

Red Hat JBoss Developer Studio 8.0

Start Developing

Tutorial for first time users

Last Updated: 2017-11-19

Tutorial for first time users

Red Hat Customer Content Services

Legal Notice

Copyright © 2014 Red Hat, Inc.

This document is licensed by Red Hat under the <u>Creative Commons Attribution-ShareAlike 3.0</u> <u>Unported License</u>. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java $\ensuremath{\mathbbm R}$ is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document details how to start developing applications with JBoss Developer Studio.

Table of Contents

CHAPTER 1. START DEVELOPING WITH JBOSS DEVELOPER STUDIO	3
1.1. ABOUT START DEVELOPING	3
1.2. ABOUT THE TICKETMONSTER APPLICATION	3
1.3. TUTORIAL ASSUMPTIONS	3
1.4. TUTORIAL OUTLINE	4
CHAPTER 2. CREATING THE TICKETMONSTER APPLICATION	5
2.1. OPEN THE JBOSS PERSPECTIVE	5
2.2. CONFIGURE MAVEN TO USE THE JBOSS EAP MAVEN REPOSITORY	5
2.3. SET UP A SERVER USING RUNTIME DETECTION	6
2.4. CREATE A JAVA EE WEB PROJECT USING JBOSS CENTRAL	7
2.5. ADD AN ENTITY USING FORGE TOOLS	9
2.6. ADD AN ENTITY USING HIBERNATE TOOLS	11
2.7. PREPARE EVENT RECORDS TO POPULATE THE DATABASE	14
2.8. DEPLOY THE APPLICATION USING JBOSS SERVER TOOLS	15
2.9. ACCESS AND QUERY THE DATABASE	17
2.10. ADD A RESTFUL WEB SERVICE	19
2.11. ADD A USER INTERFACE OPTIMIZED FOR MOBILE DEVICES	22
2.12. TEST THE MOBILE USER INTERFACE USING BROWSERSIM	25
APPENDIX A. REVISION HISTORY	26

CHAPTER 1. START DEVELOPING WITH JBOSS DEVELOPER STUDIO

1.1. ABOUT START DEVELOPING

Start Developing is a step-by-step tutorial that introduces JBoss Developer Studio to new users. It aims to give you a taste of developing applications with JBoss Developer Studio. The tutorial here demonstrates how to create a web application, TicketMonster, that includes typical enterprise functionality by using JBoss Developer Studio.

In guiding you through the tutorial steps, a number of assumptions are made about your system. You must ensure you system complies with the assumptions before proceeding to complete the tutorial. For information, see Section 1.3, "Tutorial Assumptions".

The information in this tutorial is minimal. It describes one way of completing each step in the tutorial; in fact, there may be several alternatives for each step in the tutorial. For more information about alternative ways to complete tasks, see the Red Hat JBoss Developer Studio 8.0 documentation available on the Red Hat Customer Portal.

Report a bug

1.2. ABOUT THE TICKETMONSTER APPLICATION

TicketMonster is an example web application demonstrating ticket purchasing. The application provides an online environment in which users can purchase tickets for events and administrators can manage the data relating to events and ticket sales. You can see a running example of the TicketMonster application at https://ticketmonster-jdf.rhcloud.com/.

The application consists of a user interface and web services. The user interface allows users to purchase tickets for listed events and it enables administrators to access and modify event and ticket information. The user interface is optimized for desktop and mobile clients. The web services enable access to information about members, events and venues stored in a database.

TicketMonster demonstrates the combining of Red Hat and JBoss technologies and frameworks to build, test and deploy applications. For example, the running instance of TicketMonster at https://ticketmonster-jdf.rhcloud.com/ is hosted in the cloud on the OpenShift platform.

Report a bug

1.3. TUTORIAL ASSUMPTIONS

The TicketMonster application is built on a Maven-based Java EE Web project. To ensure successful completion of the tutorial, a number of assumptions are made about your system:

- JBoss Developer Studio 8.0 stand-alone or Bring-Your-Own-Eclipse (BYOE) is installed; for instructions, see Install Red Hat JBoss Developer Studio.
- JBoss Enterprise Application Platform 6.x is installed
- Corresponding JBoss Enterprise Application Platform 6.x Maven repository is installed but Maven is not necessarily configured to use it

Report a bug

1.4. TUTORIAL OUTLINE

The diversity and complexity of the TicketMonster application makes it ideal for demonstrating some of the capabilities of JBoss Developer Studio. This tutorial sets the stage for building TicketMonster. It shows you how to develop a number of the TicketMonster application features using JBoss Developer Studio:

- Open the JBoss perspective
- Configure Maven for use with JBoss Central projects
- Set up a server using JBoss Server Tools runtime detection
- Create a Java EE web project to which the TicketMonster functionality is added using JBoss Central
- Create **Event** and **Venue** entity classes for the database entities using Forge Tools and Hibernate Tools
- Prepare event records to populate the database at runtime
- Deploy the application on a runtime server using JBoss Server Tools
- Access and query the database holding the member, event and venue records
- Add a RESTful web service so that the application can retrieve the information in the database
- Add a user interface to the application that is optimized for mobile devices using Mobile Web Tools
- Test the mobile user interface using BrowserSim

You must work through these tasks in the order they are presented because the earlier tasks are prerequisites for the later tutorial tasks.



ΝΟΤΕ

For information about building the complete TicketMonster application using JBoss Developer Studio see http://www.jboss.org/ticket-monster/ on the JBoss Developer website.

Report a bug

CHAPTER 2. CREATING THE TICKETMONSTER APPLICATION

2.1. OPEN THE JBOSS PERSPECTIVE

When following this tutorial, you must use the **JBoss** perspective. Specific views, toolbars, and menu options most used by JBoss developers are available when this perspective is open. Instructions in the tutorial are given based on the **JBoss** perspective and if you use an alternative perspective you might not be able to find some of the items as detailed in the tutorial procedures. The procedure below details how to open the **JBoss** perspective.

Procedure 2.1. Open the JBoss Perspective

- 1. Click Window→Open Perspective→Other.
- 2. From the list of perspectives, double-click **JBoss**.

Report a bug

2.2. CONFIGURE MAVEN TO USE THE JBOSS EAP MAVEN REPOSITORY

The Java EE Web project, on which the tutorial TicketMonster application is built, is constructed from an enterprise Maven archetype. Maven is distributed with JBoss Developer Studio so it is not necessary to install Maven but, to be able to use the Java EE Web project enterprise archetype, Maven must be correctly configured to use the JBoss EAP 6.x Maven repository. Instructions are given here for editing your Maven configuration file from within JBoss Developer Studio to achieve this.

Procedure 2.2. Configure Maven to use the JBoss EAP Maven Repository

- 1. Click Window→Preferences, expand JBoss Tools and select JBoss Maven Integration.
- 2. Click Configure Maven Repositories.
- 3. Click Add Repository.
- 4. Click Recognize JBoss Maven Enterprise Repositories.
- 5. Navigate to path/to/jboss-eap-6.x.y-maven-repository and click OK. JBoss Maven Tools recursively scans the path searching for the Maven repository.
- 6. Modify the information in the **ID** and **Name** fields as desired, ensure the **Active by default** check box is selected and click **OK**.

dd May	en Repository
Profile	
Profile	D: jboss-eap-6-3-0-GA-maven-repository Active by default
Reposit	ory
ID:	jboss-eap-6-3-0-GA-maven-repository
Name:	JBoss EAP Maven Repository
URL:	file:/Applications/jboss-eap-6.3.0.GA-maven-repository/
 Adva 	nced
🧲 Re	cognize JBoss Maven Enterprise Repositories
?	Cancel OK



7. Click Finish and at the prompt asking if you are sure you want to update the Maven configuration file click Yes. If the specified configuration file does not exist, JBoss Maven Tools creates it.



NOTE

Maven settings, such as the configuration file, are specified in **Preferences** under Maven→User Settings. These settings can be customized.

8. Click Apply and click OK to close the Preferences window.

Report a bug

2.3. SET UP A SERVER USING RUNTIME DETECTION

Servers can be defined from within the **Java EE Web Project** wizard. But for the purposes of exploring JBoss Developer Studio the procedure below defines the server first using JBoss Server Tools runtime detection.

Procedure 2.3. Define a Server Using Runtime Detection

- 1. Click Window → Preferences, expand JBoss Tools and select JBoss Runtime Detection.
- 2. Click Add.
- 3. Select the directory containing the **JBoss EAP 6.x** installation and click **OK**. The directory is now scanned for application servers and JBoss EAP 6.x found.

Nam	e	Type	Version	Location
1	jboss-eap-6.3	EAP	6.3	/Applications/jboss-eap/jboss-eap-6.3

Figure 2.2. JBoss EAP 6.x Found by Runtime Detection

- 4. To create a server for JBoss EAP 6.x, ensure the **jboss-eap-6**.x check box is selected and click **OK**.
- 5. Click Apply and click OK to close the Preferences window. The server is listed in the Servers view.

锦 Servers 🔀	
🕨 🏭 jboss-ea	p-6.3 [Stopped]

Figure 2.3. JBoss EAP 6.x Server Listed in Servers View

6. The server is initially shown in stopped mode. Start the server by right-clicking jboss-eap6. x and clicking Start. After a short pause, the view switches to the Console view and shows the startup output of the server.

Report a bug

2.4. CREATE A JAVA EE WEB PROJECT USING JBOSS CENTRAL

The TicketMonster application is based on a Java EE web application. A wizard is provided for creating this type of project in JBoss Central and the procedure below guides you through using the wizard.

Procedure 2.4. Create a Web Application Using the Java EE Web Project Wizard

- 1. Click the **JBoss Central** tab.
- 2. In JBoss Central under Start from scratch, click Java EE Web Project to open the Java EE Web Project wizard. If JBoss Central is not visible, click Help→JBoss Central.

JBoss Central 🖾		- D
C Welcome to JBos	s	 Q Search JBoss Community Show on Startup
Start from scratch		B •
HTMLS Project	OpenShift Application	Java EE Web Project is a sample, deployable Maven 3 project to help you get your foot in the door developing with Java EE 6 on JBoss Enterprise Application Platform 6 or JBoss Application Server
🗣 AngularjS Forge	🗊 Java EE Web Project	7.1. This exclust is sature to allow you to senate a compliant into 55.6 application using 155.2.0. CDI 1.0.
Maven Project	🗃 Hybrid Mobile Project	EJB 3.1, JPA 2.0 and Bean Validation 1.0.

Figure 2.4. Java EE Web Project in JBoss Central

3. From the Target Runtime list, select jboss-eap-6.x Runtime and click Next.

reb application project deployable Maven 3 project to help you g on Platform 6 or JBoss Application Server 7 to create a compliant Java EE 6 application ec.archetypes:jboss-javaee6-webapp-arch	et your foot 7.1. n using JSF 2 hetype:7.1.3	in the door developing with Jav .0, CDI 1.0, EJB 3.1, JPA 2.0 and .Final Maven archetype
deployable Maven 3 project to help you g on Platform 6 or JBoss Application Server 7 to create a compliant Java EE 6 application ec.archetypes:jboss-javaee6-webapp-arch	et your foot 7.1. n using JSF 2 hetype:7.1.3	in the door developing with Jav .0, CDI 1.0, EJB 3.1, JPA 2.0 and .Final Maven archetype
deployable Maven 3 project to help you g on Platform 6 or JBoss Application Server 7 to create a compliant Java EE 6 application ec.archetypes:jboss-javaee6-webapp-arch	et your foot 7.1. n using JSF 2 hetype:7.1.3	in the door developing with Jav .0, CDI 1.0, EJB 3.1, JPA 2.0 and .Final Maven archetype
Buntime		
Runtime		
	Found?	Install
xample requires JBoss Enterprise Appli		Install
xample requires $m2e \ge 1.0$.		Download and Install
xample requires m2e-wtp $>= 0.16.0$.		
xample requires JBoss Maven Tools.		
	example requires JBoss Enterprise Appli example requires m2e >= 1.0. example requires m2e-wtp >= 0.16.0. example requires JBoss Maven Tools.	Example requires JBoss Enterprise Appli example requires m2e >= 1.0. example requires m2e-wtp >= 0.16.0. example requires JBoss Maven Tools. ✓

Figure 2.5. JBoss EAP 6.x Selected as Target Runtime in Java EE Web Project Wizard

- 4. Complete the fields about the project as follows:
 - In the **Project name** field, type **ticket-monster**.
 - In the Package field, type org.jboss.jdf.example.ticketmonster.

000	New Project Example	
l ava EE Web I Create a Mave	Project n-based Java EE 6 web application project	
Project name	ticket-monster	T
Package	org.jboss.jdf.example.ticketmonster	٣
Location:	t(s) to working set	v Browse
Working set:		\$ More
 Advanced 		
	< Back Next >	Cancel Finish

Figure 2.6. Completed Project Name and Package Fields in Java EE Web Project Wizard

5. Click Finish to create the project.

During project creation, the wizard imports project dependencies. When the Java EE Web Project wizard displays 'Java EE Web Project' Project is now ready, click Finish to close the wizard. A README.md file for the project automatically opens for viewing and the project is listed in the Project Explorer view.

Report a bug

2.5. ADD AN ENTITY USING FORGE TOOLS

A new entity class can be added to the TicketMonster project using Forge Tools. In the procedure below an **Event** entity class is created to hold information about the events for which users can buy tickets. This **Event** entity has id, name, description, major and picture fields.

Procedure 2.5. Add an Entity Using Forge Tools

- In the Project Explorer view, expand ticket-monster→Java Resources→src/main/java.
- 2. Click org.jboss.jdf.example.ticketmonster.model and press Ctrl+4 (Cmd+4).
- 3. From the Forge wizards list, click JPA: New Entity.
- 4. In the Entity name field, type Event and click Finish.

JPA: New Entity Create a new JPA Entity		A CONTRACTOR
Entity name: * Target package: ID Column Generation Strategy: Table Name:	Event org.jboss.jdf.example.ticketmonster.n	nodel Browse
	Cancel	Finish

Figure 2.7. New Event Entity using Forge 2 JPA: New Entity Wizard

- 5. In the **Project Explorer** view, ensure **org.jboss.jdf.example.ticketmonster**→**Event.java** is selected and press **Ctrl+4**.
- 6. From the Forge wizards list, click JPA: New Field.
- 7. Complete the fields about the field as follows:
 - In the Field name field, type name.
 - In the Length field, type 50.

JPA: New Field Create a new field		ANNA ANA
Target Entity:	org.jboss.jdf.example.ticketmonster.model.Event	
Field Name:	• name	
Field Type:	§ String	Browse
Temporal Type:		
Column Name:		
Length:	50	
Relationship Type:	• Basic 🔘 Embedded 🔘 One-to-One 🔵 One-to-Many 🔵 Many-to-One 🔘 Many-to-Many	
	Is LOB?	
	Is Transient?	
Enum Type:		
	< Back Next > Cancel	Finish

Figure 2.8. New name Field using Forge 2 JPA: New Field Wizard

- 8. Click Finish to create the field.
- 9. Repeat steps 5 and 6 to create three more fields for the Event entity as follows:
 - o description, with Field Type String and Length 1000
 - major, with Field Type Boolean
 - picture, with Field Type String
- 10. In the **Project Explorer** view, right-click **Event**. java and click **Show** In \rightarrow Forge Console.
- 11. In the Forge Console view, on the Forge 2 command line enter



and from the list of options enter the number corresponding to NotNull.

Report a bug

2.6. ADD AN ENTITY USING HIBERNATE TOOLS

A new entity class can be added to the TicketMonster project using Hibernate Tools. In the procedure below a **Venue** entity class is created to hold information about the venues where events will be held. This **Venue** entity has id, name, description and capacity fields.

Procedure 2.6. Add an Entity Using Hibernate Tools

 In the Project Explorer view, expand ticket-monster→Java Resources→src/main/java.

- 2. Right-click org.jboss.jdf.example.ticketmonster.model and click New→Class.
- 3. In the Name field, type Venue and click Finish. This creates a Venue. java file that is automatically opened in a Java editor.

000	New Java Class	
Java Class Create a new Java	class.	C
Source folder:	ticket-monster/src/main/java	Browse
Package:	org.jboss.jdf.example.ticketmonster.model	Browse
Enclosing type:		Browse
Name:	Venue	
Modifiers:	public package private protected abstract final static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add
		Remove
Which method stub	os would you like to create?	
	public static void main(String[] args)	
	✓ Inherited abstract methods	
Do you want to add	d comments? (Configure templates and default value <u>here</u>)	
	Generate comments	
?	Cancel	Finish

Figure 2.9. New Venue Class using Java Class Wizard

4. Add fields to the entity by adding the following lines in between the braces of the Venue class

```
private Long id;
private String name;
private String description;
private int capacity;
```

- 5. Save the Venue.java file by pressing Ctrl+S (Cmd+S).
- 6. Add get and set methods by right-clicking on a new line after private int capacity; and clicking Source→Generate Getters and Setters.
- 7. Click Select All and click OK.

000	Generate Getters and Setter	S
Select getters ar	d setters to create:	
✓ ► □ capa	city ription	Select All
✓ ► □ id ✓ ► □ name	2	Deselect All
		Select Getters
		Select Setters
After 'capacity' Sort by:		*
Sort by:	leattan pales	•
Access modifi	er	•
• public	 protected package p synchronized 	rivate
Generate me	thod comments	
The format of th	e getters/setters may be configured on the Co	de Templates preference page.
i 8 of 8 select	ed.	
?		Cancel OK

Figure 2.10. List of Attributes Selected for Creating Get and Set Methods

- 8. Save the Venue. java file.
- 9. Make the class an entity by right-clicking anywhere in the Venue class and clicking Source→Generate Hibernate/JPA annotations.
- 10. Ensure org.jboss.jdf.example.ticketmonster.model.Venue is selected and click Next.

Class	1		
org.jboss.jdf.exan	ple.ticketmonst		
Preferred location	of Annotations:	Fields	÷
Default string len	gth (255 by default):		255
Enable optimistic	locking:		

Figure 2.11. Venue Class Selected for Adding JPA Annotations

11. Click Finish and save the Venue. java file.

Report a bug

2.7. PREPARE EVENT RECORDS TO POPULATE THE DATABASE

This web project uses H2 for data storage by default. H2 is a Java database integrated with the application servers of JBoss EAP 6.x. It is an in-memory database which means that the data does not persist once the application using the database stops running. For this reason, seed data must be stored in a file within the project and added to the database each time the application starts. While it might not seem practical to have to construct the database each time the application starts, this type of data storage is useful during development for testing purposes.



WARNING

The use of H2 for data storage is for testing purposes only and must not be used during production.

The procedure below demonstrates preparing two sample Event records for addition to the H2 database once the TicketMonster application starts.

Procedure 2.7. Prepare Event Records

- In the Project Explorer view, expand ticket-monster→Java Resources→src/main/resources.
- 2. Double-click the import.sql file to open it for editing.
- 3. Double-click the import.sql view label to make the editor fill the window.
- 4. Create a new Event record by copying the following on one line after the existing Member record

insert into Event (id, name, description, major, picture, version)
values (1, 'Shane''s Sock Puppets', 'This critically acclaimed
masterpiece...', true, 'http://dl.dropbox.com/u/65660684/640pxCarnival_Puppets.jpg', 1);

5. Create a second new Event record by copying the following on one line after the first new Event record

insert into Event (id, name, description, major, picture, version)
values (2, 'Rock concert of the decade', 'Get ready to rock...',
true, 'http://dl.dropbox.com/u/65660684/640pxWeir%2C_Bob_(2007)_2.jpg', 1);

```
🖪 import.sql 🛿
```

Conn	nection profile	
Туре	e: Name: Data	<u>)</u> ase:
	JBoss, Home of Professional Open Source Copyright 2013, Red Hat, Inc. and/or its affiliates, and individual contributors by the @authors tag. See the copyright.txt in the distribution for a full listing of individual contributors. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/Licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.	
in in in	You can use this file to load seed data into the database using SQL statements sert into Member (id, name, email, phone_number) values (0, 'John Smith', 'john.smith@mailinator.com', '2125551212') sert into Event (id, name, description, major, picture, version) values (1, 'Shane''s Sock Puppets', 'This critically acclai sert into Event (id, name, description, major, picture, version) values (2, 'Rock concert of the decade', 'Get ready to rock	med

Figure 2.12. Content of the import.sql File for this TicketMonster Workflow

- 6. Save the import.sql file.
- 7. Double-click the import.sql view label to make the editor return to its previous size.

Report a bug

2.8. DEPLOY THE APPLICATION USING JBOSS SERVER TOOLS

The TicketMonster application can be deployed on the JBoss EAP 6.x server created earlier in the tutorial. The procedure below guides you through how to do this.

Procedure 2.8. Deploy the Application Using JBoss Server Tools

- In the Project Explorer view, right-click the ticket-monster project name and click Run As→Run on Server.
- 2. In the Run On Server wizard, ensure Choose an existing server is selected.
- 3. From the Select the server that you want to use table, select jboss-eap-6.x and click Finish.

If not already in view, the **Console** view automatically opens and shows the runtime server output. Also, a **Web Browser** automatically opens and displays the web page of the running TicketMonster application.

000	Run On Server	
Run On Server Select which server to use		
 How do you want to select the server Choose an existing server Manually define a new server Select the server that you want to use 	?	
type filter text	e server? er erver t to use: State En Stopped latform (EAP) 6.1+ Colu	8
Server Server Image: Server <td< th=""><td>(EAP) 6.1+ ing this project</td><td>State Stopped Columns</td></td<>	(EAP) 6.1+ ing this project	State Stopped Columns
?	Next >	Cancel Finish

Figure 2.13. JBoss EAP 6.x Selected in Run on Server Wizard

In the Web Browser, enter sample Member data in the Member Registration fields and click **Register**. The submitted data is displayed in the Members table, which is visible at the bottom of the web page.

Welcome to JBoss!	RED HAT JBOSS' ENTERPRISE APPLICATION PLATFOR
You have successfully deployed a Java EE 6 Enterprise Application.	Learn more about JBoss Enterprise Application Platform 6.
Member Registration	Documentation Product Information
Name:	
Email:	
Phone #:	
Register	
Members	
Id Name Email Phone # REST UR	L
0 John Smith john.smith@mailinator.com 2125551212 /rest/mer	mbers/0
OFFT 101 fax all mambars: (rack/mambars	

Figure 2.14. Sample Data in the Member Registration Fields

Report a bug

2.9. ACCESS AND QUERY THE DATABASE

This web project uses the in-memory database, H2, of the application servers in JBoss EAP 6.x. The H2 console web application can be used to access the H2 database but this is not packaged with JBoss EAP 6.x. As the procedure below explains, the H2 console web application must be downloaded and deployed on the application server from outside JBoss Developer Studio before the H2 database can be accessed.

Procedure 2.9. Access and Query the Database

- 1. Go to http://www.jboss.org/quickstarts/eap/h2-console/.
- 2. Click Download.
- 3. Extract the contents of the archive

\$ unzip jboss-eap-quickstarts-<version>.zip

4. Copy the h2console.war file to the JBoss EAP 6.x deployments directory

\$ cp jboss-eap-quickstarts-<version>/h2-console/h2console.war
/path/to/jboss-eap/standalone/deployments/

5. Confirm the H2 console web application is running by looking at the server output in the **Console** view in JBoss Developer Studio.

45 Servers	Console	: 83											100		· •	• • •
jboss-eap-6	5.3 [JBoss Ap	plication Server S	tartup Confi	iguration] /Li	ibrary/Ja	wa/Jawa\	virtualMa	chines/ja	dk1.7.0_5	5.jdk/C	Content	s/Home	/bin/java	1 (17 C	ct 2014 2	:06:33 pm
14:06:53,	876 INFO	[org.jboss.as	.server.d	deployment]] (MSC	servic	three	nd 1-7)	JBAS01	5876:	Start	ing de	ploymen	t of	"h2cons	ole.war'
14:06:53,	986 WARN	[org.jboss.as	.dependen	ncy.unsupp	orted]	(MSC s	service	thread	1-7) J	BAS018	8568: 1	Deploy	ment "d	leploy	ment.h2	console.
14:06:53,	929 INF0	[org.jboss.we	b] (Serve	erService '	Thread	Pool -	- 12) .	BAS018	210: Re	gister	web	contex	t: /h2c	onsol	e	
14:06:53,	969 INFO	[org.jboss.as	.server]	(Deployment	ntScann	ner-thr	eads -	1) JBA	\$018559	: Depl	loyed	"h2con	sole.wo	ir" (r	untime-	name : "

Figure 2.15. Console Output showing h2console.war Deployed

- 6. In the IDE open a new Web Browser by clicking Window→Show View→Other, expand General and double-click Internal Web Browser.
- 7. In the address bar of the Web Browser, enter http://localhost:8080/h2console.
- 8. In the Login area, in the JDBC URL field type jdbc:h2:mem:ticket-monster and in both the User Name and Password fields type sa. These settings are defined in the ticketmonster-ds.xml file in the ticket-monster/src/main/webapp/WEB-INF directory.

ا 🎯	nternal Web B	rowser 🛛		-	
	🗢 🔳 🦑	http://localhost:8080/h2console/console/login.jsp?jsessionid=5540ba5a629a7691c1ff137d010e30d0	~	►	•
(English	≎ Preferences Tools Help			
	Login				
	Saved Settings:	Generic H2 (Embedded)			
	Setting Name:	Generic H2 (Embedded) Save Remove			
	Driver Class:	org.h2.Driver			
	JDBC URL:	jdbc:h2:mem:ticket-monster			
	User Name:	sa			
	Password:	••			
		Connect Test Connection			

Figure 2.16. Completed H2 Login Web Page

9. Click **Connect** to connect to the database.

The Event, Member and Venue entities created in the TicketMonster project are displayed under jbdc:h2:mem:ticket-monster. Expanding the entities shows the fields associated with each. The records stored in the database are viewed using SQL search statements. For example, to display all the Event records stored in the database, click Event, which creates a SQL search statement based on Event in the SQL statement field, and click Run. The details of the Event sample records you added to import.sql earlier in the tutorial are visible in the table below the SQL statement field.

🥹 Internal Web Browser 🛿	
🗢 🗢 🔳 🖑 http://localh	nost:8080/h2console/console/login.do?jsessionid=c35514c82657ba81841e9038473fef47
💦 🤣 🗹 Auto commit 🗠 🏼 🖉	Max rows: 1000 🗢 💽 🔳 😩 Auto complete Normal 🗧 🕐
jdbc:h2:memticket-monster □ EVENT □ D □ DESCRIPTION □ MAJOR □ NAME □ PICTURE □ VERSION □ Members	Run (Ctri+Enter) Clear SQL statement. SELECT * FROM EVENT
	SELECT * FROM EVENT;
	ID DESCRIPTION MAJOR NAME PICTURE VERSION
DESCRIPTION	1 This critically acclaimed masterpiece TRUE Shane's Sock Puppets http://dl.dropbox.com/u/65660684/640px.Carnival_Puppets.jpg 1
E INAME	2 Get ready to rock TRUE Rock concert of the decade http://dl.dropbox.com/u/65660684/640px.Weir%2C_Bob_(2007)_2.jpg 1
	(2 rows, 4 ms) Edit

Figure 2.17. Event Records Stored in the H2 Database

Report a bug

2.10. ADD A RESTFUL WEB SERVICE

For the TicketMonster application to be able to access the information stored in the database, web services are needed. The procedure below creates a RESTful web service that returns all the events in the database. It generates a POJO (plain old Java object) and adds JAX-RS annotations to create an endpoint.

Procedure 2.10. Add a RESTful Web Service

- In the Project Explorer view, expand ticket-monster→Java Resources→src/main/java.
- 2. Right-click org.jboss.jdf.example.ticketmonster.rest and click New→Class.
- 3. In the Name field, type EventService and click Finish. This creates a EventService.java file that is automatically opened in a Java editor.

	New Java Class	
ava Class Create a new Java c	lass.	C
Source folder:	ticket-monster/src/main/java	Browse
Package:	org.jboss.jdf.example.ticketmonster.rest	Browse
Enclosing type:		Browse
Name:	EventService	
Modifiers:	public package private protected abstract final static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add
		Remove
Which method stubs	would you like to create?	
	public static void main(String[] args) Constructors from superclass	
	✓ Inherited abstract methods	
	comments? (Configure templates and default value here)	
Do you want to add		

Figure 2.18. Completed Name Field in Java Class Wizard

4. Add the following lines immediately above public class EventService {

```
@Path("/events")
@RequestScoped
```

and add the following lines in between the braces of the **EventService** class

```
@Inject
private EntityManager em;
@GET
@Produces(MediaType.APPLICATION_JSON)
public List<Event> getAllEvents() {
```

```
final List<Event> results =
  em.createQuery("select e from Event e order by
e.name").getResultList();
  return results;
}
```

- 5. Save the EventService.java file.
- 6. Resolve the errors relating to missing imports by right-clicking anywhere in the **EventService** class and clicking **Source→Organize Imports**.
- 7. For each class name that is not unique, select the class to use as follows and click Next:
 - For MediaType select javax.ws.rs.core.MediaType
 - For Event select org.jboss.jdf.example.ticketmonster.model.Event
 - For Produces select javax.ws.rs.Produces
 - For List select java.util.List
 - For Inject select java.inject.Inject
 - For RequestScoped select java.enterprise.context.RequestScoped

000	Organize Imports	
Choose	type to import:	Page 1 of 6
G ^F java Gjava	a.awt.PageAttributes.MediaType ax.ws.rs.core.MediaType	
?	< Back Next > Cancel	Finish

Figure 2.19. javax.ws.rs.core.MediaType Selected for MediaType Class

- 8. When all the classes have been chosen, click **Finish**. The import statements corresponding to the class names selected in the previous step are added to the **EventService.java** file.
- 9. Save the EventService.java file.
- 10. In the Servers view expand jboss-eap-6.x, right-click ticket-monster and click Full Publish to update the deployed version of the application.
- 11. Open a new Web Browser and in the address bar of the Web Browser enter http://localhost:8080/ticket-monster/rest/events. This shows the output of the new RESTful endpoint.

Report a bug

2.11. ADD A USER INTERFACE OPTIMIZED FOR MOBILE DEVICES

The TicketMonster user interface displays on desktop and mobile clients but it is not optimized for the latter. The procedure below uses the jQuery Mobile API to create a mobile optimized user interface. The mobile user interface created varies from the existing user interface by displaying the names of the events returned by the previously created web service rather than the Member Registration web page.

Procedure 2.11. Add a User Interface Optimized for Mobile Devices

- 1. In the **Project Explorer** view, expand ticket-monster→src→main.
- 2. Right-click webapp and click New→HTML file.
- 3. Complete the information about the file as follows:
 - In the Enter or select the parent folder field, type ticketmonster/src/main/webapp.
 - In the File name field, type mobile.html.

000	New HTML File		
HTML Create a new HTML file.			<>
Enter or select the parent folder:			
ticket-monster/src/main/webapp			
 RemoteSystemsTempFiles ticket-monster .settings src main java java resources webapp test target 			
File name: mobile.html			
Advanced >>			
?	Next >	Cancel	Finish

Figure 2.20. New HTML File using the New HTML File Wizard

4. Click Next.

5. From the Templates list, select HTML5 jQuery Mobile Page (1.4) and click Finish. This creates a mobile.html file that is automatically opened in the Visual/Source tab of the Visual Page Editor.

Select HTML Template	
Select a template as initial con	nt in the HTML page.
☑ Use HTML Template	
Templates:	
Name	Description
Facelets XHTML Page	Facelets XHTML Page Template
HTML5 jQuery Mobile Page (1.	HTML5 jQuery Mobile 1.3 Template
HTML5 jQuery Mobile Page (1.	HTML5 jQuery Mobile 1.4 Template
New Facelet Composition Page	Creates a new Facelet page for use with a
New Facelet Footer	Creates a footer for use with the Facelet te
New Facelet Header	Creates a header for use with the Facelet t
New Facelet Template	Creates a basic header/content/footer Fac
New HTML File (4.01 frameset	html 4.01 frameset
Preview:	
html <html> <head> <title>jQuery Mobil <meta <="" content="text/html; charset=utf-8" http-equiv="C</td><td>Template</title>
tent-Type" td=""/></title></head></html>	
Templates are 'New H	IL' templates found in the <u>HTML Templates</u> preference page

Figure 2.21. Selected HTML5 jQuery Mobile Page (1.4) Template in New HTML File Wizard

6. List the events returned by the web service in the mobile user interface by adding the following lines immediately after the alert ("Ready To Go"); line

```
$.getJSON("rest/events", function(events) {
    // console.log("returned are " + events);
    var listOfEvents = $("#listOfItems");
    listOfEvents.empty();
    $.each(events, function(index, event) {
        // console.log(event.name);
        listOfEvents.append("<a href='#'>" + event.name + "</a>");
        });
        listOfEvents.listview("refresh");
    });
```



Figure 2.22. Completed mobile.html File

7. Save the mobile.html file.

View the mobile interface, by opening an external web browser and in the address bar entering http://localhost:8080/ticket-monster/mobile.html. At the Ready To Go prompt, click OK. The names of the events in the database are displayed.



NOTE

HTML pages are deployed immediately so there is no need to do a full publish of the TicketMonster application to see changes.

📽 jQuery Mobile	+					
💮 💽 localhost:8080/ticket-mo	onster/mobile.html		☆♥@ 🕙 Coogle	Q	🤳 🏠	0
		jQuery Mobile				
S Filter items						
Rock concert of the decad	е					0
Shane's Sock Puppets						ο
		www.jboss.org/devel	loper			

Figure 2.23. Events Displayed via Mobile Interface in External Web Browser

Report a bug

2.12. TEST THE MOBILE USER INTERFACE USING BROWSERSIM

Mobile user interfaces are best tested on mobile devices and JBoss Developer Studio provides BrowserSim for this purpose. The procedure below guides you through viewing the TicketMonster application on simulated mobile devices using BrowserSim.

Procedure 2.12. Test the Mobile User Interface Using BrowserSim

- 1. On the toolbar click the **BrowserSim** icon . A simulated device is displayed.
- 2. At the **Ready To Go** prompt, click **Close**. The names of the events in the database are displayed.



ΝΟΤΕ

If the names of the events in the database are not displayed, check the address bar of the simulated device. The address bar must state http://localhost:8080/ticket-monster/mobile.html.

3. To rotate the mobile device, click in any corner of the simulated device.

	JQuery Mobile	
		Q Filter items
D	the decade	Rock concert
Ð	uppets	Shane's Sock
	uppets	Shane's Sock

Figure 2.24. Rotated Simulated Device

- 4. Change the type of simulated device displayed by right-clicking anywhere on the simulated device, clicking **Skin** and selecting from the different skins listed.
- 5. Close BrowserSim by right-clicking anywhere on the simulated device and clicking **Close**.

Report a bug

APPENDIX A. REVISION HISTORY

Revision 8.0.0-3Fri Nov 21 2014TOOLSDOC-552: Section about using JBoss perspective addedTOOLSDOC-553: Typo in Forge Tools procedure fixed

Michelle Murray

Revision 8.0.0-2 Generated for 8.0.0 release Thu Oct 16 2014

Michelle Murray