Connecting Dots
For Your IT Career

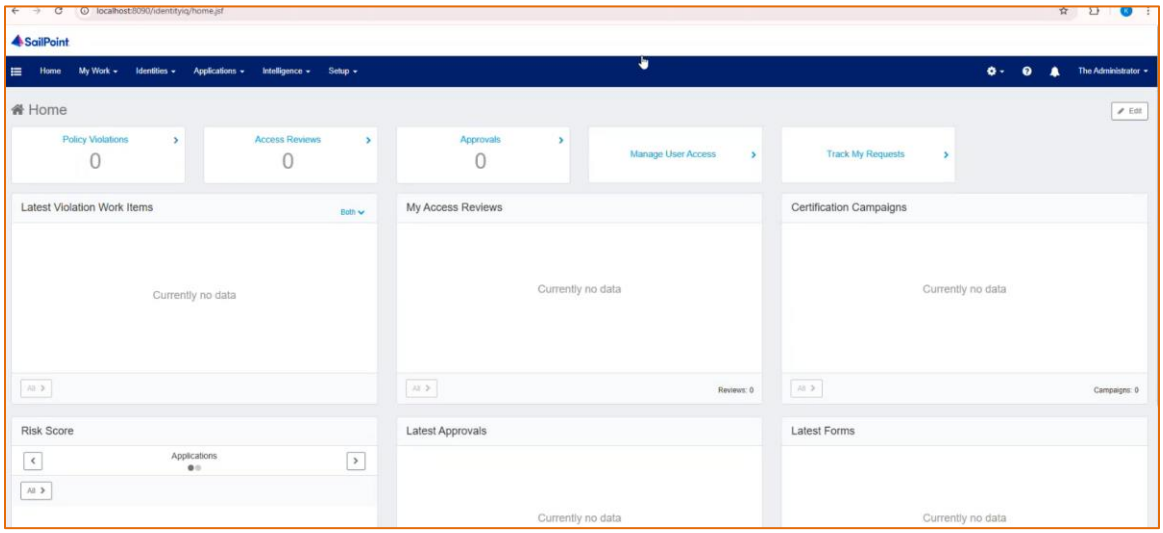
Knowledge Article: Hands-On with SailPoint IIQ (Java Intro)

This document summarizes key learnings and steps from the SailPoint IIQ Java Intro session. It includes practical takeaways

Introduction to SailPoint & IAM

Identity and Access Management (IAM) ensures employees have the right access to applications. SailPoint IIQ is a leading IAM tool, built on Java, and rules form the backbone of its customization.

Step 1 –SailPoint UI



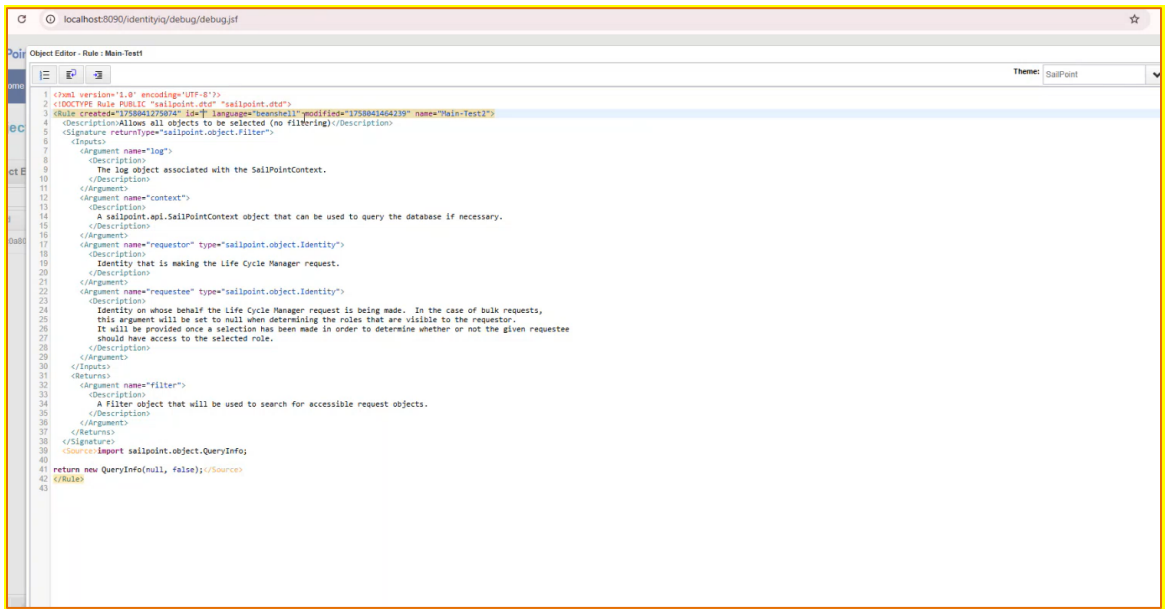
Step 2 – Accessing Debug Page

Navigate to `identityiq/debug`. This debug console acts as a lightweight IDE where developers can write, test, and execute Beanshell/Java rules directly within SailPoint.

Connecting Dots
For Your IT Career

Step 3 – Creating Rules

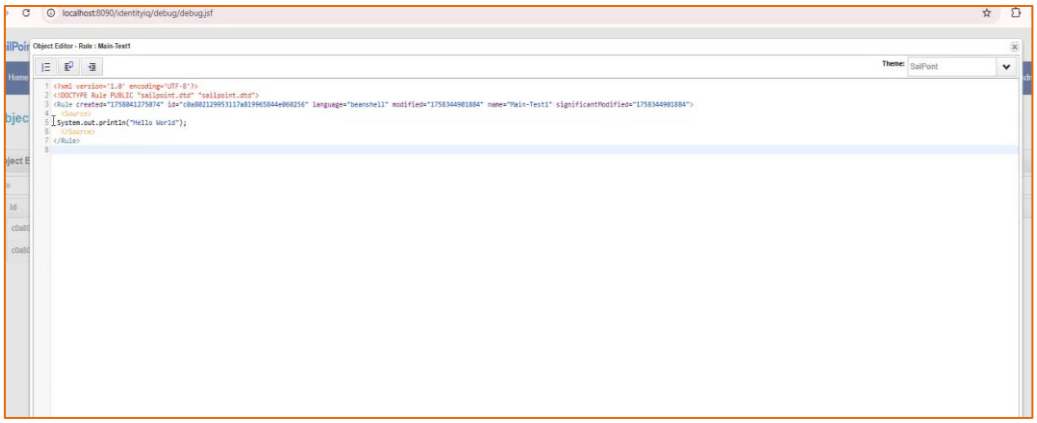
Best practice: duplicate an existing rule, rename it, and remove the ID. SailPoint automatically generates a new ID.



Step 4 – Writing Your First Rule

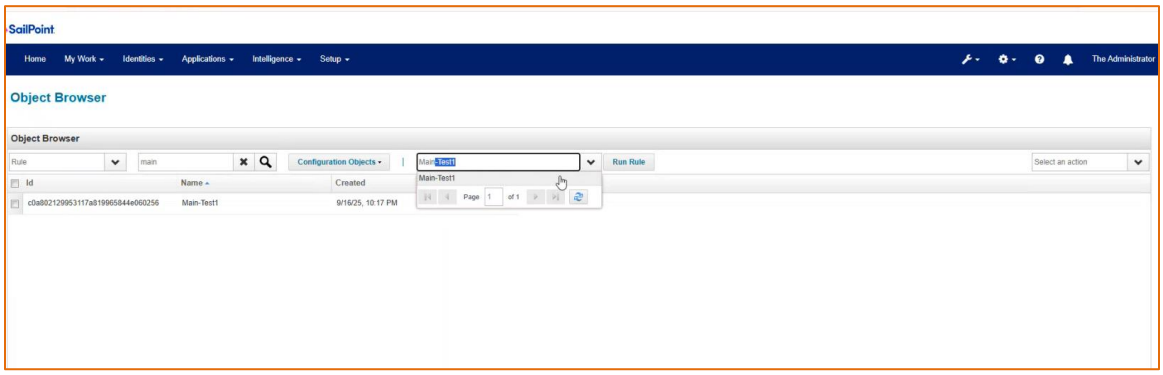
Example logic:

System.out.println("Hello World")



Step 5 - Search for Your Rule

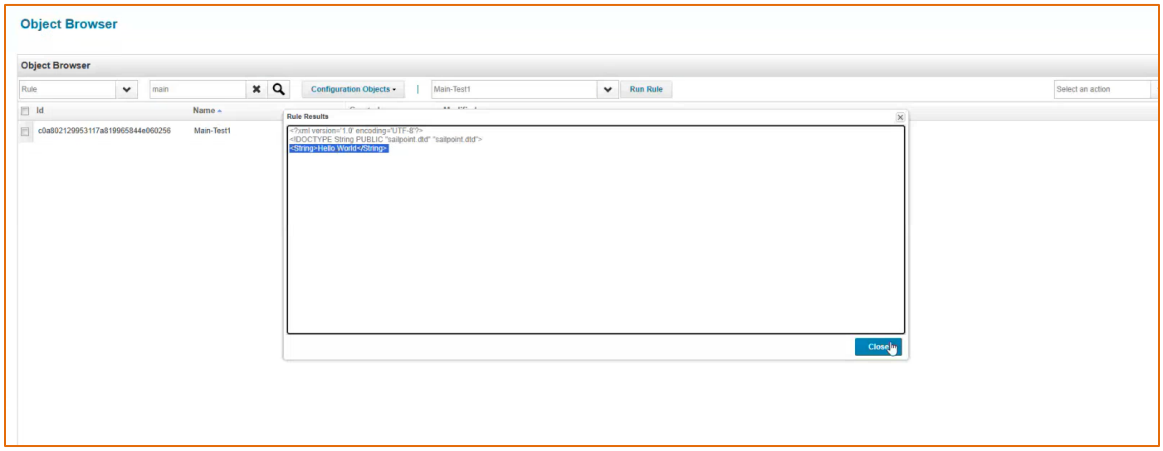
In the Object Browser, select Rule as the object type. Search by the name of your rule (e.g., `main-test1`) and open it to verify the logic.



Step 6 – Quick Output with Return

To display results instantly in the Debug console:

```
return "Hello World";
```



Key Programming Basics

The session also covered Java/Beanshell fundamentals relevant for IIQ development:

- Data Types – int, string, boolean
- String operations – concatenation (useful for building display names)
- Importance of return statements
- Difference between system logs vs console outputs

Connecting Dots
For Your IT Career

Key Takeaways

- Rules are the core of SailPoint IIQ development.
- Use Debug page to test and prototype safely.
- Duplicate rules instead of creating from scratch.
- System.out.println → server logs, return → console output.
- Java/Beanshell basics are essential for SailPoint customizations.