

Sessions Tutorial



About Terracotta Documentation

This documentation is about Terracotta DSO, an advanced distributed-computing technology aimed at meeting special clustering requirements.

Terracotta products without the overhead and complexity of DSO meet the needs of almost all use cases and clustering requirements. To learn how to migrate from Terracotta DSO to standard Terracotta products, see [Migrating From Terracotta DSO](#). To find documentation on non-DSO (standard) Terracotta products, see [Terracotta Documentation](#). Terracotta release information, such as release notes and platform compatibility, is found in [Product Information](#).

- **Introduction**
- [How DSO Clustering Works](#)
- [Platform Concepts](#)
- [Hello Clustered World](#)
- **Setup and Configuration**
- [Planning for a Clustered App](#)
- [Configuring Terracotta DSO](#)
- [Configuration Reference](#)
- [Installation](#)
- **APIs**
- [Using Annotations](#)
- [Cluster Events](#)
- [Data Locality Methods](#)
- [Distributed Cache](#)
- [Clustered Async Data Processing](#)
- **Tool Guides**
- [Developer Console](#)
- [Operations Center](#)
- [tim-get \(TIM Management Tool\)](#)
- [Platform Statistics Recorder](#)
- [Eclipse Plugin](#)
- [Sessions Configurator](#)
- [Clustering Spring Webapp with Sessions Configurator](#)
- [Maven](#)
- [JMX](#)
- **Testing, Tuning, and Deployment**
- [Top 5 Tuning Tips](#)
- [Testing a Clustered App](#)
- [Tuning a Clustered App](#)
- [Deployment Guide](#)
- [Operations Guide](#)
- **FAQs and Troubleshooting**
- [General FAQ](#)
- [DSO Technical FAQ](#)
- [Troubleshooting Guide](#)
- [Gotchas](#)
- [Non-portable Classes](#)
- **Reference**
- [Migrating From DSO](#)
- [Concept and Architecture Guide](#)
- [Examinator Reference Application](#)
- [Clustered Data Structures Guide](#)
- [Integrating Terracotta DSO](#)
- [Clustering Spring Framework](#)
- [Integration Modules Manual](#)
- [AspectWerkz Pattern Language](#)
- [Glossary](#)

Release: 3.6

Publish Date: November, 2011 [Documentation Archive](#) »

Clustering a Spring Web Application

- [Purpose](#)
- [What You'll Need](#)
- [What You'll See](#)
- [Overview](#)
- [Getting Started](#)
- [Build the Spring JPetStore Sample](#)
- [Start the JPetStore Database](#)
- [Start the Terracotta Sessions Configurator](#)
- [Import the JPetStore WAR](#)
- [Start the Application Servers Without Terracotta](#)
- [View the Application Unclustered](#)
- [Start the Application Servers With Terracotta](#)
- [Inspect the Clustered Session Object Graph](#)
- [Conclusions](#)
- [Resources](#)

Purpose

This tutorial shows how to configure a web application for session clustering using Terracotta Sessions. By the end of the tutorial, you should be ready to use Terracotta Sessions on your own Web applications.

What You'll Need

1. Terracotta Software
2. [The Spring Framework](#) - We'll be using the *Spring JPetStore* example application.
3. [Apache Ant](#)

What You'll See

At the end of the tutorial, you will have multiple application servers running a web application with clustered sessions.

Overview

The basic steps of this tutorial are:

1. [Download and install Terracotta.](#)
2. [Download Spring and build the Spring JPetStore sample.](#)
3. [Start the JPetStore database.](#)
4. [Start the Terracotta Sessions Configurator](#)
5. [Import the JPetStore WAR into the Terracotta Sessions Configurator.](#)
6. [Start the application servers without Terracotta enabled.](#) Verify that the application works without *Terracotta* and that its sessions are not clustered.
7. [Restart the application with Terracotta enabled. Verify that session clustering works.](#)
8. [Inspect the clustered sessions object graph.](#) *Terracotta* enables runtime visibility into live clustered objects.

Getting Started

1. Download and install .
2. Download and unzip [The Spring Framework](#). We'll be using the *JPetStore* sample application that comes with the *Spring Framework* kit.
3. Download and install [Apache Ant](#) if you don't have it installed already. Detailed instructions for installing *Ant* are in the [Apache Ant Manual](#)

Build the Spring JPetStore Sample

Note: For more detailed instructions on building the *JPetStore* sample, see the *readme.txt* file in the *jpetstore* samples directory.

1. Change directories to `<spring_dir>/samples/jpetstore`
2. Run `ant warfile`

```
%> cd <spring_dir>/samples/jpetstore
%> ant warfile
```

This will leave the *JPetStore* WAR file in `<spring_dir>/samples/jpetstore/dist/jpetstore.war`

Start the JPetStore Database

1. Change directories to `<spring_dir>/samples/jpetstore/db/hsqldb/`
2. Run the `server.sh` script.

```
%> cd <spring_dir>/samples/jpetstore/db/hsqldb/ %> sh server.sh
```

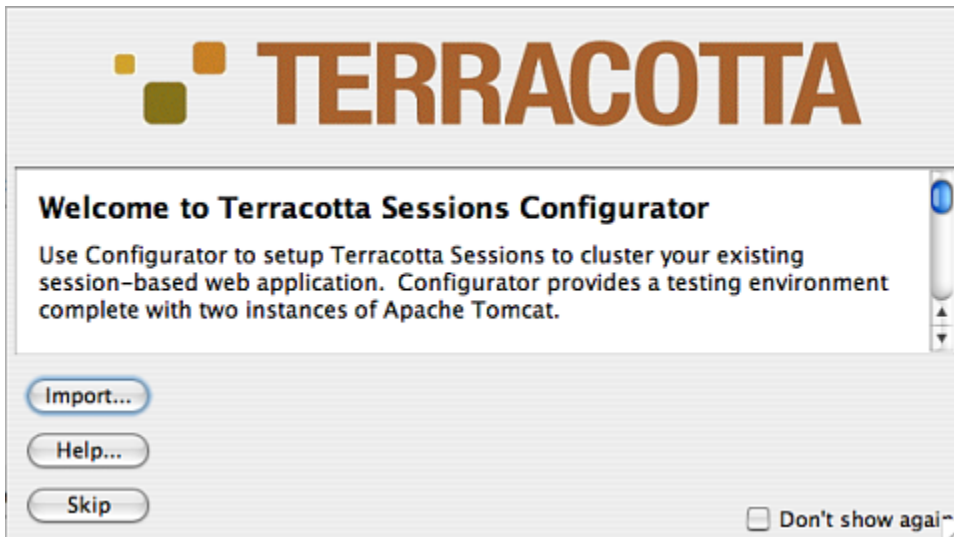
Start the Terracotta Sessions Configurator

You can [obtain the Sessions Configurator from the Terracotta forge](#). See the [README.txt](#) for information on how to run the Sessions Configurator. See the [Sessions Configurator Guide](#) for an introduction to the Sessions Configurator.

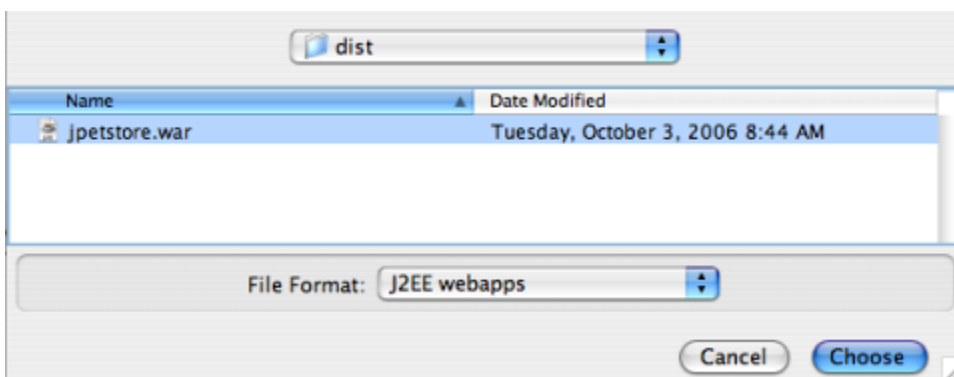
Import the JPetStore WAR

Note: If the welcome screen is not visible, you can use the `File -> Import...` menu.

Import the *JPetStore* application by clicking the `Import...` button.



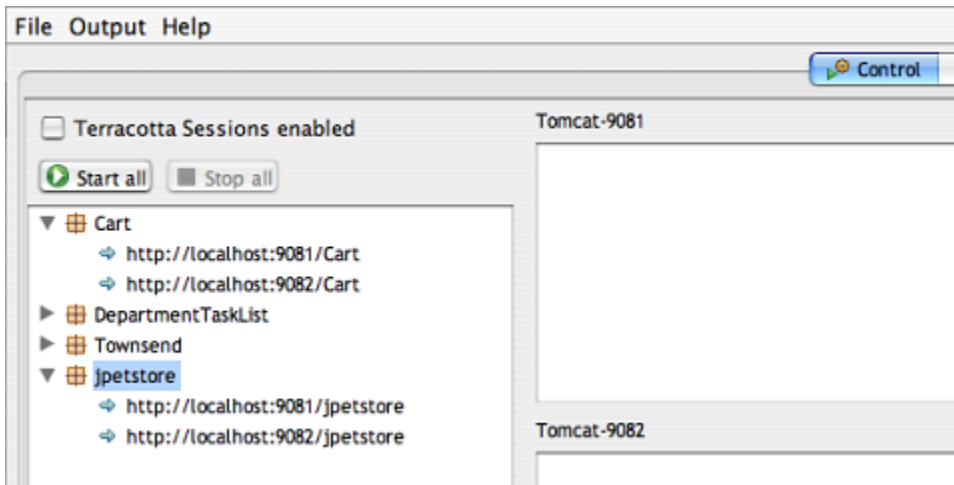
Use the file chooser to select `<spring_dir>/samples/jpetstore/dist/jpetstore.war`



Start the Application Servers Without Terracotta

Ensure that the *Terracotta Sessions enabled* checkbox is *NOT* selected. Click the *Start all* button to start the application servers.

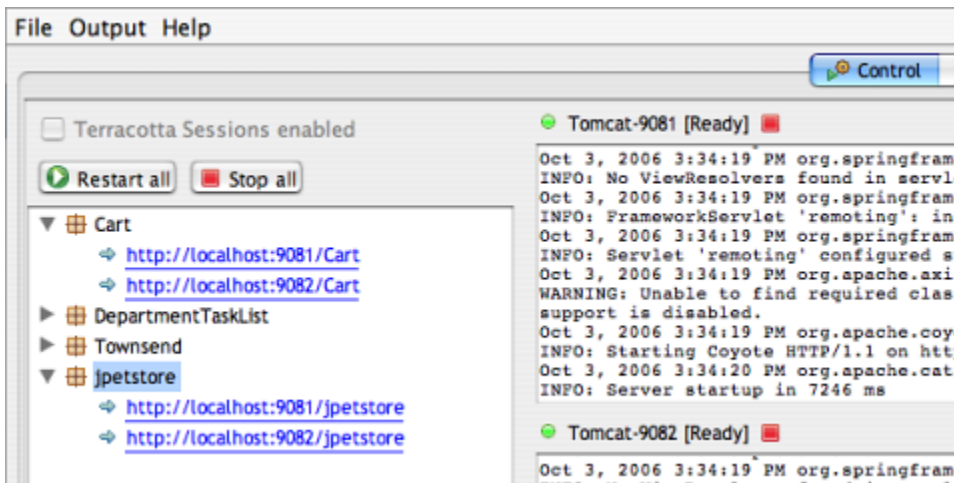
This will start two instances of the application server in two separate Java Virtual Machines. Each of these application servers has its own instance of the web applications deployed inside them. The first few applications are *Terracotta* sample applications. You should see the *JPetStore* application listed below.



View the Application Unclustered

When the application servers have started, the hyper links to the different instances of the web applications will become active.

Click on the <http://localhost:9081/jpetstore/> link. This will open your browser to the application server instance listening on port 9081.



Click on the *Enter the Store* link.

Welcome to the Spring JPetStore, by Juergen Hoeller

Based on the iBATIS JPetStore, by Clinton Begin

of Spring for the middle tier, including declarative transaction management applied to POJO business logic or JDBC for transaction management, so it allows declarative transaction management in a web application.

There are alternative Spring and Struts MVC layers built on a shared Spring middle tier.

[Enter the Store](#)

You should see the *JPetStore* homepage.



Note: If you get an error that says "Could not open JDBC Connection for transaction", the JPetStore database is not running. [Go back to that step](#) and start the database.

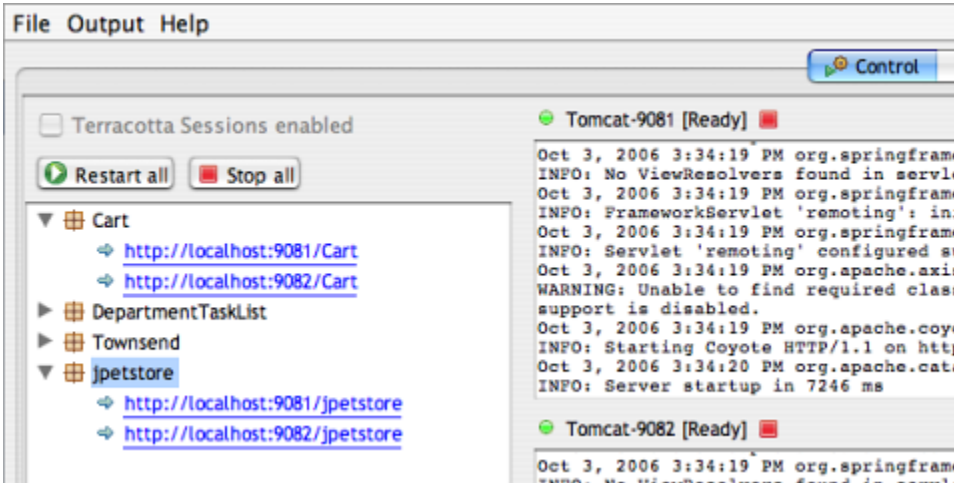
Browse to one of the product pages and add an item to your shopping cart. Your cart should look something like this:

Shopping Cart

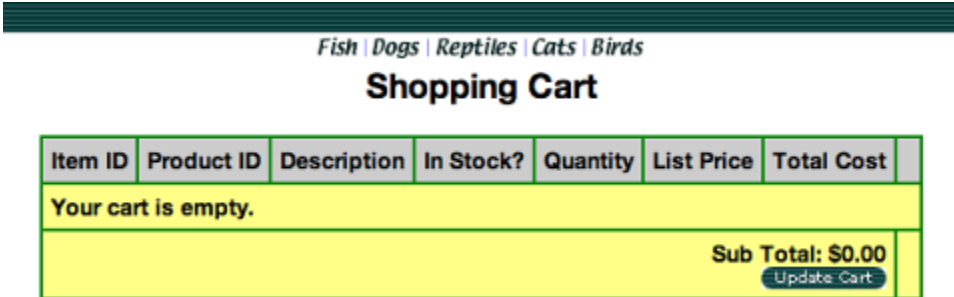
Item ID	Product ID	Description	In Stock?	Quantity	List Price	Total Cost	
1	FI-FW-01	Spotted Koi	true	1	\$18.50	\$18.50	Remove
						Sub Total: \$18.50	Update Cart

[Proceed to checkout](#)

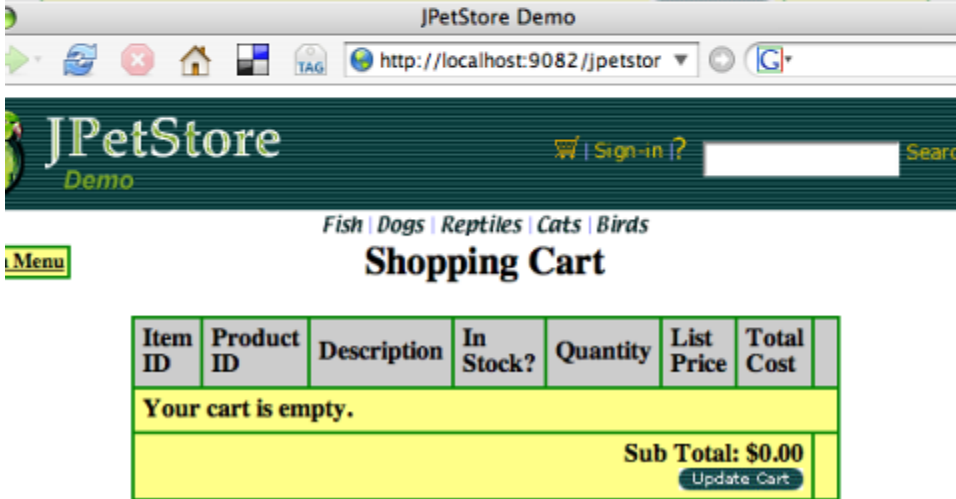
Now switch back to the *Terracotta Sessions Configurator* and click on the <http://localhost:9082/jpetstore/> link. This will open your browser to the application server instance listening on port 9082.



In the new browser window, click the shopping cart icon. You should see an empty shopping cart.



Because the two application servers, 9081 and 9082, are running in separate JVMs and they aren't clustered, the session data that contained your shopping cart items is lost when you move from one application server to the next. The result is frustrated customers and lost sales.



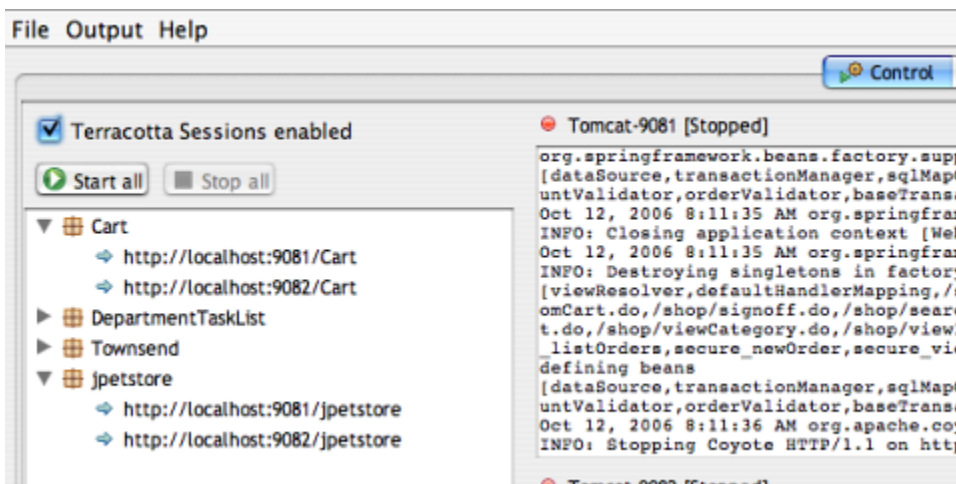
This is precisely what happens when multiple application servers are load balanced and user sessions move from one application server to another. This may happen, for example, in the event that one or more application servers have crashed or have been taken down for maintenance.

Now let's watch what happens to your shopping cart when *Terracotta* session clustering is turned on.

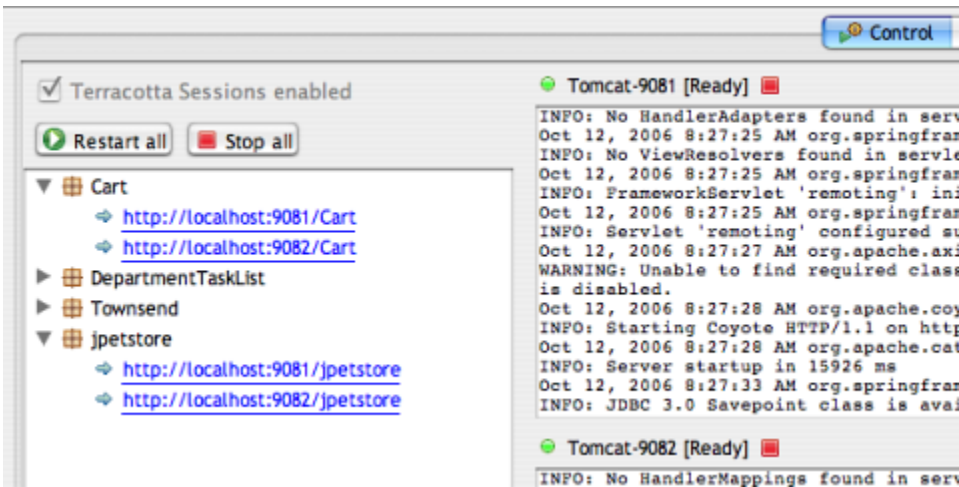
Start the Application Servers With Terracotta

Switch back to the *Terracotta Sessions Configurator* and stop the application servers by clicking on the *Stop all* button.

When the servers have stopped, click on the *Terracotta Sessions enabled* checkbox.

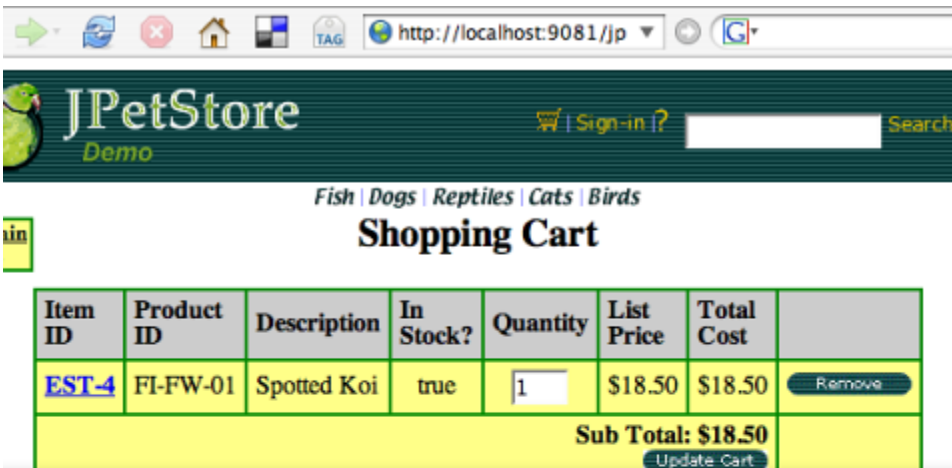


Click on the *Start all* button and, this time, the application servers will start with *Terracotta* session clustering enabled.



When the servers have started, go through the same process of adding items to your cart in one server and checking the contents of the cart in the other server:

1. Click on the <http://localhost:9081/jpetstore/> link.
2. Enter the store and add an item to your cart.
3. Switch back to the *Terracotta Sessions Configurator* and click on the <http://localhost:9082/jpetstore/> link.
4. In the new browser window, browse to the shopping cart. Your shopping cart from the server on port 9081 should be intact on the server on port 9082.



Inspect the Clustered Session Object Graph

Switch back to the *Terracotta Sessions Configurator* and click on the *Monitor* tab at the top of the window.

