序。

拨：JDBC 和 InterSystems 数据库 1

JDBC: 探索它

JDBC: 共享内存连接 (Shared Memory Connections) 的独特之处

与其他数据库平台一样, 到远程 InterSystems IRIS 实例的 JDBC 连接是通过 TCP/IP 进行的。为了最大限度地提高性能, InterSystems IRIS 还提供了 Java 共享内存连接 (shared memory connection)。与 InterSystems IRIS 实例在同一计算机上运行的许多 Java 应用程序都可以使用共享内存连接 (shared memory connection)。

共享内存连接 (shared memory connection) 是一个临时设备, 支持虚拟内存, 由 JDBC 客户端和运行在同一物理计算机上的 InterSystems IRIS 实例共享。此外, 这些连接不会对内核网络堆栈进行潜在的昂贵调用。通过使用直接从 JDBC 客户端到 InterSystems IRIS 的通道, 它们最终为 JDBC 提供了低延迟和高吞吐量。

有关共享内存 (shared memory connection) 的详细信息, 请参见 [Using Java with the InterSystems JDBC Driver](#) (在 InterSystems JDBC 驱动程序中使用 Java) 中的 "Shared Memory Connections (共享内存连接)"。

JDBC: 探索它

我们开发了一个演示, 向您展示如何使用 JDBC 和 InterSystems IRIS-----以及这有多么简单。

请注意, 这段代码并没有演示 [InterSystems Java 共享内存连接](#) (shared memory connection) 的性能提升, 因为它无法处理共享内存连接 (shared memory connection) 可以高效处理的大量数据。

想试试 InterSystems IRIS Java 开发和互操作性的在线视频演示吗? 请查看 [Java QuickStart \(Java 快速入门\)](#)!

用前须知

要使用该程序, 您需要在系统上工作, 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 的选择包括各种已授权的和免费的评估实例; 实例不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 教程: 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。对于这个演示, 您可以连接到 `USERnamespace`, 如下面的代码所示, 或者您可以指定在已安装的实例中创建的另一个命名空间。

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。如果您已经在您的本地计算机或您能访问的另一台计算机上安装了 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。

尝试示例代码

将示例代码剪切并粘贴到您的 IDE 中, 使用 [InterSystems IRIS Basics: Connecting an IDE](#) (InterSystems IRIS 教程: 连接一个 IDE) 中的 connection settings described for your instance (为您的实例描述的连接设置) 更新 `url` 和 `connection` 变量以及用户名和密码。

```
import java.sql.*;
```

```

public class JDBCSample {
?
public static void main(String[] str) throws Exception {

String url = "jdbc:IRIS://127.0.0.1:1972/USER";
?
Class.forName("com.intersystems.jdbc.IRISDriver");
?
Connection connection = DriverManager.getConnection(url,"_SYSTEM","SYS");

// Replace _SYSTEM and SYS with a username and password on your system
?
String createTable = "CREATE TABLE People(ID int, FirstName varchar(255), LastN
ame varchar(255))"; String insert1 = "INSERT INTO People VALUES (1, 'John', 'Sm
ith')";
?
String insert2 = "INSERT INTO People VALUES (2, 'Jane', 'Doe')"; String query =
"SELECT * FROM People";
?
Statement statement = connection.createStatement(); statement.executeUpdate(cre
ateTable); statement.executeUpdate(insert1); statement.executeUpdate(insert2);
?
ResultSet resultSet = statement.executeQuery(query); System.out.println("Printi
ng out contents of SELECT query: "); while (resultSet.next()) {
?
System.out.println(resultSet.getString(1) + ", " + resultSet.getString(2) + ",
" + resultSet.getString(3));
?
}
?
resultSet.close(); statement.close(); connection.close();
?
}
?
}
}

```

如果连接和查询已经成功完成，您应该会看到一个控制台窗口(console window)，其中包含 SELECT 查询的结果。

4 了解有关 JDBC 的更多信息

要了解有关 JDBC、InterSystems IRIS 中的其他 Java 互操作性和其他相关主题的更多信息，请参见：

- Using Java JDBC with InterSystems IRIS (在 InterSystems IRIS 中使用 Java JDBC) 中的 "InterSystems Java Connectivity Options (InterSystems Java 连接选项)" ----- 对 JDBC 驱动程序支持的所有 InterSystems IRIS Java 接口的概述。
- Using Java with the InterSystems JDBC Driver (在 InterSystems JDBC 驱动程序中使用 Java) ----- InterSystems 文档: step-by-step instructions for using JDBC (使用 JDBC 的详细说明)。
- First Look: XEP Object Persistence with InterSystems IRIS (摘要: 使用 InterSystems IRIS 持久化 XEP 对象) ----- InterSystems 文档: Java XEP First Look (Java XEP 摘要)
- First Look: InterSystems IRIS Native API for Java (摘要: InterSystems IRIS Native API for

- Java)-----InterSystems 文档: InterSystems IRIS Native API First Look((InterSystems IRIS Native API 教程))
- [Java Overview\(Java 概述\)](#)-----InterSystems 在线学习: 介绍视频
- Persisting Java Objects with InterSystems XEP((使用 InterSystems XEP 持久化 Java 对象))-----InterSystems 文档: step-by-step instructions for using XEP((使用 XEP 的详细说明))
- InterSystems Implementation Reference for Java Third Party APIs ((Java 第三方 API 的 InterSystems 实施参考))-----InterSystems 文档: connecting to InterSystems IRIS using JDBC, Hibernate, and Spark((使用 JDBC、Hibernate 和 Spark 连接 InterSystems IRIS))。
- Using the InterSystems Spark Connector((使用 InterSystems Spark 连接器))-----InterSystems 文档: using InterSystems IRIS as an Apache data source((使用 InterSystems IRIS 作为 Apache 数据源))
- [Hibernate and JDBC compared\(\(Hibernate 和 JDBC 比较\)\)](#)-----栈溢出(Stack Overflow)文章

[#文档](#)

源 URL: <https://cn.community.intersystems.com/post/iris-2021-%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3-first-look-14-jdbc-%E5%92%8C-intersystems-%E6%95%B0%E6%8D%AE%E5%BA%93>