

Example 3: Deploying a Tomcat application

2 minute read

Developer productivity

In this exercise, you use a service catalog item to build out a Tomcat application model. You also create a few environment models and map them to the application so that the Tomcat application can be deployed to the environments.

Note

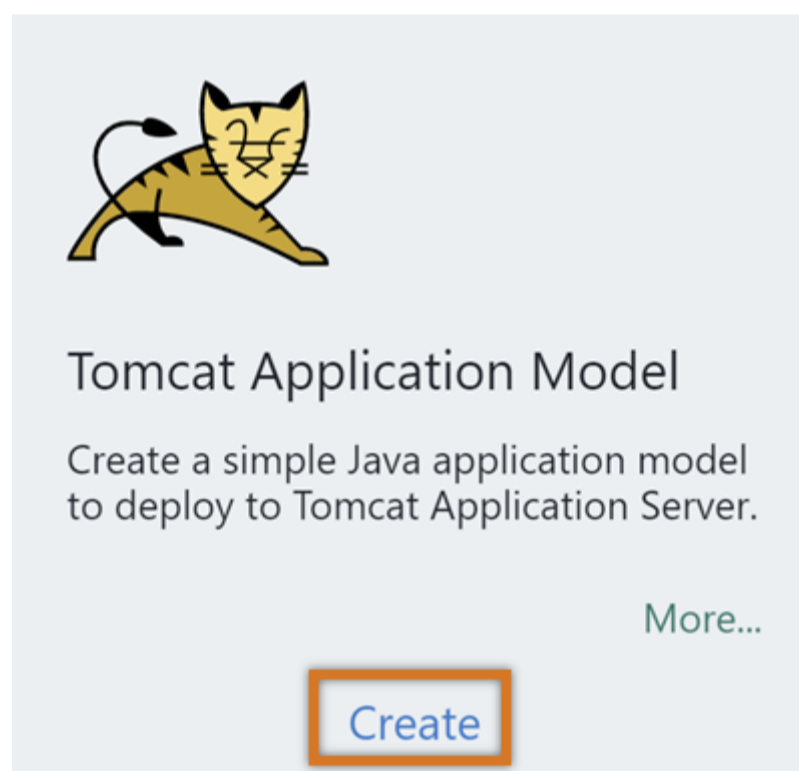
You must have access to GitHub to complete this exercise.

Get a deployable artifact

From GitHub, download [jpetstore.war](#) to your computer.

Use the Tomcat service catalog item to create a model

1. Select **Service catalog** from the banner.
2. Navigate (scroll down) to the Tomcat Application Model.



3. Select **Create**. The New Tomcat Application Model dialog appears.
4. In the **Application details** section, enter these field values:
 - Application Name:** *jpetstore*
 - Application version:** *1.0*
 - Application Project Name:** Trial Guide
5. Using the expansion controls on the right, expand the **Artifact details** section and provide this information:
 - Group Id:** *com.acme*
 - Artifact Key:** *jpetstore*
 - Version:** *7.0.16*
 - Filename:** *jpetstore.war*

Submit Feedback

Tomcat Application Model

1. Application details →

Application Name Instructions

Application Version Provide details required to create application model. Click 'OK' button to see all required fields.

Application Project Name

File to upload: Select **Choose File** to upload the jpetstore.war file you downloaded earlier.

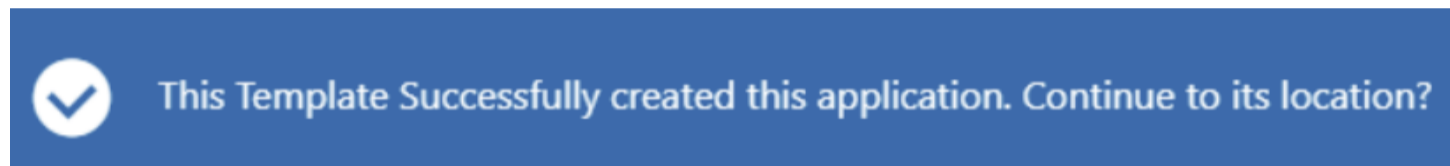
6. Expand the **Deployment details** section and add this information:

Retrieve to Folder: */tmp*

Tomcat Config: *Tomcat*

Context Path: *jpetstore*

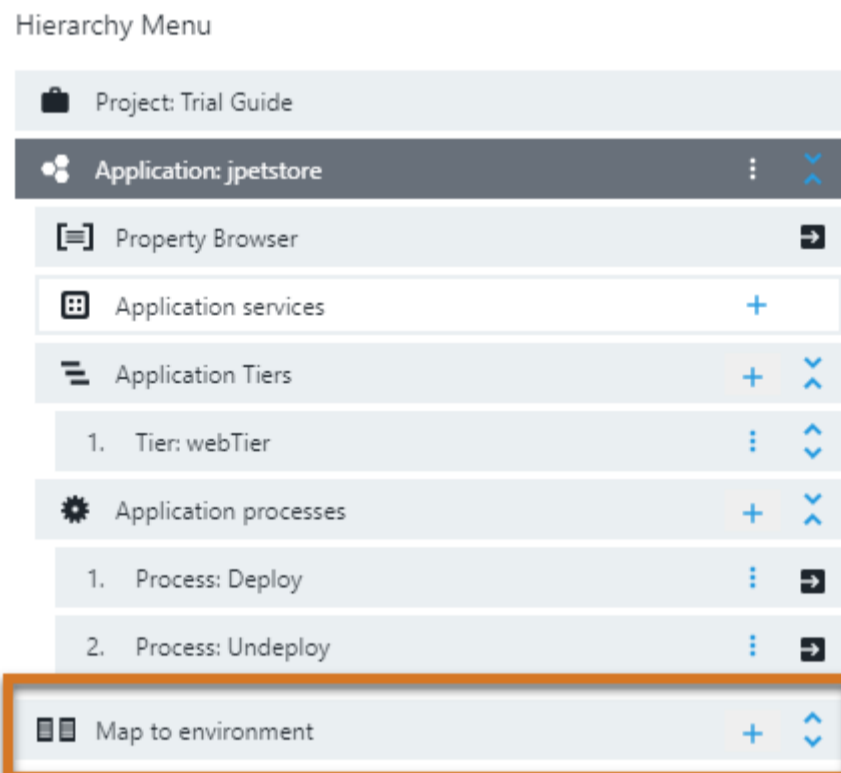
7. Select **OK** to generate the Application model. A "success" message is displayed and you are asked to navigate to the new Application model.



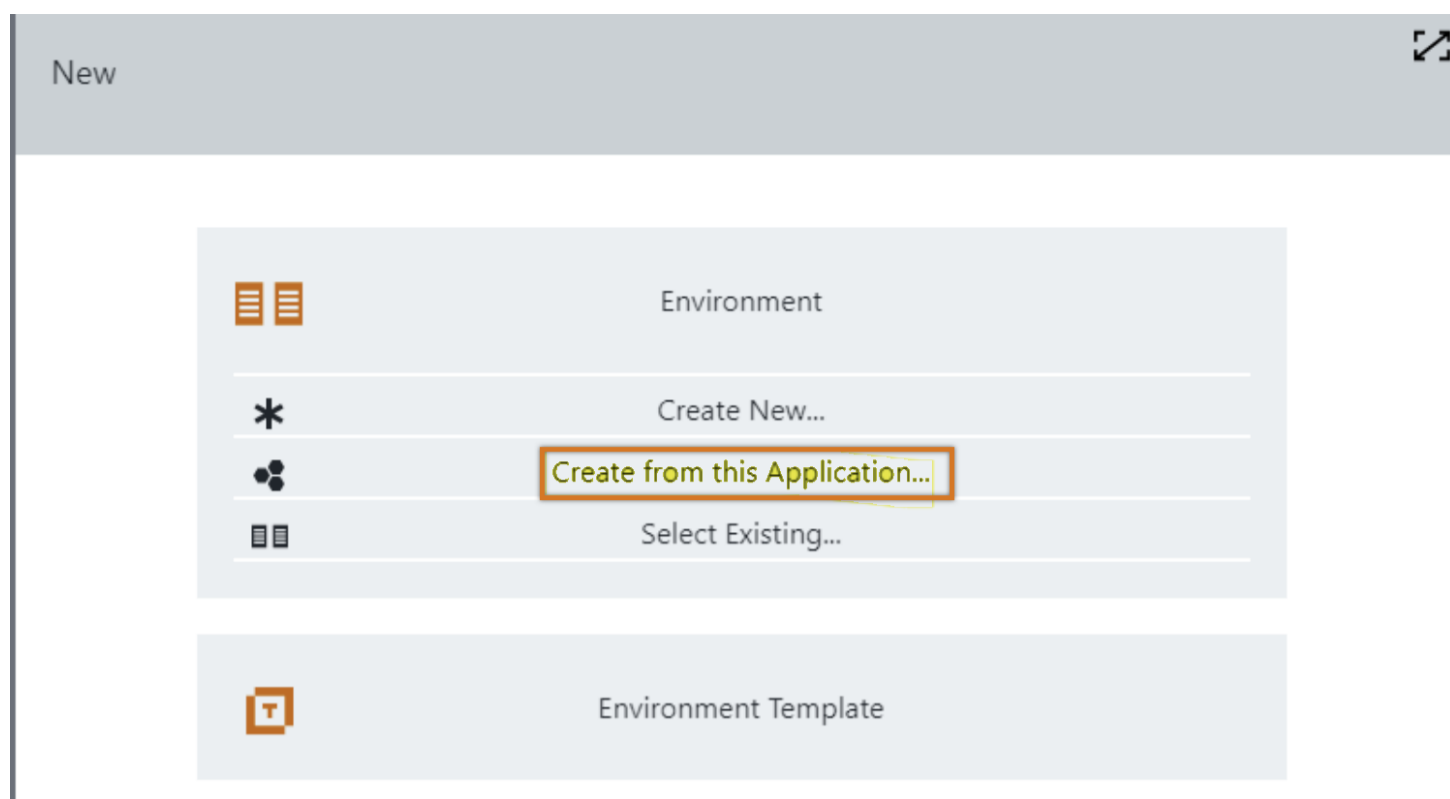
8. Select **Yes** to navigate to the new application model.

Create the QA environment

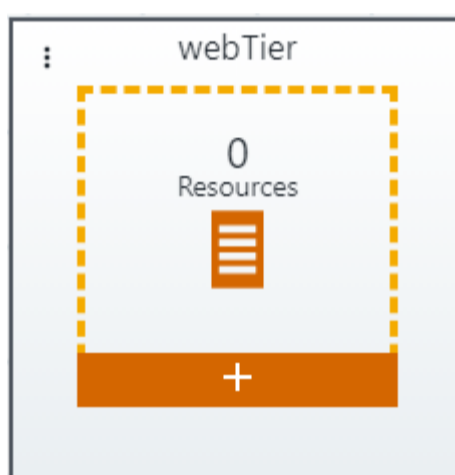
1. Select **Requires Setup** to open the Create Environment dialog.



2. Select **Create from this Application**.



3. Enter **QA** for the environment name and select the project. Select **OK** to create the environment.
4. Select the Resources plus (+) button.



5. Select **Add resources**.
6. Choose **tomcat_mysql_qa** then select **OK**.
7. Return to the application by selecting the **Application: jpetstore** link in the Hierarchy Menu.

Create the UAT environment

1. Use the plus (+) option on **Map to Environment** to create the *UAT* environment. The New Environment dialog appears.
2. Select the resource **tomcat_mysql_uat**.
3. Navigate back to the application model.

Deploy the application

1. Select **Run** and then **New Run**. The Deployment dialog appears.
2. Set **Deploy** to **QA** and turn off the **Smart Deploy**, **Check Dependency** and **Artifact Staging** options. Select **OK**.

Run Application: jpetstore
In Project: Trial Guide

Deploy ▼
QA ▼

1 Select Snapshot ▼ +? Compare →
1 Artifacts & Containers Full 1 Selected →
0 Parameters →

Smart Deploy off ? Check Dependency off ? Artifact Staging off ?

3. Select the job name to examine the Job Details.
4. Select the resource where the application was deployed and note the IP address.

Resource Details

tomcat_mysql_qa View Usage Edit Access Control

No Description


General Properties

Status: The agent is alive
Version: 2020.10.0.144378
Host OS: Linux tomcat2-tomcat-agent-qa 4.19.0-13-cloud-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64 GNU/Linux
Host Platform: linux
Agent Host Name: 35.202.90.57
Agent Host Type:
Agent Port Number:
Default Workspace:

5. Use the IP address of the resource to access the application. For example, <http://35.202.90.57:8080/jpetstore/>

JPetStore Demo

Not secure | 35.202.90.57:8080/jpetstore/ Guest

 **JPetStore** Demo Sign-in? Search

Welcome to the Spring JPetStore, by Juergen Hoeller

Based on the iBATIS JPetStore, by Clinton Begin

This application demonstrates the use of Spring for the middle tier, including declarative transaction management applied to POJO business objects. This application can easily be configured to use JTA or JDBC for transaction management, so it allows declarative transaction management in a web container without JTA.

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

[Enter the Store](#)



6. Close the Job Details dialog to return to the Deployment run.
7. Navigate to the environment where the application was deployed, which is QA.
8. Examine the inventory.
9. Create a snapshot by selecting the snapshot (camera) icon. The New Snapshot dialog appears.
10. Enter version number *1.0* in the top field, and then select **Next**. You now have a snapshot of this artifact.

 jpetstore /  1.0

 Component  Microservice  Application Process

 Component	 Artifact	 Version
1. Tomcat	com.acme:jpetstore	7.0.16

[← Example 2: Creating a service catalog item](#) | [Example 4: Running a Helm chart deployment](#) →