[New Guide] Download the 2018 Guide to Java: Features, Improvements, & Updates
Download Guide

Debugging Wildfly Swarm Applications in Intellij

by Matthew Casperson MVB · Apr. 25, 16 · Integration Zone · Tutorial

Heads up...this article is old!

Technology moves quickly and this article was published **2 years ago**. Some or all of its contents may be outdated.

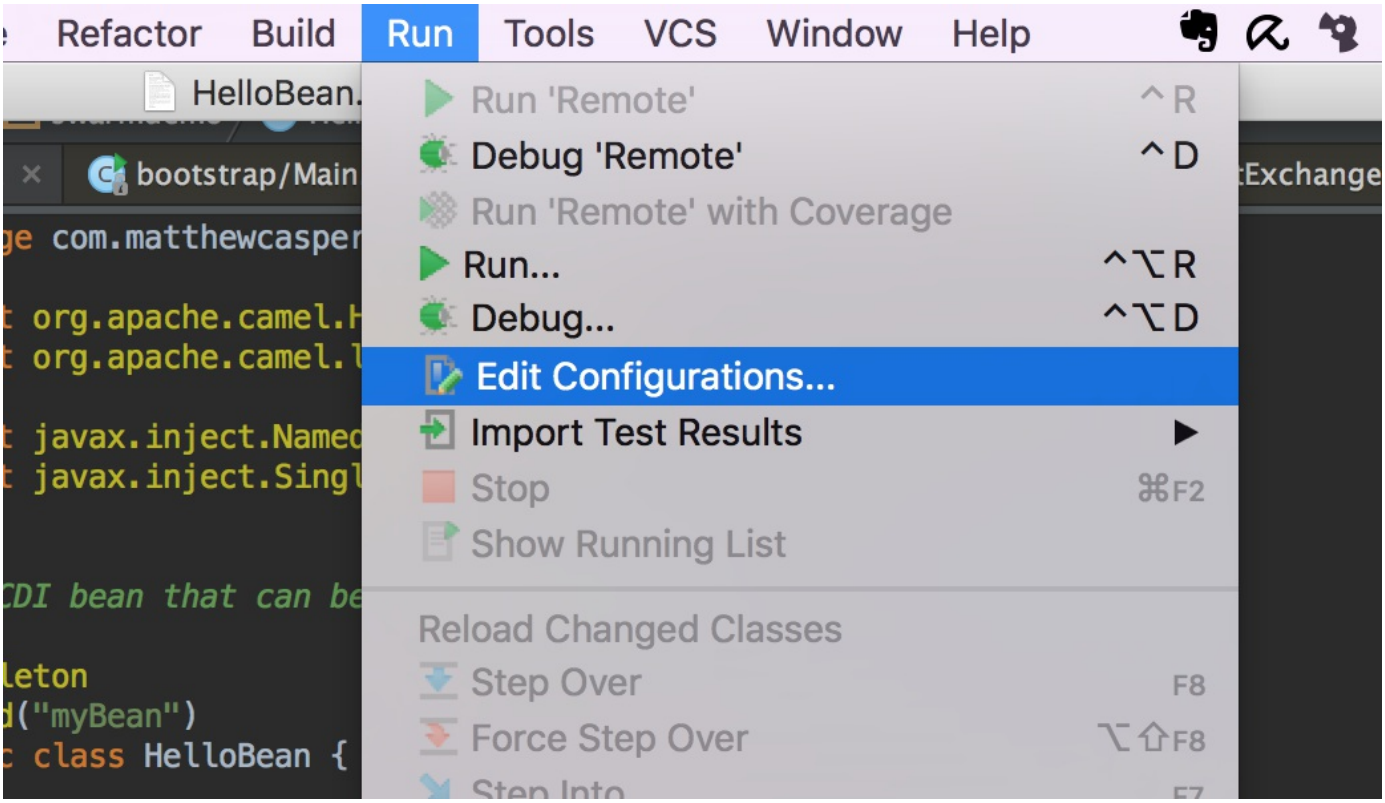
Like (2) Comment (3) Save Tweet

Ready for feedback: How would you use Capital One's Virtual Card Numbers API?

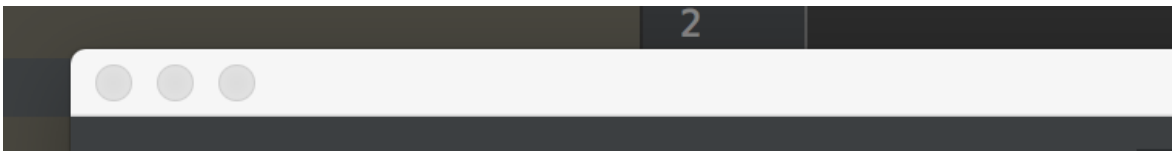
Once you start developing apps in WildFly Swarm, you'll soon find yourself wanting to step through your code to debug issues that may arise. But how do you do that given that the environment that runs your code isn't available until you build it?

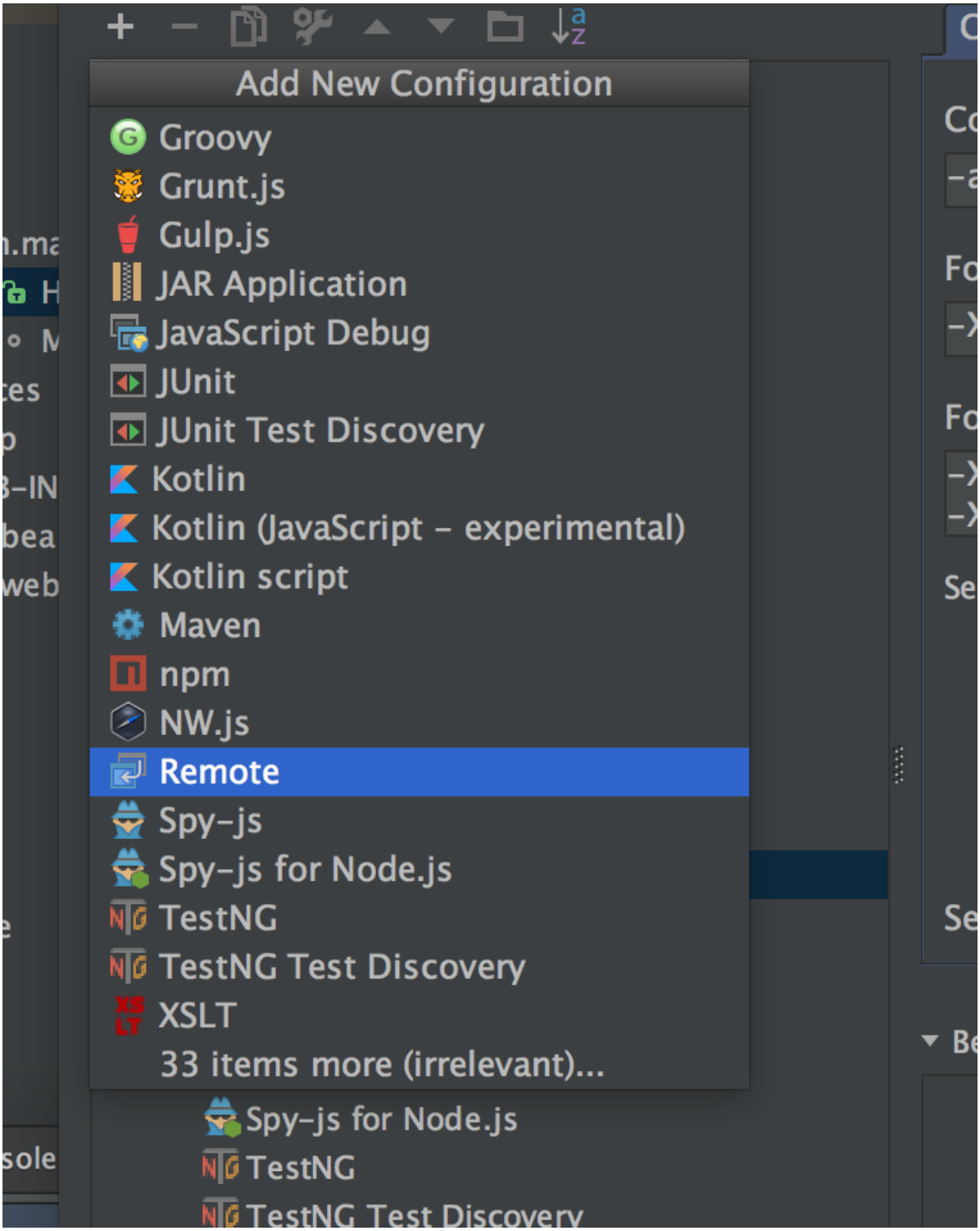
Typically you debug WildFly applications by configuring a WildFly instance that is run by IntelliJ with your applications inside it. To debug a WildFly Swarm application, you need to use IntelliJ's ability to debug remote applications.

Click **Run -> Edit Configurations...**



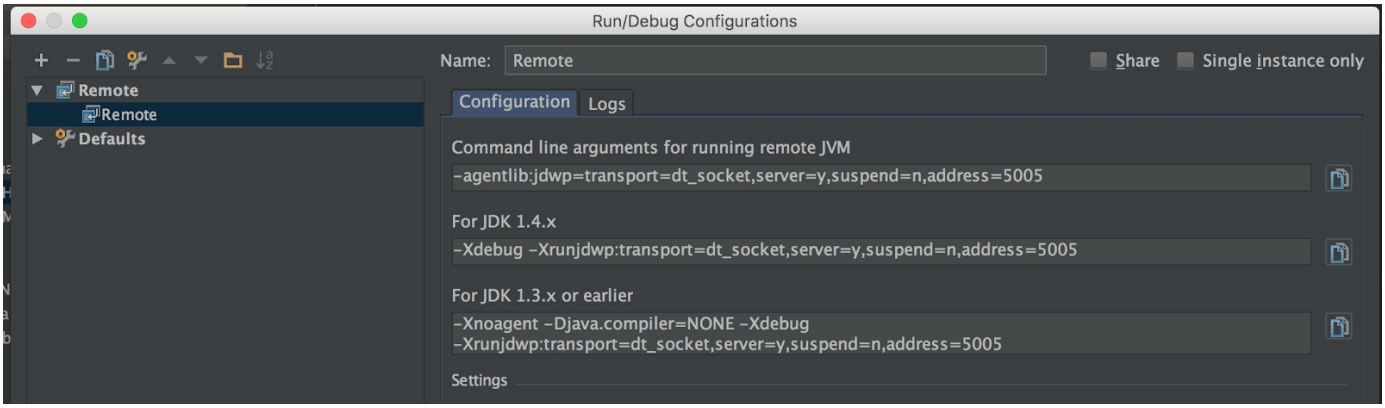
Then click the plus icon in the top left hand corner of the dialog, and select the **Remote** option.

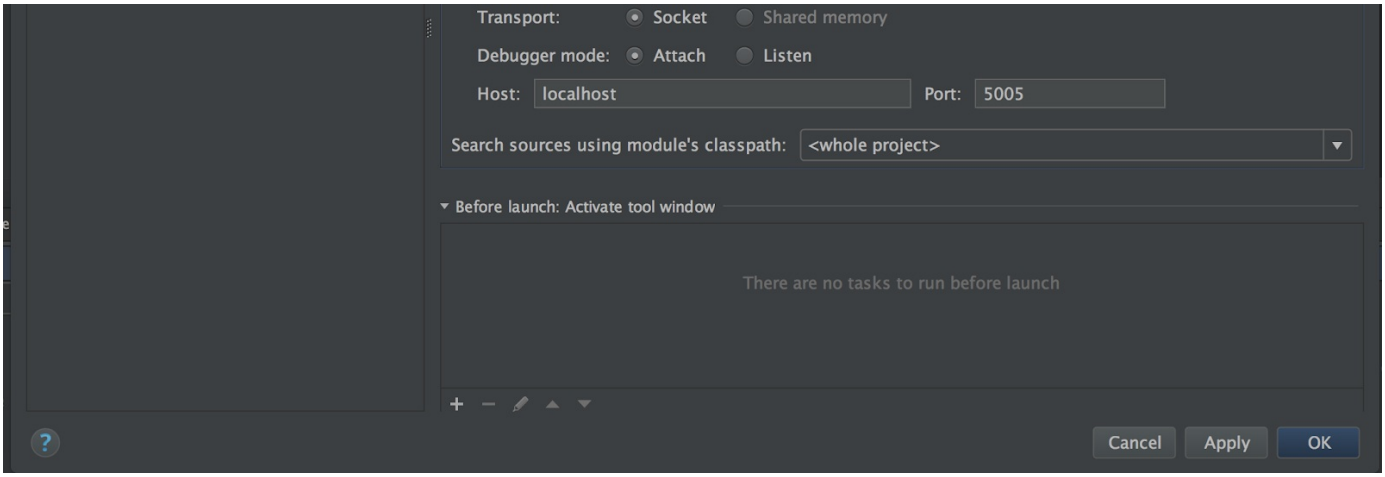




You can accept all the defaults here. All you need to do is give your configuration a name. But before you close the dialog, grab a copy of the **For JDK 1.4.x** string. Mine was:

```
-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005
```



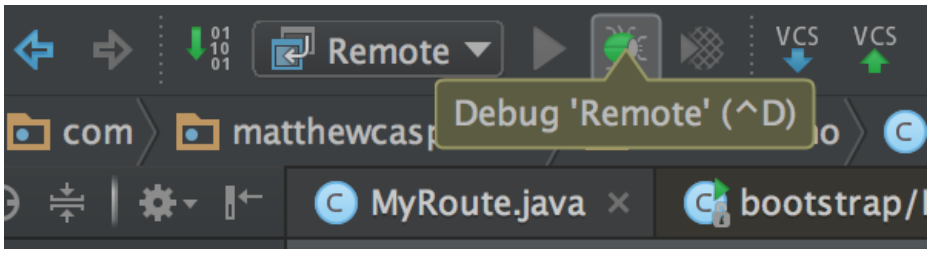


Click **OK** to close the dialog and save your changes.

Now you can run your Swarm application with the options supplied by IntelliJ. I run the sample project from this article with this command:

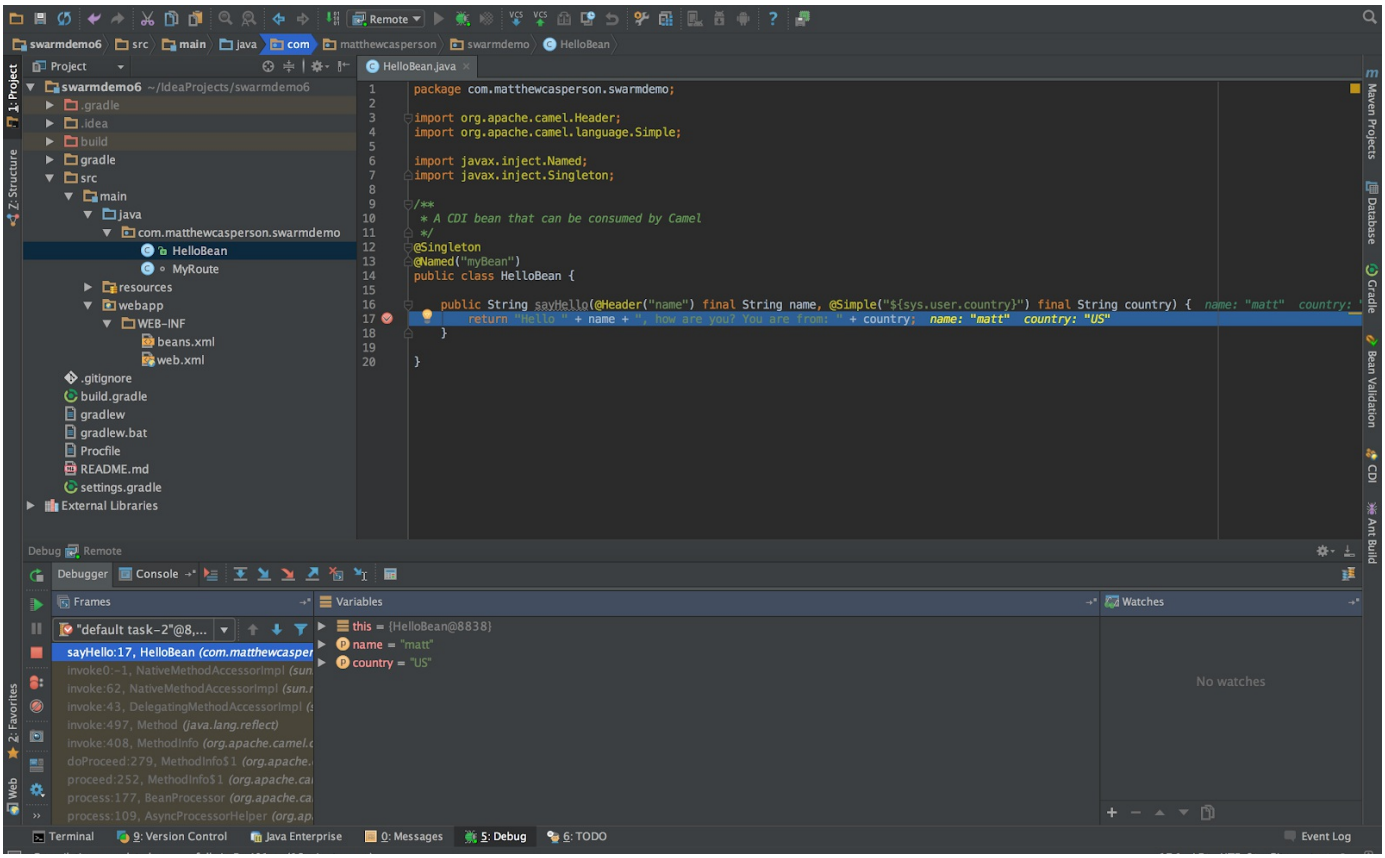
```
java -jar -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005 build/libs/swarmdemo6-swarm.jar
```

Once you application is running, you can debug the remote configuration you just created.



This will connect to the Swarm application you just started, and step through the code like any other application you would debug with IntelliJ.

Note in the screenshot below I have set a breakpoint in my CDI bean, and can inspect the variables that were passed in.



Learn how Capital One's Virtual Card Numbers can benefit digital merchants and consumers.

Like This Article? Read More From DZone

Microservices With Camel and WildFly Swarm - Introduction

A Simple Camel Route in WildFly Swarm

Security Considerations with Camel HTTP Services

Free DZone Refcard
RESTful API Lifecycle Management

Topics: WILDFLY , WILDFLY SWARM , CAMEL

Like (2) Comment (3) Save Tweet

Published at DZone with permission of Matthew Casperson , DZone MVB. See the original article here.
Opinions expressed by DZone contributors are their own.

Integration Partner Resources

The Future of Enterprise Integration: Data Virtualization
Cloud Elements

How event streaming can benefit API design.
CapitalOne

Modernizing Application Architectures with Microservices and APIs
CA Technologies

Analyst Report: The Death of Traditional Data Integration
SnapLogic