

# Red Hat CodeReady Studio 12.19

## Getting Started with CodeReady Studio Tools

Introduction to using Red Hat CodeReady Studio tools

Last Updated: 2021-06-03

# Red Hat CodeReady Studio 12.19 Getting Started with CodeReady Studio Tools

Introduction to using Red Hat CodeReady Studio tools

Eva-Lotte Gebhardt egebhard@redhat.com

Levi Valeeva levi@redhat.com

Supriya Takkhi sbharadw@redhat.com

Yana Hontyk yhontyk@redhat.com

## Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

http://creativecommons.org/licenses/by-sa/3.0/

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

Java <sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

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

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

The OpenStack <sup>®</sup> 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 compilation of topics contains information on how to start using Red Hat CodeReady Studio Tools for efficient development.

## Table of Contents

CHAPTER 1. GIT BASICS IN CODEREADY STUDIO       5         11. SETTING UP GIT PERSPECTIVE       5         12. MANAGING REPOSITORIES IN GIT PERSPECTIVE       7         12.1. Creating an ewiding for the pository       7         12.2. Adding an existing local Git repository       9         12.3. Choing an existing dire pository       9         12.4. Adding a remote for the repository       10         13.1. MANAGING BRANCHES IN GIT PERSPECTIVE       16         13.2. Working in the branch       16         13.2. Working in the branch       19         13.3. Updating your local repository       20         14. COMMITTING AND PUSHING CHANGES       21         CHAPTER 2. MAVEN BASICS IN CODEREADY STUDIO       23         21. CREATING A NEW MAVEN PROJECT       27         22.1. Importing an existing locally stored Maven project       20         23. CREATING A NEW MAVEN PROJECT       27         24. ADDING A MAVEN MODULE       22         25. ADDING MAVEN SUPPONET TO AN EXISTING NON-MAVEN PROJECT       27         26. ADDITIONAL RESOURCES       39         CHAPTER 3. APPLICATION DEPLOYMENT IN CODEREADY STUDIO       40         31. CONFIGURING A REMOTE SERVER       43         32. CONFIGURING A REMOSTORIES       50         32. CONFIGURING A PENDESTO	MAKING OPEN SOURCE MORE INCLUSIVE	4
2.1. CREATING A NEW MAVEN PROJECT232.2. IMPORTING EXISTING MAVEN PROJECTS272.2.1. Importing an existing locally stored Maven project282.2.2. Importing an existing remotely stored Maven project302.3. CREATING A NEW MAVEN MODULE322.4. ADDING A MAVEN DEPENDENCY TO A MAVEN PROJECT352.5. ADDING A MAVEN DEPENDENCY TO A MAVEN PROJECT372.6. ADDITIONAL RESOURCES39CHAPTER 3. APPLICATION DEPLOYMENT IN CODEREADY STUDIO3.1. CONFIGURING A LOCAL SERVER403.2. CONFIGURING A REMOTE SERVER433.3. DEPLOYING A REMOTE SERVER433.3. DEPLOYING AN APPLICATION40CHAPTER 4. JBOSS EAP AND JBOSS WFK BASICS IN CODEREADY STUDIOCHAPTER 4. JBOSS EAP AND JBOSS WFK BASICS IN CODEREADY STUDIOCIAPTER 5. OPENSHIFT BASICS IN CODEREADY STUDIO604.2. SETTING UP JBOSS EAP53CHAPTER 5. OPENSHIFT APPLICATION EXPLORER VIEW615.1. SETTING UP THE OPENSHIFT APPLICATION EXPLORER VIEW615.2. CONNECTING THE OPENSHIFT CLUSTER USING OPENSHIFT APPLICATION EXPLORER5.3. SETTING UP A DEVELOPER SANDBOX USING OPENSHIFT APPLICATION EXPLORER5.4. DEVELOPER SANDBOX USING OPENSHIFT APPLICATION EXPLORER5.4.1. Creating an empty OpenShift project using OpenShift Application Explorer7.4.3. Creating a devifie component using OpenShift Application Explorer7.5.3. Creating	<ul> <li>1.1. SETTING UP GIT PERSPECTIVE</li> <li>1.2. MANAGING REPOSITORIES IN GIT PERSPECTIVE</li> <li>1.2.1. Creating a new Git repository</li> <li>1.2.2. Adding an existing local Git repository</li> <li>1.2.3. Cloning an existing Git repository</li> <li>1.2.4. Adding a remote for the repository</li> <li>1.3. MANAGING BRANCHES IN GIT PERSPECTIVE</li> <li>1.3.1. Creating a new branch</li> <li>1.3.2. Working in the branch</li> <li>1.3.3. Updating your local repository</li> </ul>	5 7 8 9 11 16 16 19 20
3.1. CONFIGURING A LOCAL SERVER403.2. CONFIGURING A REMOTE SERVER433.3. DEPLOYING AN APPLICATION47CHAPTER 4. JBOSS EAP AND JBOSS WFK BASICS IN CODEREADY STUDIO4.1. CONFIGURING MAVEN REPOSITORIES504.2. SETTING UP JBOSS EAP53CHAPTER 5. OPENSHIFT BASICS IN CODEREADY STUDIO615.1. SETTING UP THE OPENSHIFT APPLICATION EXPLORER VIEW615.2. CONNECTING TO THE OPENSHIFT CLUSTER USING OPENSHIFT APPLICATION EXPLORER635.2.1. Browser-based token retrieval655.3. SETTING UP A DEVELOPER SANDBOX USING OPENSHIFT TOOLS5.4. BUILDING AN APPLICATION BASED ON DEVFILES695.4.1. Creating an empty project5.4.3. Creating an empty OpenShift project using OpenShift Application Explorer715.4.4. Devfile registry management using OpenShift Application Explorer725.4.4. Devfile registry management using OpenShift Application Explorer745.5. BUILDING AN APPLICATION BASED ON S2I FILES765.5.1. Creating a launcher project775.5.2. Creating an empty OpenShift project in OpenShift Application Explorer795.6. DEPLOYING A COMPONENT ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.7. DEFINING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER	<ul> <li>2.1. CREATING A NEW MAVEN PROJECT</li> <li>2.2. IMPORTING EXISTING MAVEN PROJECTS <ul> <li>2.2.1. Importing an existing locally stored Maven project</li> <li>2.2.2. Importing an existing remotely stored Maven project</li> </ul> </li> <li>2.3. CREATING A NEW MAVEN MODULE</li> <li>2.4. ADDING A MAVEN DEPENDENCY TO A MAVEN PROJECT</li> <li>2.5. ADDING MAVEN SUPPORT TO AN EXISTING NON-MAVEN PROJECT</li> </ul>	23 27 28 30 32 35 37
4.1. CONFIGURING MAVEN REPOSITORIES504.2. SETTING UP JBOSS EAP53CHAPTER 5. OPENSHIFT BASICS IN CODEREADY STUDIO5.1. SETTING UP THE OPENSHIFT APPLICATION EXPLORER VIEW615.2. CONNECTING TO THE OPENSHIFT CLUSTER USING OPENSHIFT APPLICATION EXPLORER635.2.1. Browser-based token retrieval655.3. SETTING UP A DEVELOPER SANDBOX USING OPENSHIFT TOOLS675.4. BUILDING AN APPLICATION BASED ON DEVFILES695.4.1. Creating an empty project695.4.2. Creating an empty OpenShift project using OpenShift Application Explorer715.4.3. Creating a devfile component using OpenShift Application Explorer745.5. BUILDING AN APPLICATION BASED ON S2I FILES765.5.1. Creating a launcher project775.5.2. Creating an empty OpenShift project in OpenShift Application Explorer795.5.3. Creating an S2I component using OpenShift Application Explorer795.5.3. Creating an S2I component using OpenShift Application Explorer795.6. DEPLOYING A COMPONENT ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.7. DEFINING AN EXTERNAL ACCESS URL USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION	3.1. CONFIGURING A LOCAL SERVER 3.2. CONFIGURING A REMOTE SERVER	40 43
5.1. SETTING UP THE OPENSHIFT APPLICATION EXPLORER VIEW615.2. CONNECTING TO THE OPENSHIFT CLUSTER USING OPENSHIFT APPLICATION EXPLORER635.2.1. Browser-based token retrieval655.3. SETTING UP A DEVELOPER SANDBOX USING OPENSHIFT TOOLS675.4. BUILDING AN APPLICATION BASED ON DEVFILES695.4.1. Creating an empty project695.4.2. Creating an empty OpenShift project using OpenShift Application Explorer715.4.3. Creating a devfile component using OpenShift Application Explorer725.4.4. Devfile registry management using OpenShift Application Explorer745.5. BUILDING AN APPLICATION BASED ON S2I FILES765.5.1. Creating a launcher project775.5.2. Creating an empty OpenShift project in OpenShift Application Explorer795.5.3. Creating an empty OpenShift project in OpenShift Application Explorer795.6. DEPLOYING A COMPONENT ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER825.7. DEFINING AN EXTERNAL ACCESS URL USING OPENSHIFT APPLICATION EXPLORER825.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER84	4.1. CONFIGURING MAVEN REPOSITORIES	50
	<ul> <li>5.1. SETTING UP THE OPENSHIFT APPLICATION EXPLORER VIEW</li> <li>5.2. CONNECTING TO THE OPENSHIFT CLUSTER USING OPENSHIFT APPLICATION EXPLORER</li> <li>5.2.1. Browser-based token retrieval</li> <li>5.3. SETTING UP A DEVELOPER SANDBOX USING OPENSHIFT TOOLS</li> <li>5.4. BUILDING AN APPLICATION BASED ON DEVFILES</li> <li>5.4.1. Creating an empty project</li> <li>5.4.2. Creating an empty openShift project using OpenShift Application Explorer</li> <li>5.4.3. Creating a devfile component using OpenShift Application Explorer</li> <li>5.4.4. Devfile registry management using OpenShift Application Explorer</li> <li>5.5. BUILDING AN APPLICATION BASED ON S2I FILES</li> <li>5.5.1. Creating a launcher project</li> <li>5.5.2. Creating an empty OpenShift project in OpenShift Application Explorer</li> <li>5.3. Creating an S2I component using OpenShift Application Explorer</li> <li>5.6. DEPLOYING A COMPONENT ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER</li> <li>5.7. DEFINING AN EXTERNAL ACCESS URL USING OPENSHIFT APPLICATION EXPLORER</li> </ul>	61 63 65 67 69 71 72 74 76 77 79 82 82
	CHAPTER 6. QUARKUS TOOLS BASICS IN CODEREADY STUDIO	86

6.1. CREATING A NEW QUARKUS PROJECT	86
6.2. RUNNING A QUARKUS APPLICATION	89
6.3. DEBUGGING A QUARKUS APPLICATION	91
6.4. USING LANGUAGE SUPPORT IN CODEREADY STUDIO	93
6.4.1. Using Quarkus content assist	93
6.4.2. Enabling language support for MicroProfile REST Client properties	95
CHAPTER 7. HIBERNATE TOOLS BASICS IN CODEREADY STUDIO	99
7.1. CREATING A NEW JPA PROJECT	99
7.2. ADDING LIBRARIES	107
7.3. GENERATING TABLES FROM ENTITIES	110
7.4. CREATING A HIBERNATE MAPPING FILE	112
7.5. CREATING A HIBERNATE CONFIGURATION FILE	115
7.6. CREATING A HIBERNATE CONSOLE CONFIGURATION FILE	118
7.7. EDITING HIBERNATE PROJECT CONFIGURATIONS	124
CHAPTER 8. MOBILE WEB TOOLS BASICS IN CODEREADY STUDIO	130
8.1. CREATING AN HTML5 PROJECT	130
8.2. ADDING A NEW HTML5 JQUERY MOBILE FILE	132
8.3. ADDING A NEW MOBILE PAGE	136

## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

## CHAPTER 1. GIT BASICS IN CODEREADY STUDIO

CodeReady Studio includes Git Perspective, which allows developers to manage their Git repositories from a graphical interface. The following section outlines the basic workflow of a Git project in Git Perspective and describes how to accomplish the most common Git-related tasks.

## **1.1. SETTING UP GIT PERSPECTIVE**

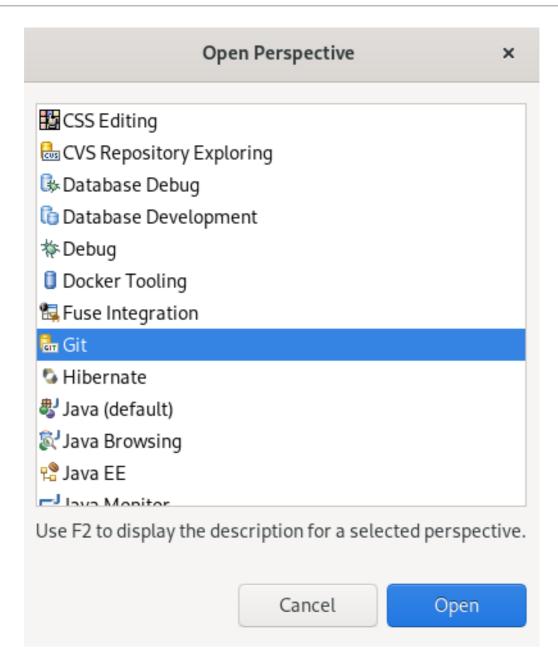
The following section describes how to open Git Perspective in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click Window → Perspective → Open Perspective → Other.

Run	window Help			
•	New Window	(	😂 🛷 🕶 🗄 👻 🖄 🕶 🕅 🕶	* * * * * * 1
	Editor			Q i 🖻 🛃
	Appearance	•		• •
	Show View	•		
	Perspective	•	📫 Open Perspective 🔹 🕨	🎋 Debug
	Navigation	•	Customize Perspective	🕵 Java Browsing
	Preferences		Save Perspective As	<u>O</u> ther
			Reset Perspective	
			Close Perspective	
			Close All Perspectives	

The **Open Perspective** window appears.



- 3. Select Git.
- 4. Click **Open**.

The **Git Repositories** view appears.



Select one of the following to add a repository to this view:

- Add an existing local Git repository
- Clone a Git repository
- Create a new local Git repository

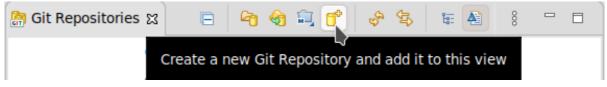
## **1.2. MANAGING REPOSITORIES IN GIT PERSPECTIVE**

#### 1.2.1. Creating a new Git repository

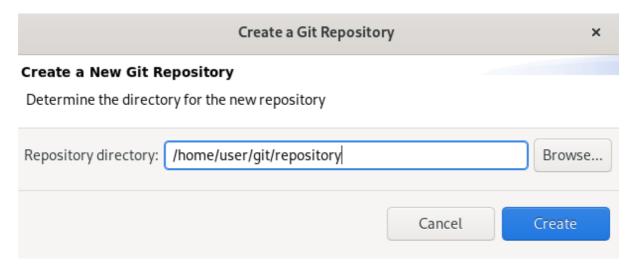
The following section describes how to use Git Perspective to create a new Git repository.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.
- 3. Click the Create a new Git Repository and add it to this view con.



The Create a Git Repository window appears.



The path to the default **Repository directory** is generated automatically. Choose the path where you want your repository to be stored at and continue with the repository creation.

Optionally, you can select the **Create as bare repository**check box.



#### NOTE

Bare repositories are recommended for central repositories, not for development environments. They do not contain a working or checked out copy of any source file. This prevents editing files and committing changes. Additionally, they store the Git revision history for your repository in the **root** folder instead of a **.git** subfolder.

#### 4. Click Create.

A new Git repository is created on your local machine and is now listed in the **Git Repositories** view.

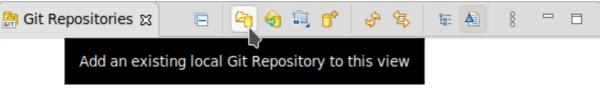
GIT	Git Repos	sitorie	es 🛙							
	E	6	କ୍ତ	î,	6*	C.	<del>(</del> 1)	ŧ.		000
۲	📋 reposi	tory [	mas	ter]	- /ho	me/u	ser/gi	it/rep	ositor	y/.git

#### 1.2.2. Adding an existing local Git repository

The following section describes how to use Git Perspective to add a local Git repository to CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.
- 3. Click the Add an existing local Git Repository to this viewicon.



The Add Git Repositories window appears.

4. Click **Browse** to locate your local Git repository.

Add Git Repositories earch and select Git repositories on your local file system			GIT
Found one repository scanning one folder in Oms			
Search criteria			
Directory: /home/user/local-repository		Browse	Search
Look for nested repositories			
Skip hidden directorie:			
earch results			
type filter text			
🗹 î /home/user/local-repository/.git			
	Cancel		Add

- 5. In the **Search results** field, select the checkbox displaying the path to the **.git** file.
- 6. Click Add.

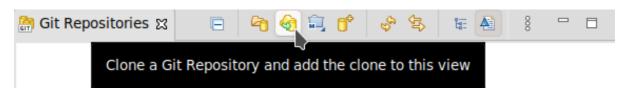
Your local repository is now listed in the **Git Repositories** view.

GIT	Git	Repo	sitorie	es 🛙							
		E	6	କ୍ତ	î,	٥¢	Ŷ	<del>(</del> 1)	te:		000
►	0	local-i	repos	itory	/ [ma	aster	] - /ho	me/	user/l	ocal-r	eposit
۲	0	reposi	itory [	mas	ter]	- /ho	me/u	ser/g	jit/rep	osito	ry/.git

#### 1.2.3. Cloning an existing Git repository

The following section describes how to use Git Perspective to create a local clone of a repository that already exists online (GitHub, GitLab).

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.
- 3. Click the Clone a Git Repository and add the clone to this view con.



The Clone Git Repository window appears.

Clone Git Repository	×

#### Source Git Repository

Enter the location of the source repository.

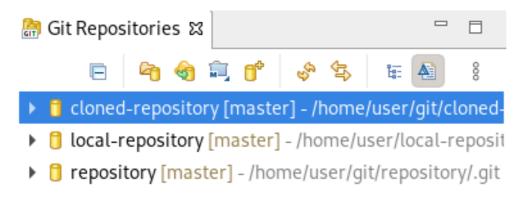
Location				
URI:	https://github.com	/user/cloned-rep	ository.git	Local File
Host:	github.com			
Repository path:	/user/cloned-repo	sitory.git		
Connection				
Protocol: https	•			
Port:				
Authentication				
User:				
Password:				
Store in Secur	e Store			
?	< Back	Next >	Cancel	Finish

- Add the address for the source repository to the URI field. The Host and Repository path fields are populated automatically.
- 5. Click Next.
- 6. Select the branches you want to clone.
- 7. Click Next.
- 8. Ensure that the **Directory** path and **Initial branch** are set correctly.

Brows
r clone finishes
New
▼ Select

#### 9. Click Finish.

Your cloned repository is now listed in the **Git Repositories** view of CodeReady Studio.



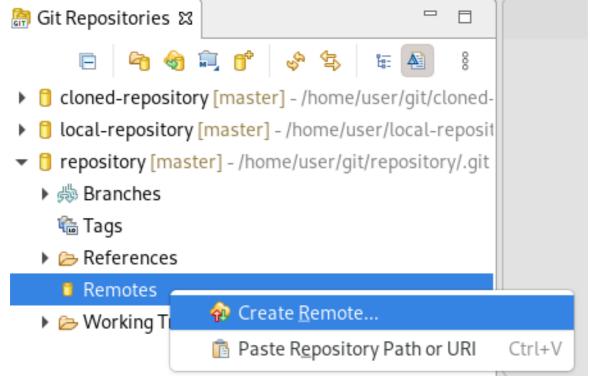
#### 1.2.4. Adding a remote for the repository

After setting up your repository in Git Perspective for the first time, add a remote for the repository. This is a one-time set up step for newly created or added repositories.

The following section describes how to use Git Perspective to set up the remote for your repository.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.
- 3. Expand your repository.



 Right-click Remotes → Create Remote. The New Remote window appears.

	New Remote	×
Please enter	a name for the new remote	
	nfigure the new remote for either fetch or push; you can add for the other direction later	
Remote name:	origin	
O Configure p ○ Configure f		
?	Cancel	Create

- 5. Name your remote.
- 6. Ensure that **Configure push** is selected.
- 7. Click Create.

The **Configure Push** window appears.

Configure Push	×
Configure push for remote 'origin'	
😣 Please provide at least one URI	ľ
JRI:	Change Remov
Push URIs	
Ref mappings	
No Push Refspec, will push currently checked out branch instead.	Add
	Modify
	Delete
	Сору
	Paste
	Advanced
Save Dry-Run Revert Cancel	Save and Push

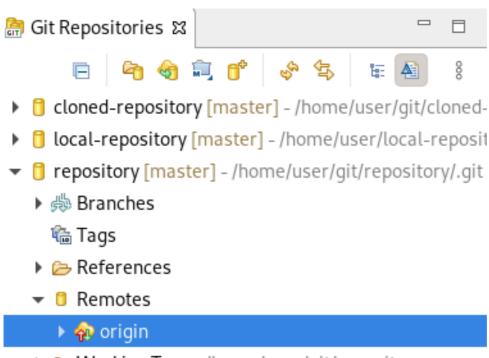
8. Click Change.

The **Select a URI** window appears.

ource Git Repo	sitory	G
Enter the location	of the source repository.	2
Location		
URI:	https://github.com/user	Local File
Host:	github.com	
Repository path:	/user	
Connection		
Protocol: https	•	
Port:		
Authentication		
User:	user	
	•••••	
Password:		
Password:	e Store	

- 9. Add the URI, username and password for the source repository. The **Host** and **Repository** path fields are populated automatically.
- 10. Click Finish.
- 11. Click Save.

Your newly added remote is now listed in the **Git Repositories** view in CodeReady Studio.



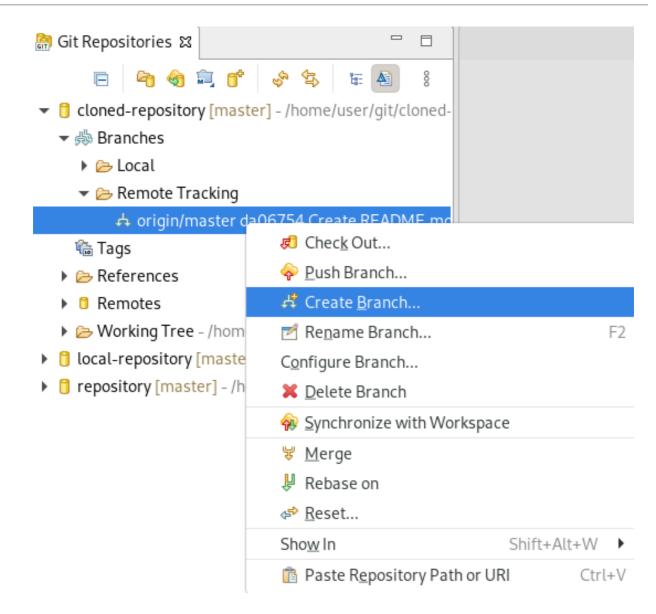
Working Tree - /home/user/git/repository

## **1.3. MANAGING BRANCHES IN GIT PERSPECTIVE**

#### 1.3.1. Creating a new branch

The following section describes how to use Git Perspective to create a new branch.

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.
- 3. Expand your repository.
- 4. Under **branches** → **Remote Tracking**, right-click **master** → **Create Branch**.

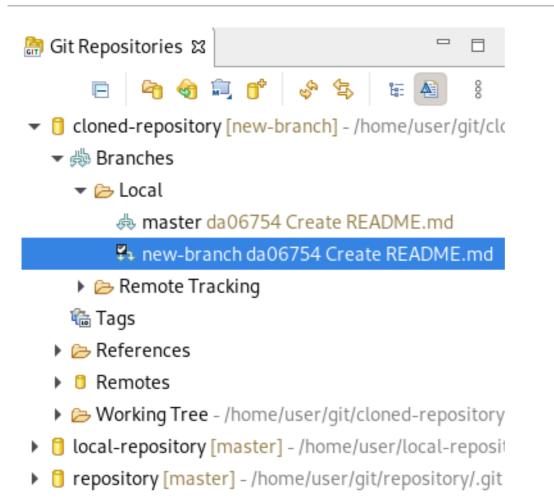


The Create Branch window appears.

	Create Branch									
Create a new l	Create a new branch in repository cloned-repository									
(i) Local branch a	s upstream is not recommended, u	se remote branch								
Source: 🖑	master		Select							
Branch name:	new-branch									
🔽 Configure up	stream for push and pull									
When pullin	g: Merge	•								
🗹 Check out ne	w branch									
(?)		Cancel	Finish							
0										

- 5. Click **Select** to pick the source of the new branch.
- 6. Name your branch.
- 7. Select the **Configure upstream for push and pull**and **Checkout new branch** check boxes.
- 8. Select an option in the **When pulling** field.
- 9. Click Finish.

Your newly added branch is now listed in the **Git Repositories** view under **branches**  $\rightarrow$  **Local** in CodeReady Studio.



#### 1.3.2. Working in the branch

The following section describes how to open a built-in terminal in Git Perspective so you can work on the created branch.

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.
- Press Shift+Ctrl+Alt+T. The Launch Terminal window appears.

Launch Terminal ×								
Choose terminal: Local Terminal 🔹								
Encoding: Default (ISO-8859-1) 🔹								
Cancel OK								

- 4. Choose Local Terminal.
- 5. Set Encoding to Default (ISO-8859-1).
- 6. Click OK.

The **Terminal** window now displays the command-line terminal.

🗐 History 🖆 Synchronize 🛃 Git Staging 🔺 Git Reflog	<b>.</b>	Tei	rm	inal	X				
		R	1	₫ <sub>Γ</sub>		×		Ē	۹
🗐 localhost.localdomain 🛱									

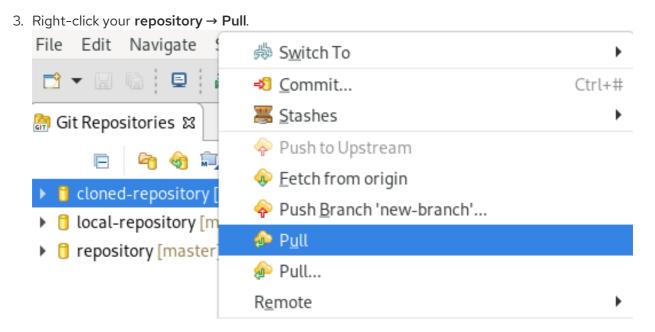
Note that by default the current working directory is the home directory of your current user.

#### 1.3.3. Updating your local repository

To avoid merge conflicts, update your local repository before merging your changes, especially when working in a shared repository.

To update your local repository, pull changes from the remote repository and merge them into your local repository.

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.



The Pull Results window appears.

Pull Result for cloned-repository	×
etch Result	
▼ 🖑 master : origin/master [ae470136ce9eb8] (2)	E
🕨 🖓 ae47013: Merge pull request #2 from Levi-Le/user-patch-1 (Levi-Le on 2020-05-20 18:10:0	Ŧ
🕨 🍇 f8c559a: Create fixes-to-v3.6 (Levi-Le on 2020-05-20 18:08:54, committed by GitHub)	
odate Result	
Result Fast-forward	
lew HEAD Merge pull request #2 from Levi-Le/user-patch-1 [ae47013]	
1erge input	
6ce9eb8: Merge pull request #1 from Levi-Le/fixes-3.2 (Levi-Le on 2020-05-20 18:06:14, committee	ed b
	mitt

#### 4. Click Close.

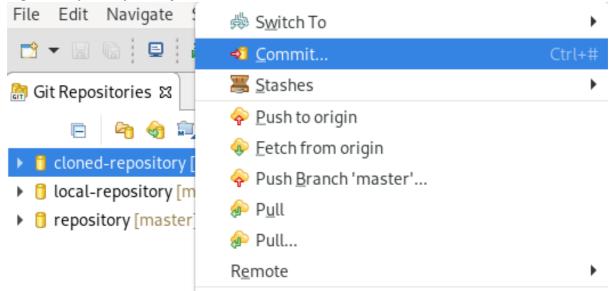
Now the changes from the remote repository are merged into your local repository.

## **1.4. COMMITTING AND PUSHING CHANGES**

The following section describes how to commit and push changes in CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.

3. Right-click your **repository** → **Commit**.



The **Git Staging** view appears.

	eclipse-worksp	ace - Eclipse IDE			-	• ×
File	Edit Navigate <u>S</u> e <u>a</u> rch Project <u>R</u> un Window Help					
«	Section 2 In the section of the	∽ on:	~ 🏟 🖬	• 🛛 🖬 🕴	0 🖉 •	·
<b>*</b> •	<b>○ - 4 - ○</b> <i>🖉 - 8</i> 🗄 - 🗄 - 🗄 - ↔ ↔ ↔	- 🖻 📥 👘			٩	<b>P</b>
	🎒 History 🖆 Synchronize 🏝 Git Staging 🛱 🔊 Git Reflog 🔲 Proper	ties				5 5
<u>61</u>			<b>Q</b> Filter files	\$ \$ D	- 6	8
	<pre>[] &gt; test-repo [master]</pre>					
	Unstaged Changes (1)	Commit Message	1		7 6 5	
	Add selected files to the index	Configuring mave	en dep			
	Staged Changes (1) — 📮 🖬 📄					
		Author: roo	ot <root@localhost></root@localhost>			
		💏 Committer: roo	ot <root@localhost></root@localhost>			
			<pre>       Commit and Push     </pre>		ommit	
		125M of 256M				0

- 4. Select the changes you want to stage.
- 5. Click the Add selected files to the indexicon to stage the changes.
- Add a commit message to the Commit Message field.
   Author and Committer fields are populated automatically.
- 7. Click **Commit** to commit your changes, or **Commit and Push** to commit your changes and push them to the remote repository.

Note that when selecting the **Commit and Push** option you are prompted to enter the repository address, your access username, and password for the repository.

## **CHAPTER 2. MAVEN BASICS IN CODEREADY STUDIO**

Maven provides a standardized build system for application development, and facilitates fetching dependencies from one or more repositories.

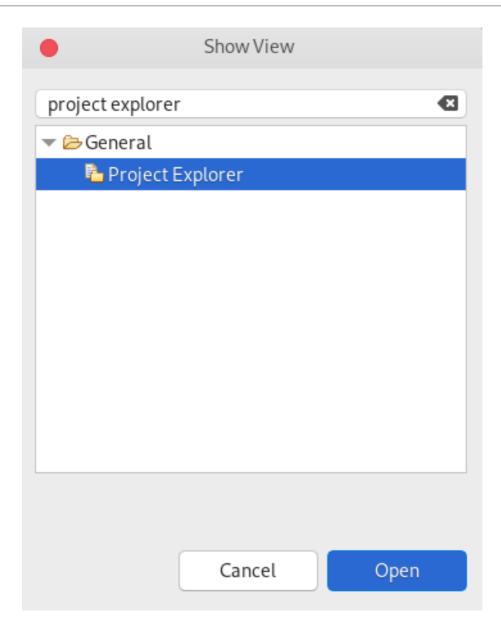
Root Maven projects can serve as aggregators for multiple Maven modules (sub-projects). For each module that is part of a maven project, a <module> entry is added to the project's **pom.xml** file. A **pom.xml** contains <module> entries and is often referred to as an **aggregator pom**.

When modules are included into a project it is possible to execute Maven goals across all modules by a single command issued from the parent project directory.

## 2.1. CREATING A NEW MAVEN PROJECT

The following section describes how to create a new Maven project in CodeReady Studio.

- 1. Start CodeReady Studio.
- Click Window → Show View → Other. The Show View window appears.



- 3. Enter Project Explorer in the search field.
- 4. Select Project Explorer.
- 5. Click **Open**.

The Project Explorer view appears.

눰 Project Explorer 🛿				
	F	€£ĵ	7	000
🕨 🗁 org.company-nai	me_	proj	ect-n	iame

6. Press Ctrl+N.

The **Select a wizard** window appears.

		New		×
Select a wizard				
Create a Maven proje	ct			
Wizards:				
maven				☑
🔻 🗁 JBoss Tools				
🐸 Configure M	aven Repositorie	S		
🕶 🗁 Maven				
🗐 Check out M	aven Projects fro	m SCM		
🖄 Maven Modu	le			
🔛 Maven Proje	ct			
🔻 🗁 Red Hat Fuse				
M Fuse Integra				
Show All Wizards				
?	< Back	Next >	Cancel	Finish

- 7. Enter Maven in the Wizards field.
- 8. Select Maven Project.
- 9. Click Next.

The New Maven Project window appears.

	New Maven Project	
New Maven project		
Select project name an	dlocation	141
Create a simple pro	ject (skip archetype selection)	
🗹 Use default <u>W</u> orksp	acelocation	
Location:		- Brows <u>e</u>
	rking set	
Working set:		▼   Mor <u>e</u>
		▼ Mor <u>e</u>
Working set:		▼ Mor <u>e</u>

10. Select the **Create a simple project** check box.



#### NOTE

By selecting the **Create a simple project**check box you are skipping the archetype selection and the project type is automatically set to Project Object Model (POM), which is a requirement for multi-module Maven projects.

To create a standalone Maven project instead, clear the **Create a simple project**check box and follow the onscreen instructions to set the packaging option to **jar** or **war**.

- 11. Click **Browse** to select the workspace location.
- 12. Click Next.

New Maven	project					-
Configure pr	oject					V
Artifact						
Group Id:	org.company-nar	ne_p	roject-name			
Artifact Id:	org.company-nar	ne_p	roject-name			
Version:	0.0.1-SNAPSHOT	•				
Packaging:	pom	•				
Name:						
Description:						
Parent Projec	:t					
Group Id:						
Artifact Id:						
Version:		•			Browse	Clea
Advanced						

13. Enter the group ID and the artifact ID.



#### NOTE

The values for the IDs cannot include spaces or special characters. The only special characters allowed are periods (.), underscores (\_), and dashes (-). An example of a typical group ID or artifact ID is **org.company-name\_project-name**.

Optionally, you can name your project and add a description.

- 14. Set Packaging to pom, jar or war.
- 15. Click Finish.

Your newly created Maven project is now listed in the Project Explorer view.

#### 2.2. IMPORTING EXISTING MAVEN PROJECTS

The following section describes how to import existing Maven projects into CodeReady Studio.

#### 2.2.1. Importing an existing locally stored Maven project

The following section describes how to import an existing locally stored Maven project into CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click File  $\rightarrow$  Import. Navigate Search Project File Edit Run Window Shift+Alt+N New ۲ Open File... Open Projects from File System... Recent Files ۲ Close Editor Ctrl+W Close All Editors Shift+Ctrl+W Save Ctrl+S Save As... 🐚 Save All Shift+Ctrl+S Revert Move... Marken Rename... F2 Refresh F5 Convert Line Delimiters To ۲ ៉ Print... Ctrl+P 🖮 Import... 🖆 Export...

The Import window appears.

			Import		×
elect					
mport existi	ng Maven proje	ects			
Select an imp	oort wizard:				
maven					×
🕶 🗁 Maven					
	ck out Maven f	-	n SCM		
	ting Maven Pr				
			a Maven reposi	tory	
	erialize Maver		-		
Mat Mat	erialize Maver	n Projects fr	om SCM		
0		Pack	Novt S -	Cancel	Finish
•		Back	Next >	Cancel	Finish

- 3. Enter Maven in the Select an import wizard field.
- 4. Select Existing Maven Projects
- 5. Click Next.

The Import Maven Project window appears.

	Import Maven Projects	
Maven Project		
Select Maven pr	ojects	
<u>R</u> oot Directory:	/home/user/maven-project	<u>B</u> rowse
<u>P</u> rojects:		
	l com.example.maven-project:maven-project:1.0-SNAPSHOT: /pom.xml com.example.maven-project:server:1.0-SNAPSHOT	Select <u>A</u> ll
server	Deselect A	
		Select Tree
		Deselect Tre
		<u>R</u> efresh
🗹 Add project(	s) to working set	
	ect	
maven-proj		
maven-proj		

- 6. Click **Browse** to locate your Maven project.
- 7. Select the Add project(s) to working setcheck box.
- 8. Click Finish.

Your local Maven project is now listed in the Project Explorer view.

### 2.2.2. Importing an existing remotely stored Maven project

The following section describes how to import an existing remotely stored Maven project into CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. Open Git Perspective.

З.	Click the Clone a Git repository and import existing Maven projectsicon.	
	🔚 Git Repositories 🕱 📄 🖻 🔄 😭 😭 😭 😭 😫 😫 😑 🗖	]
	jboss-eap-quickClone a Git repository and import existing Maven projects arts/.git	
	The Check out as Maven project from SCM window appears.	
	Check out as Maven project from SCM ×	
	Target Location	
	Select target location and revision	
	SCM URL: git   https://github.com/user/maven-project.git	
	Check out Head Revision	
	Revision:	
	Check out All Projects	
	<ul> <li>Advanced</li> </ul>	
	Find more SCM connectors in the m2e Marketplace	e
	(?) < Back Next > Cancel Finish	
	(?) < Back Next > Cancel Finish	

- 4. Add the address for the source repository to the **SCM URL** field.
- 5. Click Next.

The Select Project Location window appears.

	Check out as Maven project from SCM		×
Select Project I	Location		
Select project loc	ation and working set		
🕑 Use default W	/orkspace location		
Location:		-	Browse
Add project(s)	to working set		
Working set:		-	More
Advanced			
~		el	

- 6. Click **Browse** to select the workspace location.
- 7. Click Finish.

Your remote Maven project is now listed in the **Git Repositories** view.

#### 2.3. CREATING A NEW MAVEN MODULE

The following section describes how to create a new Maven module.

#### Prerequisites

An existing Maven project.
 For more information on how to create a Maven project, see Section 2.1, "Creating a new Maven project".

- 1. Start CodeReady Studio.
- Press Ctrl+N.
   The Select a wizard window appears.

		New		×
Select a wizard				-
Create a Maven mod	ule			
Wizards:				
maven				Ø
Configure M	aven kepositories	5		
🕶 🗁 Maven				
詞 Check out M	aven Projects fror	n SCM		
👺 Maven Modu	ıle			
🖄 Maven Proje	ct			
🔻 🗁 Red Hat Fuse				
😁 Fuse Integra	tion Project			
😁 Fuse Online	Extension Project			
Show All Wizards				
(?)	< Back	Next >	Cancel	Finish
J	Dack	NCAU	Cancer	THIST

- 3. Enter Maven in the Wizards field.
- 4. Select Maven Module.
- 5. Click Next.

The New Maven Module window appears.

	New Maven Module	×
New Maven Me	odule	-
Select the modu	le name and parent	M
✓ Create a sim	ple project (skip archetype selection)	
<u>M</u> odule Name:	new-maven-module	•
Parent Project:	maven-project	Br <u>o</u> wse
<u>A</u> dd project(	;) to working set	
Working set:	▼	Mor <u>e</u>
▶ Advanced		
(?)	< Back Next > Cancel	Finish

6. Select the **Create a simple project**check box.



#### NOTE

By selecting the **Create a simple project**check box you are skipping the archetype selection and the project type is automatically set to Project Object Model (POM), which is a requirement for multi-module Maven projects.

To create a standalone Maven project instead, clear the **Create a simple project**check box and follow the onscreen instructions to set the packaging option to **jar** or **war**.

- 7. Name your module.
- 8. Click **Browse** to select the parent project.
- 9. Click Next.

The **Configure Project** window appears.

Configure pr	oject				
Artifact					
Group Id:	com.example.ma	aven-p	roject		
Artifact Id:	new-maven-mod	dule			
Version:	1.0-SNAPSHOT	•			
Packaging:	pom	•			
Name:					
Description:					
Parent Projec	:t				
Group Id:	com.example.ma	aven-p	roject		
Artifact Id:	maven-project				
Version:	1.0-SNAPSHOT	•			
Advanced					

#### 10. Set Packaging to pom, jar or war.

Optionally, you can name your module and add a description.

11. Click Finish.

Your newly created Maven module is now listed below your Maven project.

### 2.4. ADDING A MAVEN DEPENDENCY TO A MAVEN PROJECT

The following section describes how to add a Maven dependency to a Maven project in CodeReady Studio.

#### Prerequisites

An existing Maven project.
 For more information on how to create a Maven project, see Section 2.1, "Creating a new Maven project".

#### Procedure

1. Start CodeReady Studio.

- 2. Open Project Explorer.
- 3. Right-click your Maven **project** → **Maven** → **Add Dependency**.

			pace - Eclipse IDE
	New	•	·
File Edit Navigate	Go Into		
📑 🖛 🖪 🗋 📃	Show In Shi	ft+Alt+W 🕨	<ul> <li>▲ 1</li> <li>▲ 1</li> <li>★ 4</li> <li>★ 4</li> <li>★ 4</li> <li>★ 4</li> </ul>
눱 Project Explorer 🕱	Show in Local Terminal	•	
> maven-project	🗈 Сору	Ctrl+C	
<ul> <li>Image: Second project</li> <li>Ima</li></ul>	💼 Paste	Ctrl+V	
5 1 /	🗙 Delete		
	Move		
	Rename	F2	
	🔤 Import		
	🖆 Export		
	Refresh	F5	
	Close Project		
	Close Unrelated Projects		
	🕮 Mark as Deployable		
	Show in Remote Systems view		
	💁 Coverage As	•	Add <u>D</u> ependency
	🜔 Run As	•	Add <u>P</u> lugin
	🕸 Debug As	•	👺 New <u>M</u> aven Module Project
	Profile As	•	Download Javadoc
	Restore from Local History		Download <u>S</u> ources
	Maven	•	<b>֎ầ Update Project</b> Alt+F5
	Team	+	Select Maven Profiles Ctrl+Alt+P
	Compare With	•	Disable Workspace <u>R</u> esolution
	Replace With	•	Disable Maven <u>N</u> ature

The Add Dependency window appears.

	Add Dependenc	Ŷ	×
Group Id: *	org.company-name_project		
	org.company-name_project	)	
Version:	0.0.1-SNAPSHOT	Scope:	compile 👻
Enter groupl	d, artifactId or sha1 prefix or p	attern (*):	
org.compa	any-name_project-name		
🖲 Index do	wnloads are disabled, search r	esults ma	y be incomplete.
Search Resu	lts:		
🕨 🗋 org.co	mpany-name_project-name	org.compa	any-name_project
?	Ca	ancel	ОК

4. Enter the group ID or the artifact ID in the **Enter groupId**, artifactId or sha1 prefix or pattern field.

The fields above are populated automatically.

5. Click **OK**.

The dependency is now added to the **pom.xml** file of your project.

# 2.5. ADDING MAVEN SUPPORT TO AN EXISTING NON-MAVEN PROJECT

The following section describes how to add Maven support to an application created without Maven support.

1. Start CodeReady Studio.

2. Open Project Explorer.

	New	•	- Eclipse IDE
File Edit Navigate Sear	Go Into		
📑 🛨 🔚 🕼 🖳 🗮 🕶	Show In Shift+Alt+	• W	③ ▼
•	🗎 Сору	Ctrl+C	Q 🖻 🖏
陷 Project Explorer នេ	🗎 Copy Qualified Name		으 🖬 Task List 🟻
E to jest Explorer to E	🛅 Paste	Ctrl+V	
	🔀 Delete		
<ul> <li>B &gt; maven-project [mave</li> <li>D org.company-name_pr</li> </ul>	Build Path	•	8
✓ dig.company-name_pi ✓ dig.company-name_pi ✓ dig.company-name_pi	Refactor Shift+Alt	+T 🕨	Find +
▶ ■ JRE System Library [	🔤 Import		
🕨 🖶 doc	🖆 Export		Add Seam support
Handbox doc.javadoc_tags	🗞 Refresh	F5	Add JSF Capabilities
# > javadoc_tags	Close Project		Add CDI (Contexts and Dependency Injection) support
📑 LICENSE	Close Unrelated Projects		Convert to Faceted Form
🕞 README.md	🖺 Mark as Deployable		Configure and Detect Nested Projects
	Show in Remote Systems view		Convert to JPA Project
	💁 Coverage As	•	Add Gradle Nature
	💿 Run As	+	Add Project Archives Support
	☆ Debug As	•	Convert to Plug-in Projects
	Profile As	•	Add JAX-RS Support
	Restore from Local History		Add JBoss Tools Knowledge Base Support
	Team	•	🦙 Add Fuse Integration Support
	Compare With	+	Convert to <u>J</u> avaScript Project
	Replace With	•	Enable preview features
	Configure		Create module-info.java
😂 sample-java-project	~		Convert to Maven Project

3. Right-click your **project** → **Configure** → **Convert to Maven Project** 

The **Create a new POM** window appears.

	Create new POM ×					
Maven POM						
This wizard cr	eates a new POM (pom.xml) descriptor for Maven.					
Project: /sar Artifact	nple-java-project					
Group Id:	sample-java-project 🗸					
Artifact Id:	sample-java-project 🔹					
Version:	0.0.1-SNAPSHOT -					

Packaging:	jar 🗸	
Name:	•	
Description:		
?	Cancel	

All fields are populated automatically. If you want to change the group ID or the artifact ID, note that the values cannot include spaces or special characters. The only special characters allowed are periods (.), underscores (\_), and dashes (-). An example of a typical group ID or artifact ID is org.company-name\_project-name.

#### 4. Click Finish.

Your newly generated **pom.xml** file appears under your Java project.

## 2.6. ADDITIONAL RESOURCES

• For more information on how to use the Maven software project management and comprehension tool, see the JBoss Community Archive.

# CHAPTER 3. APPLICATION DEPLOYMENT IN CODEREADY STUDIO

In order to deploy applications to a server from within CodeReady Studio you must configure the IDE with information about the server. For a local server this information includes the following:

- A server runtime environment with details about the server location, runtime JRE, and configuration files
- A server adapter with management settings for the server runtime environment, including access parameters, launch arguments, and publishing options

JBoss Server Tools enables you to efficiently configure a local server ready for use with CodeReady Studio using Runtime Detection. This feature is useful for quickly configuring a server for deploying and testing an application.

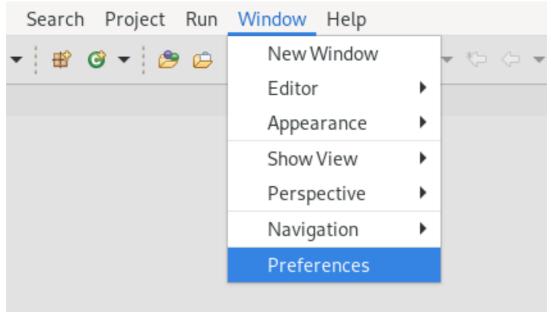
# **3.1. CONFIGURING A LOCAL SERVER**

Runtime Detection searches a given local system path to locate certain types of runtime servers. For any servers found, Runtime Detection automatically generates both a default server runtime environment and a default server adapter. These items can be used for immediate application deployment as is or they can be customized to meet your requirements.

The following section describes how to configure a local server in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click **Window** → **Preferences**.



The **Preferences** window appears.

	Preferences		
jboss 💌	JBoss Runtime Detection		⇔ • ⇒ •
JBoss Tools JBoss Maven Integration JBoss Runtime Detection Web Services	Description Each path on this list will be autom a new workspace is created or if se Click Edit to configure rules/filters	elected at every Ecli	
JBossWS Preferences	Paths		
	Scan Path On Every start		Add
	<ul> <li>✓ ⊗ /home/user/jboss-runtimes</li> <li>✓ ⊗ /home/user/.minishift</li> </ul>		Edit
			Remove
			Search
			Download
	Available runtime detectors		
	Туре		Link
	JBoss AS		Link
	🗹 Seam		<u>Link</u>
	Tomcat		<u>Link</u>
	Red Hat Fuse		
	🗹 Apache Karaf		
		Restore Defaults	Apply
		Restore Defaults	Арр

- 3. Enter **JBoss** in the search field.
- 4. Select JBoss Runtime Detection
- 5. Click **Add**.
- 6. Locate the directory containing the runtime server.
- Click **Open**. The **Searching for runtimes** window appears.

Searching for runtimes				
No new runtime found. Searching runtimes is finished				
Name	Туре	Version	Errors / QuickFix	Location
Red Hat JBoss EAP 7.2	EAP	7.2		/home/user/jboss-eap-
<ul> <li>Hide already created runtim</li> <li>Add runtime path to list of p</li> </ul>	aths search		·	
	OK	S	elect All Desele	ct All Cancel

- 8. Click **OK**.
- 9. Select the path to the runtime server directory.

	Preferences		×
jboss 💌	JBoss Runtime Detection		<
<ul> <li>JBoss Tools         JBoss Maven Integ         JBoss Runtime Det         Web Services     </li> </ul>	Description Each path on this list will be automa a new workspace is created or if sel Click Edit to configure rules/filters f	lected at every Ec	
JBossWS Preferen	Paths		
	Scan Path On Every start		Add
	/home/user/jboss-eap-7.3		Edit
	✓ Ø /home/user/jboss-runtimes		Euit
	V lome/user/.minishift		Remove
	/home/user/jboss-eap-7.2		Search
			Download
	Available runtime detectors		<u></u>
	Туре		Link
	JBoss AS		Link
	🗹 Seam		<u>Link</u>
	Tomcat		<u>Link</u>
	Red Hat Fuse		
	🗹 Apache Karaf		
		Restore Defaults	Apply
? 占 🖌 🔘		Cancel	Apply and Close

10. Click Apply and Close.

# **3.2. CONFIGURING A REMOTE SERVER**

The following section describes how to configure a remote server in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Press Ctrl+N.

The **Select a wizard** window appears.

	New		×
Select a wizard			-
Define a new serve	er		
Wizards:			
server			Ø
🕶 🗁 Server			
🚏 Server			
🗁 Tern			
Show All Wizar	ds.		
?	< Back Next >	Cancel	Finish
Enter <b>Server</b> in the s	search field.		
Select <b>Server</b> .			
Click <b>Next</b> . The <b>Define a New S</b>	Server window appears.		
	New Server		×
Define a New Se	rver		
Choose the type of	f server to create		
Select the <u>s</u> erver ty	ype:		
type filter text			
🥯 Red Hat J	Boss Enterprise Application Platform	7.2	ſ
🤜 Red Hat J	Boss Enterprise Application Platform	7.3	
Red Hat JBoss Ente	erprise Application Platform (EAP) 7.2		
	0		
Server's host name	e: localhost		
Server na <u>m</u> e:	Red Hat JBoss EAP 7.2		

- 6. Select a server type.
- 7. Click Next.

The Create a new Server Adapter window appears.

	r	New Server		×
Create a new Serve Red Hat JBoss Enterpr		tform (EAP) 7.2		RED HAT' JBOSS' MIDDLEWARE
Controlled by: O Fil	nts and keeps trac cal emote lesystem and shell anagement Opera externally manage oes not require a re vithout one. o this server	k of which modu operations tions d.	les have been deplo	byed.
?	< Back	Next >	Cancel	Finish

- 8. Select the **Remote** check box.
- 9. Select a **Controlled by** option.
- 10. Select the **Server lifecycle externally managed** check box.
- 11. Select the **Assign a runtime to the server**check box.
- 12. Click Next.

The **JBoss Runtime** window appears.

		New Server				×
JBoss Runtime						RED HAT JBOSS
Red Hat JBoss Enterprise	Application Platform (E	AP) 7.2				MIDDLEWAR
	sspaths for projects wh	ation directory. ich depend on this runtime, and stop instances of JBoss.				
Name						
JBoss EAP 7.2 Runtime						
Home Directory					Download and ir	nstall runtime
/var/home/user/jboss-e	eap-7.2					Browse
Runtime JRE						
O Execution Environme	ent:	JavaSE-1.8	•		Environments	
O Alternate JRE:		java-1.8.0-openjdk-1.8.0.252.b09-1.1	fc32.x81 👻		Installed JREs	
Server base directory:	standalone					Browse
Configuration file: star	idalone.xml					Browse
?			< Back	Next >	Cancel	Finish

13. Click **Browse** in the **Home Directory** field to locate the runtime server.

#### 14. Click Next.

The **Remote System Integration** window appears.

New Server	×
onnecting to a remote system.	<b>BOSS</b> by Red Hat
	▼ New Host
<u>Ор</u>	en Remote System Explorer View
user/jboss-eap-7.2	Browse
ndalone	Browse
standalone.xml	
	onnecting to a remote system. Op /user/jboss-eap-7.2 ndalone

15. Click **Browse** in the **Remote Server Home** field. The **Browse remote system** window appears.

Browser	emote system	×
Select a file		
Connection: Local		▼ New
/home/user/jboss-eap-7.2		
🕶 🚔 My Home		
🕨 🗀 Desktop		
Documents		
Downloads		
eclipse		
eclipse-workspace		
Contractions and the second		
eclipse-workspace	2	
<ul> <li>igit</li> <li>iboss-eap-7.2</li> </ul>		
<ul> <li>jboss-eap-7.3</li> </ul>		
Music		
	Cancel	ОК

- 16. Specify the path to the directory that contains the remote server.
- 17. Click Finish.

## **3.3. DEPLOYING AN APPLICATION**

After configuring the local server, you can deploy applications to the server from CodeReady Studio using the server adapter. The server adapter enables runtime communication between the server and CodeReady Studio for easy deployment of applications and server management.

The following section describes how to deploy an application to the server in CodeReady Studio.

#### Procedure

1. Start CodeReady Studio.

2.	Right-click your	project $\rightarrow$ Run as $\rightarrow$ Run on Server	•
----	------------------	--	---

A L			
File Edit Navigate Se Gol		'	
	Into	:	
🖻 🔻 🗟 📮 📮 Sho	win Shift+Alt+W		
	Copy Ctrl+	-C	
ြဲ Project Explorer 🕱 👘 🗎	Copy Qualified Name		
	Paste Ctrl+	.V	
	Delete		
	ld Path	•	
kontent-sample-ap Reference	actor Shift+Alt+T	•	
<ul> <li>b &gt; maven-project [ma</li> <li>b &gt; org.company-name_</li> </ul>	port	•	
<ul> <li>Exp</li> <li>Sample-java-proje</li> </ul>	port	•	
	Refresh	-5	
Clos	se Project		
Clos	se Unrelated Projects		
<u>E</u>	Mark as Deployable		
Sho	ow in Remote Systems view		
Q_ (	Coverage As	•	📕 1 Run on Server
	Run As	•	2 Java Application
*	Debug As	•	m2 <u>3</u> Maven build
Pro	file As	•	m² 4 Maven build
Res	store from Local History		m² <u>5</u> Maven clean
Java	a EE Tools	•	m² <u>6</u> Maven clean verify
Mai	ven	•	m2 7 Maven generate-sources
1*10			5
Tea	im	•	m2 8 Maven install
Tea	ım npare With	•	m² <u>8</u> Maven install m² <u>9</u> Maven test

The **Run on Server** window appears.

		Run On Server		:
Run On Server				
Select which server to	use			
How do you want to se	elect the server?			
🔾 Choose an existi	ng server			
🔘 Manually define	a new server			
Select the server that	you want to use:			
type filter text				
Server			State	
🕶 🗁 localhost				
🧠 Red Hat JBos	s EAP 7.0		🖡 Started	
🧠 Red Hat JBos	s EAP 7.3		遣 Stopped	
🔛 WildFly 18.0			🛅 Stopped	
📷 WildFly 19			遣 Stopped	
Red Hat JBoss Enterp	rise Application	Platform (EAP) 7.0		Columns
Always use this se	rver when runni	ng this project		
?	< Back	Next >	Cancel	Finish

- 3. Select the **Choose an existing server** check box.
- 4. Select the server you want to deploy.
- 5. Click Finish.

Your application opens in the internal CodeReady Studio web browser.

# CHAPTER 4. JBOSS EAP AND JBOSS WFK BASICS IN CODEREADY STUDIO

The Eclipse IDE supports application development and deployment with Red Hat JBoss Enterprise Application Platform (JBoss EAP) and Red Hat JBoss Web Framework Kit (JBoss WFK).

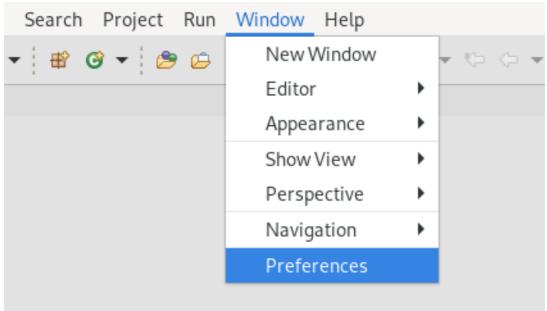
However, you need to configure Maven repositories first. This configuration is essential for using the enterprise versions of the example Maven projects provided in Red Hat Central. These projects are intended for deployment to JBoss EAP and require IDE access to JBoss EAP and JBoss WFK repositories.

## **4.1. CONFIGURING MAVEN REPOSITORIES**

The following section describes how to configure Maven repositories.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click Window → Preferences.



The **Preferences** window appears.

	Preferences ×
jboss 💌	JBoss Maven Integration $\Leftrightarrow \checkmark \Rightarrow \Rightarrow \$$
<ul> <li>JBoss Tools</li> <li>JBoss Maven Integration</li> <li>JBoss Runtime Detection</li> <li>Web Services</li> <li>JBossWS Preferences</li> </ul>	<ul> <li>When importing Maven projects configure the following:</li> <li>Seam</li> <li>Seam Runtime</li> <li>Seam Artifacts (view folder, model source folder, package)</li> <li>CDI facet</li> <li>Hibernate</li> <li>Spring Boot</li> <li>Configure Maven Repositories</li> <li>Enable 'Maven clean verify' menu</li> </ul> Restore Defaults Apply
? 눱 🖆 🔘	Cancel Apply and Close

- 3. Enter **JBoss** in the search field.
- 4. Select JBoss Maven Integration.
- Click Configure Maven Repositories.
   The Configure Maven Repositories window appears.

Configure Maven Repositorie	S	
User settings: /home/user/.m2/s Repositories	ettings.xml	
		Remove Remove All Add Repository. Edit Repository.
Preview: Old settings	New settings 1 xml version<br 2 <settings xm<br="">3</settings>	n="1.0" encoding="UTF-8 lns="http://maven.apach

#### 6. Click Add Repository.

The Add Maven Repository window appears.

	Add Maven Reposito	ry	×
Add Ma	aven Repository		M
Profile			
Profile	D: redhat-ga-repository	•	🗹 Active by default
Reposito	ry		
ID:	redhat-ga-repository		
Name:	Red Hat GA repository		
URL:	http://maven.repository.redhat.com/ga/		
<ul> <li>Adva</li> </ul>	nced		
🧲 Rec	ognize JBoss Maven Enterprise Repositories		
?		Cancel	ОК

- 7. Click the down-arrow in the **Profile ID** field.
- 8. Select the **redhat-ga-repository**. Other fields are populated automatically.
- 9. Click **OK**.
- 10. Click Finish.The Confirm File Update window appears.
- 11. Click Yes.
- 12. Click Apply and Close.

#### Additional resources

• For more information on Maven repositories, see Maven: Getting Started - Developers.

## 4.2. SETTING UP JBOSS EAP

To set up JBoss EAP in Eclipse, you must direct the IDE to the local or remote runtime servers. This establishes a communication channel between the IDE and the JBoss EAP server for efficient deployment and server management workflows.

The following section describes how to install JBoss EAP in CodeReady Studio.

#### Prerequisites

• Configured Maven repositories.

For more information on how to configure Maven repositories, see Section 4.1, "Configuring Maven repositories".

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click **Window** → **Preferences**.

Search Project Run	Window Help	
- 😰 🎯 - 😕 😝	New Window	- *> <> -
	Editor	•
	Appearance	•
	Show View	•
	Perspective	•
	Navigation	F
	Preferences	

The **Preferences** window appears.

	Preferences	>
jboss	JBoss Runtime Detection	▼
JBoss Tools JBoss Maven Integr JBoss Runtime Dete Web Services	Each path on this list will be automatically s	t every Eclipse startup.
JBossWS Preferenc	Paths Paths	
	Scan Path On Every start	Add
	<ul> <li>Image: Weighted Science</li> <l< td=""><td>Edit</td></l<></ul>	Edit
		Remove
		Search
		ſ
		Download
	Available runtime detectors	
	Туре	Link
	Type       JBoss AS	Link Link
	<ul> <li>✓ JBoss AS</li> <li>✓ Seam</li> <li>✓ Tomcat</li> </ul>	Link
	<ul> <li>JBoss AS</li> <li>Seam</li> <li>Tomcat</li> <li>Red Hat Fuse</li> </ul>	Link Link
	<ul> <li>✓ JBoss AS</li> <li>✓ Seam</li> <li>✓ Tomcat</li> </ul>	Link Link

3. Enter **JBoss** in the search field.

#### 4. Select JBoss Runtime Detection

#### 5. Click **Download**.

The **Download Runtimes** window appears.

#### **Download Runtimes**

#### **Download Runtimes**

Please select a runtime to download and install.

Name	Version
Red Hat JBoss EAP 6.0.0	6.0.0
Red Hat JBoss EAP 6.0.1	6.0.1
Red Hat JBoss EAP 6.1.0	6.1.0
Red Hat JBoss EAP 6.2.0	6.2.0
Red Hat JBoss EAP 6.3.0	6.3.0
Red Hat JBoss EAP 6.4.0	6.4.0
Red Hat JBoss EAP 7.0.0	7.0.0
Red Hat JBoss EAP 7.1.0	7.1.0
Red Hat JBoss EAP 7.2.0	7.2.0
Red Hat JBoss EAP 7.3.0	7.3.0
Selected Runtime Details	
Project URL: <u>https://developers.redhat.com/produ</u>	<u>cts/eap</u>
Download URL: https://www.jboss.org/download-mai	nager/jdf/file/jboss-eap-7.3.0.zip
Registration required. Downloads require accepting th	
Over the second seco	Cancel Finish

6. Select the JBoss EAP version you need.



#### NOTE

If you select the JBoss EAP version 6.0.x or earlier, follow the on-screen instructions. If you select a later version, follow the instructions below.

7. Click Next.

The Credentials window appears.

	Download Runtimes	×
Credentials		
Please use yo	our access.redhat.com single sign-on credentials to begin your down	load.
	your access.redhat.com credentials below. If ve an access.redhat.com account, please sign	
Domain:	access.redhat.com	•
Username:		Add
Password:		
?	< Back Next > Cancel F	inish

- 8. Click Add.
- 9. Enter your **access.redhat.com** username and password.
- 10. Click **OK**.
- 11. Click Next.

Review the license agreement, if satisfied, accept the license and click **Next** to continue with the installation.

The **Download Runtimes** window appears.

**Download Runtimes** 

×

#### Download Runtime

URL:	https://www.jboss.org/download-manager/jdf/file/jboss-eap-7.0.0.zip						
Install folder:	/home/user Browse						
Download folder:	/tmp Browse						
✓ Delete archive after installing							
?	< Back	Next >	Cancel	Finish			

- 12. Click **Browse** to select the **Install folder**.
- 13. Click **Browse** to select the **Download folder**.

#### 14. Click Finish.

Note that downloading and installing the Runtime might take a while.

The JBoss Runtime Detection window appears.

Preferences ×					
jboss <b>IJBoss Runtime Detection</b>					
<ul> <li>JBoss Tools         JBoss Maven Integration         JBoss Runtime Detection     </li> <li>Web Services</li> </ul>	Description Each path on this list will be automatically scanned for runtimes a new workspace is created or if selected at every Eclipse startu Click Edit to configure rules/filters for the search.				
JBossWS Preferences	Paths				
	Scan Path On Every start	Add			
	<ul> <li>/home/user/jboss-eap-7.3</li> <li>% /home/user/jboss-runtimes</li> </ul>	Edit			
	✓ Ø /home/user/.minishift	Remove			
		Search			
		Download			
	Available runtime detectors				
	Туре	Link			
	Source State	Link			
	✓ Seam	Link			
	Tomcat	Link			
	Red Hat Fuse				
	Apache Karaf				
	Restore Default	Apply			
? 🔁 🚄 🔘	Cancel	Apply and Close			

- 15. Select the path to the JBoss EAP installation file check box.
- 16. Click Apply and Close.

#### Verification steps

1. Click Window → Show View → Other.

Run	Window Help		
- 0	New Window		🖶 🞯 🕶 😂 😂 🛷 👻 🗄
	Editor	►	
	Appearance	►	
	Show View	Þ	条 Ant
	Perspective	•	📮 Console
	Navigation	►	😫 Declaration
	Preferences		🕙 Error Log
			@ Javadoc
			🕾 Navigator (Deprecated)
			📴 Outline
			🛱 Package Explorer
			🖹 Problems
			nogress
			陷 Project Explorer
			🛷 Search
			🗐 Task List
			🧟 Tasks
			📔 Templates
			😫 Type Hierarchy
			<u>O</u> ther

The **Show View** window appears.

	Show View	×
servers		
👭 Servers		
(	Cancel	Open

- 2. Type **Servers** in the search field.
- 3. Select Servers.
- 4. Click **Open**.

The **Servers** view appears.

Your newly added JBoss EAP is now listed in the **Servers** view.

🖹 Problems @ Javadoc 😣 Declaration 📮 Console 👫 Servers 🛱	🖻 🎋 🚺 🌮 🗏 🖺	80 □ □
✓ Sed Hat JBoss EAP 7.3 [Stopped]		
XML Configuration		
🕨 🗁 Server Details		
🕨 🚉 Filesets		
JMX[Disconnected]		

# **CHAPTER 5. OPENSHIFT BASICS IN CODEREADY STUDIO**

CodeReady Studio includes the OpenShift Application Explorer view, which provides a simplified user experience allowing easy and rapid feedback through the inner loop as well as debugging.

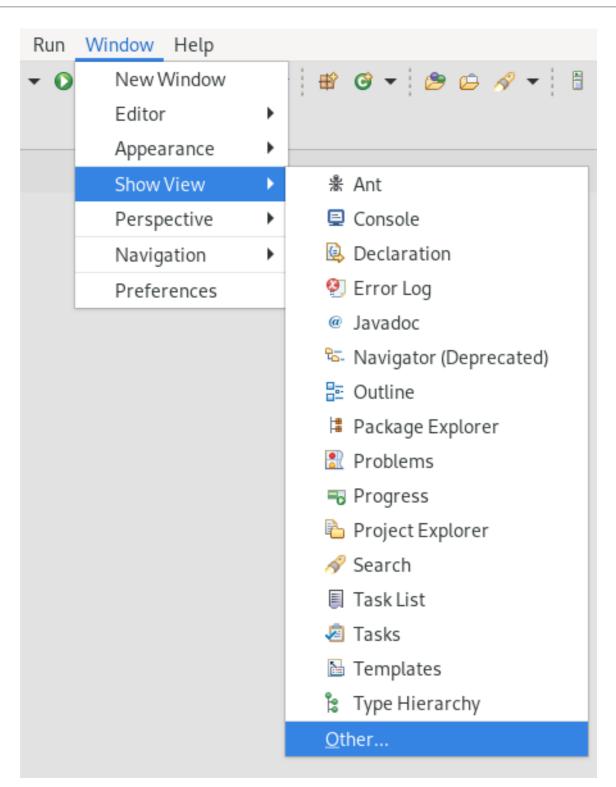
The OpenShift Application Explorer is set in CodeReady Studio as the default view. In case you need to open it manually, follow the instructions in Setting Up Openshift App Explorer View.

## 5.1. SETTING UP THE OPENSHIFT APPLICATION EXPLORER VIEW

The following section describes how to open OpenShift Application Explorer in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click Window → Show View → Other.



The **Show View** window appears.

Sh	ow View		×	
openshift			Ξ	
👻 🗁 JBoss Tools				
🗢 OpenShift App	olication Explo	rer		
😂 OpenShift Exp	olorer			
	Cancel	Open		
3. Enter <b>OpenShift</b> in the search f	ield.			
4. Select OpenShift Application E	Explorer.			
5. Click <b>Open</b> . The <b>OpenShift Application Exp</b>	<b>plorer</b> view appears	5.		
History 🖞 Synchronize 📩 Git Stagin			OpenShift Applica	57
	g 🔁 on nentrog 🖽		оренони стрриса	~

O https://kubernetes.default.svc/

# 5.2. CONNECTING TO THE OPENSHIFT CLUSTER USING OPENSHIFT APPLICATION EXPLORER

80

The following section describes how to login to the OpenShift cluster in CodeReady Studio using OpenShift Application Explorer.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Open OpenShift Application Explorer.

З.	Click C	an'i	t connect	to cluster.	Click to lo	ogin						
	🗐 Hist	ory	<mark>≝</mark> 0 Synchr	齿 Git Sta	🐴 Git Refl	🔲 Propert	陷 Project	🖉 Termin	🖸 OpenS	ឌ		
									-	Ē	¢2,	000
	<b>- O</b> hi	ttps	://kubernet	es.default.sv	c/							
		· · · · · · · · · · · · · · · · · · ·		to cluster. C								
					ينغ ا							
	The <b>Lo</b>	gin	window a	ppears.								
						Login						
	-					5						
			o Opens								ŗ	7
	😢 Use	er ai	nd passw	ord or toke	n must be	provided						
										0	PENS	HIFT
	URL	.:		https://ap	i.crc.testir	ng:6443/		Pa	ste login o	om	man	d
	Use	rna	me:									
	Pass	514/6	ard.									5
	ras:	5000	лu.					_				
	Toke	en:						Re	trieve tok	en		
	?							Cancel		Fin	ish	
	0											

- Paste your OpenShift API URL into the URL field.
   For more information on accessing your cluster through OpenShift API URL, visit Red Hat OpenShift - Accessing your Services.
- 5. Enter your username and password or token.
- 6. Click Finish.

#### 5.2.1. Browser-based token retrieval

Alternatively to providing your username and password or token, you can use browser based token retrieval to log in to your OpenShift cluster. There are two login options, "Paste Login Command" and "Retrieve Token".

#### Use "Paste Login Command"

- Paste your OpenShift API URL into the URL field.
   For more information on accessing your cluster through OpenShift API URL, visit Red Hat OpenShift - Accessing your Services.
- 2. Visit the OpenShift Container Platform web UI.

Red Hat OpenShift Container Platfo	orm 🏼 🕀	€ kube:admin ◄	
🗢 Administrator 🗸	You are logged in as a temporary administrative user. Update the <u>cluster OAr</u> to log in.	Copy Login Command 🛛 🖾	•
Home 🗸	Dashboards		
Dashboards			
Projects	Overview		
Search			
Explore	Cluster Health		
Events			
Operators	✔ Cluster is healthy		
Workloads 🗸	Cluster Capacity		•

- 3. Click the drop-down menu in the top right corner.
- 4. Click Copy Login Command.
- 5. Click Display Token.
- 6. Copy the login command.
- 7. In the Sign in to OpenShift window, click Paste login command.

_			
•	Login		
Sign in to Open	Shift		<b>C</b>
😣 User and passw	ord or token must be provided		
			OPENSHIFT
URL:	https://api.crc.testing:6443/	Paste log	in command
Username:			
Password:			
Token:		Retrieve	token
i okeni			
?		Cancel	Finish

#### 8. Click Finish.



#### NOTE

For OpenShift 3, the login command is copied into your clipboard automatically.

#### Use "Retrieve Token"

- Paste your OpenShift API URL into the URL field.
   For more information on accessing your cluster through OpenShift API URL, visit Red Hat OpenShift - Accessing your Services.
- 2. Click **Retrieve token**.

•	Login	
Sign in to Ope	nShift	<b>C</b>
😣 User and pass	word or token must be provided	
		OPENSHIFT
URL:	https://api.crc.testing:6443/	Paste login command
Username:		
Password:		
Token:		Retrieve token
0		Cancel Finish
		Cancel Finish

- 3. Enter your username and password.
- 4. Click Log in.
- 5. Click **Display Token**.
- 6. Click Finish.

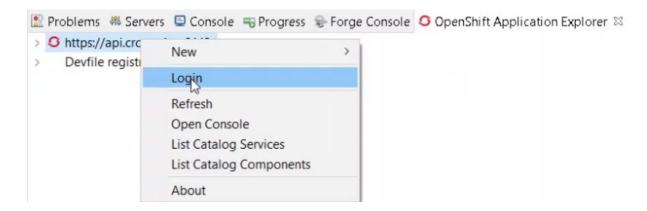
Your projects now appear in the **OpenShift Application Explorer** view.

## 5.3. SETTING UP A DEVELOPER SANDBOX USING OPENSHIFT TOOLS

The following section describes how to bootstrap and login to a Developer Sandbox in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Open OpenShift Application Explorer.



- 3. Right-click your OpenShift connection.
- 4. Click Login.

The **Sign in to OpenShift** window appears.

•	Login			
Sign in to Ope	enShift	6		
😢 User and pass	sword or token must be provided			
		OPENSHIFT		
Enter the cluster URL and the required credentials. You can also bootstrap a Red Hat Developer Sandbox cluster using your Red Hat account				
URL:	https://api.crc.testing:6443/	Paste login command		
Username:				
Password:				
Token:		Retrieve token		
?		Cancel Finish		

- 5. Click Red Hat Developer Sandbox
- 6. Provide the credentials of your Red Hat account and click **Log in**. Your Developer Sandbox has been bootstrapped.

The Login to Red Hat Developer Sandboxwindow appears.

•	Red Hat Developer Sandbox	
Login to Red Hat Developer Sandbox		C
Please login to Red Hat SSO if required, then provid	de required information to bootstrap your Red Hat Developer Sandbox.	
	0	PENSHIFT
Your Red Hat Developer Sandbox is ready, let's log	ogin now !!!	
	< Back Next > Cancel Fin	nish

7. Click Next.

### 8. Click DevSandbox.

9. Provide the credentials of your Red Hat account again and click Log in.

### 10. Click Display Token.

### 11. Click Finish.

Your Token is displayed in the Sign in to OpenShift window.

•	Login	
Sign in to Ope	nShift	0
Please sign in to	your OpenShift server.	J S
		OPENSHIFT
Enter the clust	er URL and the required credentials. You can also bootstrap a <u>Red Hat Developer Sandbox</u> cluster us	ing your Red Hat account
URL:	https://api.sandbox.x8i5.p1.openshiftapps.com:6443	Paste login command
Username:		
Password:		
Token:	******	Retrieve token
?	Car	ncel Finish

### 12. Click Finish.

You are now logged in to your Developer Sandbox.

Your Developer Sandbox shows in the OpenShift Application Explorer view.

## 5.4. BUILDING AN APPLICATION BASED ON DEVFILES

To deploy applications based on devfiles, you need an empty project in your local workspace as well as an empty project in OpenShift, for which you need to create a devfile component. After the component is established, your project will be updated and local and remote artifacts created in OpenShift.

### 5.4.1. Creating an empty project

The following section describes how to create an empty project in CodeReady Studio.

- 1. Start CodeReady Studio.
- Press Ctrl+N. The Select a wizard window appears.

•	New		
Select a wizard			-
Create a new project resource			
Wizards:			
type filter text			
🔻 🗁 General			
🖆 Faceted Project			
🗳 File			
🗳 Folder			
📂 Project			
🖹 Untitled Text File			
🕨 🗁 Batch			
Show All Wizards.			
Reck	Next >	Cancel	Finish
C Dack	Next>	Cancer	FIIISI

- 3. Select **General → Project**.
- 4. Click Next.

The **New Project** window appears.

•	New Project	
Project		
Create a new proje	ct resource.	
Project name: my	-empty-project	
<mark> </mark>	ition	
Location: /home/e	egebhard/workspace12.18/my-empty-project	B <u>r</u> owse
Working sets		
Add projec <u>t</u> to	working sets	Ne <u>w</u>
Working sets:	•	S <u>e</u> lect
	< Back Next > Cancel	Finish

- 5. Name your project.
- 6. Select the location for your project.
- 7. Click Finish.

Your newly created empty project is now listed in the **Project Explorer** view.

### 5.4.2. Creating an empty OpenShift project using OpenShift Application Explorer

The following section describes how to create an empty project using OpenShift Application Explorer in CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Right-click any place in **OpenShift Application Explorer** → **New** → **Project**.

	New	•	Project	t
	Login			-R
	Open Console			
	List Catalog Services			
	List Catalog Components	;		
	About			
The <b>New project</b> window app				
<sup>-</sup> he <b>New project</b> window app	ears. New project			
The <b>New project</b> window app				
The <b>New project</b> window app Project name: my-new-project				
Project name:				

- 4. Name your project.
- 5. Click **OK**.

Your newly created project is now listed in the **OpenShift Application Explorer** view.

### 5.4.3. Creating a devfile component using OpenShift Application Explorer

The following section describes how to create a devfile component using OpenShift Application Explorer in CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Right-click the target  $Project \rightarrow New \rightarrow Component$ .

🖹 Problems 🛛 @ Javadoc	😣 Declaration		🖥 Progress	🖸 OpenSł	nift Application Explorer នេ
→ O https://api.crc.testing:6443/					
my-new-project			,		
	New	×	Compo	nent	
	Delete		Service	2	]

The **Create component** window appears.

•	Create component		
Create component Specify component par	rameters.		OPENSHIFT
Name: Eclipse Project:	Devfile my-empty-project		Browse
i Your project is en Component type:	t is empty, you can initialize it from starters (template e: java-maven java-openliberty java-quarkus java-springboot java-springboot java-vertx nodejs python python-django > S21		
Component version: Project starter: Application: Push after create:	vertx-ex My App		▼
?		Cancel	Finish

4. Name your project.

- 5. Click **Browse** to select your **Eclipse Project**.
- 6. Set your Component type to java-vertx.
- 7. Set the Project starter to java-vertx.
- 8. Name your application.
- 9. Clear the **Push after create** check box.
- 10. Click Finish.

The **Console** view appears, displaying the validation process.

Your newly created component is now listed in the **OpenShift Application Explorer** view under your project.

### 5.4.4. Devfile registry management using OpenShift Application Explorer

The following section describes how to create, delete, and edit devfile registries using OpenShift Application Explorer in CodeReady Studio.

### Adding a devfile registry

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Devfile registries are displayed under the **Devfile registries** node.

📃 Console 🛭 OpenShift Explorer 🗳 OpenShift Application Explorer 🛱

- S https://api.engint.openshift.com:443/
- Devfile registries
- 4. If you expand the devfile registry node, all devfiles of that registry are shown.

📮 Console 🛯 OpenShift Explorer 🗳 OpenShift Application Explorer 😂

- S https://api.engint.openshift.com:443/
- Devfile registries
  - DefaultDevfileRegistry https://github.com/odo-devfiles/registry
  - Django Python3.7 with Django
  - Maven Java Upstream Maven and OpenJDK 11
  - NodeJS Runtime Stack with NodeJS 12
  - Open Liberty Open Liberty microservice in Java
  - Python Python Stack with Python 3.7
  - Quarkus Java Upstream Quarkus with Java+GraalVM
  - Spring Boot<sup>®</sup> Spring Boot<sup>®</sup> using Java
  - 🕨 😭 Vert.x Java Upstream Vert.x using Java
  - ▶ 🗇 WildFly Bootable Jar Java stack with WildFly in bootable Jar mode, OpenJDK 11 and Maven 3.5
  - WildFly Java Upstream WildFly
- To add a new devfile registry, right-click **Devfile registries** and click **new**.

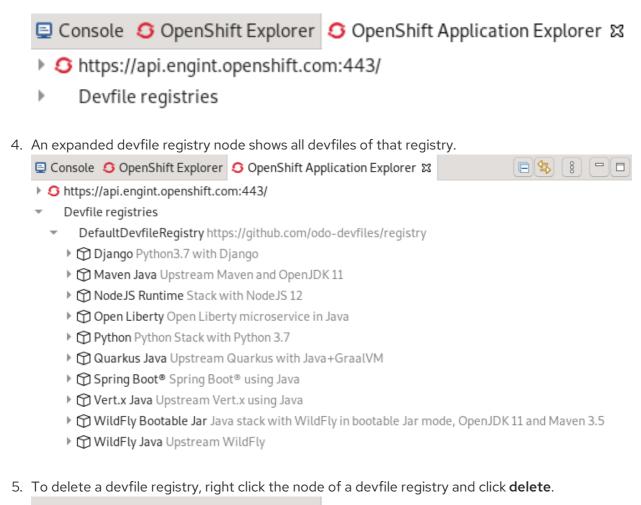
📃 Conse	ole 🖸 OpenShift Explorer	🟮 OpenShift Appl	ication Explorer 🛛
🕨 🖸 htt	ps://api.engint.openshift.co	m:443/	
▼ De <sup>*</sup>	vfile registries	New	Registry
~	DefaultDevfileRegistry http		
) ⊀	Django Python3.7 with D	jango	
) ⊀	🕤 Maven Java Upstream Ma	aven and OpenJDK	11
The <b>Create</b>	e devfile registry window appears.		
•	Create dev	/file registry	
Create de	evfile registry		6
Specify a I	name for the registry, and enter th	e URL to the registry.	$\mathbf{i}$
			OPENSHIFT
Name:	my-new-registry		
URL:	https://github.com/eclipse/che.gi	t	
Secure:			
?		Cancel	Finish

- 5. Name your devfile registry.
- 6. Paste your devfile URL.
- 7. Click Finish.

Your newly created devfile registry is now listed in the **OpenShift Application Explorer** view under **Devfile registries**.

### Deleting a devfile registry

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Devfile registries are displayed under the **Devfile registries** node.



💷 Console 🖸 OpenShift Explorer 🗳 OpenShift Application Explorer 🖾

- S https://api.engint.openshift.com:443/
- Devfile registries
  - DefaultDevfileRegistry https://github.com/odo\_devfiles/registry
    - Delete Diango Python 3.7 with Diange
    - Maven Java Upstream Maven and OpenJDK 11

Your devfile registry is now deleted.

### Editing a devfile registry

• To edit a devfile registry, use the YAML editor. The YAML editor provides syntax validation and content assist.

### Additional resources

For further information on devfiles, visit Introduction to Devfile.

## 5.5. BUILDING AN APPLICATION BASED ON S2I FILES

To deploy applications based on S2I files, you need a launcher project in your local workspace as well as an empty project in OpenShift, for which you need to create a component. After the component is established, your project will be updated and local and remote artifacts created in OpenShift.

### 5.5.1. Creating a launcher project

The following section describes how to create a launcher project in CodeReady Studio.

### Procedure

- 1. Start CodeReady Studio.
- Press Ctrl+N.
   The Select a wizard window appears.

		New		×
Select a wizard				Ď
Wizards:				
launcher				•
🕶 🗁 Launcher				
Launcher pro				
Show All Wizards.				
?	< Back	Next >	Cancel	Finish

- 3. Enter Launcher in the search field.
- 4. Select Launcher project.
- 5. Click Next.

The **New Launcher project** window appears.

Generate a	project based on mission and runtime. 👘 🛄	
Generate an	Eclipse project by specifying a mission and runtime variant.	
	ll generate an application for you. By picking a mission you determine wha ion will do. The runtime then picks the software stack that's used to his aim.	
Mission:	rest-http 👻	
	Map business operations to a remote procedure call endpoint over HT using a REST framework	
Runtime:	nodejs v12-community 🗸	
	Runs a Node.js HTTP application	
Project nam	e: my-launcher-project	
🔽 Use defa	ault location	
Location:	/home/levi/eclipse-workspace/my-launcher-project	
Maven Artifa	act:	
Artifact id:	booster	
Group id:	io.openshift	
Version:	0.0.1-SNAPSHOT	

### 6. Set Mission to rest-http.

### 7. Set Runtime to vert.x community.

- 8. Name your project.
- 9. Select the location for your project.
- 10. Click Finish.

Note that the process of resolving dependencies might take some time to complete.

Your newly created launcher project is now listed in the **Project Explorer** view.

### 5.5.2. Creating an empty OpenShift project in OpenShift Application Explorer

The following section describes how to create an empty project using OpenShift Application Explorer in CodeReady Studio.

### Procedure

1. Right-click any place in **OpenShift Application Explorer** → **New** → **Project**.

🖹 Problems 🛛 Javadoc 🖳 Declara	ation 🤇 OpenShift Application	Explo	orer 🏼
O https://api.crc.testing:6443/			
	New	•	Project
	Login		<u>A</u>
	Open Console		
	List Catalog Services		
	List Catalog Components		
	About		

The **New project** window appears.

New project	×
Project name:	
my-new-project	
Cancel	ЭК

- 2. Name your project.
- 3. Click **OK**.

Your newly created project is now listed in the **OpenShift Application Explorer** view.

### 5.5.3. Creating an S2I component using OpenShift Application Explorer

The following section describes how to create a component using OpenShift Application Explorer in CodeReady Studio.

### Procedure

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Right-click the target  $Project \rightarrow New \rightarrow Component$ .

🖹 Problems @ Javadoc 🗟 Declaration 🔫 Progress	S OpenShift Application Explorer ☎
---	------------------------------------

🔹 🕨 🛉 my-new-project 🥻			
	New	•	Component
	Delete		Service

The Create component window appears.

•	Create component	
Create component		$\sim$
Specify component pa	rameters.	<u> </u>
		OPENSHIFT
Name:	my-s2i-component	
Eclipse Project:	my-launcher-project	Browse
Component type:	nodejs python python-django S2I dotnet golang httpd java nginx nodejs perl	
Component version:	latest	•
Project starter:		•
Application:	my-s2i-App	
Push after create:		
?		Cancel Finish

- 4. Name your project.
- 5. Click **Browse** to select your **Eclipse Project**.
- 6. Click on the arrow next to **S2I** and set your **Component type** to **java**.
- 7. Set the **Component version** to **latest**.
- 8. Name your application.
- 9. Clear the **Push after create** check box.

### 10. Click Finish.

The **Console** view appears, displaying the validation process.

Your newly created component is now listed in the **OpenShift Application Explorer** view under your project.

# 5.6. DEPLOYING A COMPONENT ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER

The following section describes how to deploy a component on a cluster using OpenShift Application Explorer in CodeReady Studio.

### Procedure

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Expand your project.
- 4. Expand your application.
- 5. Right-click your component → Push.
   Problems @ Javadoc Declaration Console Progress OpenShift Application Explorer
  - - 🕶 🏜 my-new-project
      - 🕶 🗞 my-new-application

▶ I my-new-component  o not pushed		
	New	•
	Push	
	Delete	

The **Console** view appears, displaying the process of file synchronization.

# 5.7. DEFINING AN EXTERNAL ACCESS URL USING OPENSHIFT APPLICATION EXPLORER

The following section describes how to define an external access URL using OpenShift Application Explorer in CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. Start OpenShift Application Explorer.
- 3. Expand your project.
- 4. Expand your application.
- 5. Right-click your **component**  $\rightarrow$  **New**  $\rightarrow$  **URL**.

	New 🕨	URL
	Follow Log	Storage
	Show Log	
	Describe	
	Link Component	
	Link Service	
📙 Getting Started 🎐 Software/Update	Debug	
🖹 Problems @ Javadoc 😣 Declaration 🚍	Undeploy	oplication Explorer ຊ
🕨 🏜 kube-system	Watch	
👻 🏜 my-new-project	Push	
👻 🗞 my-new-application	Delete	
🔤 my-new-component 💿 pushed 🚽		
🕨 🏜 openshift		

The **Create URL** window appears.

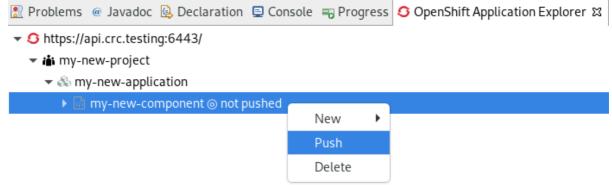
	Create url	×
Create un Specify a scheme o	name for the url, choose a port to bind to and select a secure (http	s) OPENSHIFT
Name: Port: Secure:	my-new-url 8080	
?	Cancel	Finish

- 6. Name your URL.
- 7. Set the **Port** value to **8080**.

### 8. Click Finish.

The **Console** view appears, displaying the process of URL creation.

## 9. In **OpenShift Application Explorer**, right-click your **component** $\rightarrow$ **Push**.



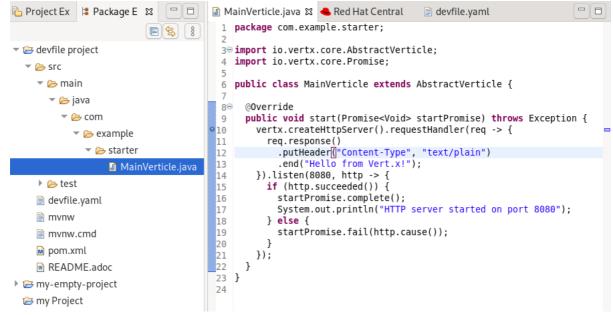
The **Console** view appears, displaying the process of file synchronization.

Your newly created URL is now listed in the **OpenShift Application Explorer** view under your component.

# 5.8. DEBUGGING AN APPLICATION ON A CLUSTER USING OPENSHIFT APPLICATION EXPLORER

The following section describes how to debug an application on a cluster using OpenShift Application Explorer in CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. In the Project Explorer view, locate the **MainVerticle.java**(devfiles) or **HttpApplication.java**(S2I) file and double-click to open it.
- 3. Double-click on the left ruler column to set a breakpoint.



- 4. Open OpenShift Application Explorer.
- 5. Expand your project.
- 6. Expand your application.

7. Right-click your **component**  $\rightarrow$  **Debug**.

	New 🕨	
	Follow Log	
	Show Log	
	Describe	
	Link Component	
	Link Service	
📙 Getting Started 💲 Software/Update	Debug	
🖹 Problems @ Javadoc 😣 Declaration 🗐	Undeploy	plication Explorer 없
🕨 🏜 kube-system	Watch	
👻 🏜 my-new-project	Push	
🔻 🗞 my-new-application	Delete	
🕨 🕨 🕞 my-new-component 🔘 pushed		

🕨 📸 openshift

The **Console** view appears.

- 8. In OpenShift Application Explorer, expand your component.
- 9. Right-click your **url** → **Open in Browser**.

🛃 Problems @ Javadoc 😣 Declaration 🗐 Console 🖸 OpenShift Application Explorer 🛱

- кире-ририс
- 🕨 📸 kube-system
- 🕶 📸 my-new-project
  - ▼ ♣ my-new-application
    - ▼ In my-new-component 
       pushed

Open in Browse     Delete	🚥 my-new-url 8080 Pushed 4	
Delete	1	
	<ul> <li>iii openshift-apiserver</li> </ul>	Delete

The Confirm Perspective Switch window appears.

### 10. Click Switch.

The **Debug Perspective** window appears displaying the debugging process.

## CHAPTER 6. QUARKUS TOOLS BASICS IN CODEREADY STUDIO

Quarkus is a Kubernetes-Native full-stack Java framework aimed to optimize work with Java virtual machines. Quarkus provides tools for Quarkus application developers, helping to reduce the size of Java applications and container image footprints, as well as the amount of memory required.

### Prerequisites

• The latest version of JBoss Tools is installed. For more information, see JBoss Tools Downloads.

## **6.1. CREATING A NEW QUARKUS PROJECT**

The following section describes how to create a new Quarkus project in CodeReady Studio.

### Procedure

- 1. Start CodeReady Studio.
- 2. Press Ctrl+N.

The Select a wizard window appears.

		New		×
Select a wizard				Ď
Wizards:				
quarkus				×
🔻 🗁 Quarkus				
🕨 Quarkus Pro	ject			
Show All Wizards				
٢	Pack	Nexts	Cancel	Finish
	< Back	Next >	Cancet	Finish

- 3. Enter **Quarkus** in the search field.
- 4. Select Quarkus Project.
- 5. Click **Next**.

The New Quarkus project window appears.

	New Quarkus project	×
Project type		
Select the code.	quarkus.io endpoint and project type	
	o will generate an application for you. Select the project type accord build tool. Then select the Quarkus dependencies you plan to use in n.	
Project type:	Maven -	
Project name:	my-quarkus-project	
🗌 Use default	location	
Location:	/home/user/eclipse-workspace/my-quarkus-project	
?	< Back Next > Cancel Finish	h

- 6. Select the needed project type.
- 7. Name your project.
- 8. Select the location for your project.
- 9. Click Next.

The **Project type** window appears.

Project type	New Quarkus projec	t	
Select the code	e.quarkus.io endpoint and project type		
Maven Artifa	:t:		
Artifact id:	code-with-quarkus		
Group id:	org.acme		
Version:	1.0.0-SNAPSHOT		
REST:			
Class name:	org.acme.ExampleResource		
Path:	/hello		
?	< Back Next >	Cancel	Finish

- 10. Ensure that the default values are correct.
- 11. Click Next.

The **Quarkus extensions** window appears.

New Quarkus project ×				
Quarkus extensions				
Select the Quarkus extensions for your project				
Clicking on a category will display the extensions in the middle colu are displayed in the third column.	mn. Double clicking on an extension will add/remove the extension f	rom the selected extensions list. The current selected exte		
Categories	Extensions	Selected		
Web	RESTEasy JAX-RS (Included)	RESTEasy JAX-RS (Included)		
Data	RESTEasy JSON-B	RESTEasy Qute (Experimental)		
Messaging	RESTEasy Jackson			
Core	Hibernate Validator			
Reactive	REST Client			
Cloud	REST Client JAXB			
Observability	REST Client JSON-B			
Security	REST Client Jackson			
Integration	REST resources for Hibernate ORM with Panache (Experimental)			
Business Automation	RESTEasy JAXB			
Serialization	RESTEasy Mutiny (Preview)			
Miscellaneous	RESTEasy Qute (Experimental)			
Compatibility	Reactive Routes			
Alternative languages	SmallRye GraphQL (Preview)			
	SmallRye JWT			
	SmallRye OpenAPI			
	Undertow Servlet			
	Undertow WebSockets			
	gRPC (Experimental)			
Qute Templating integration for RESTEasy. Click to open guide				
(?)	< Back	Next > Cancel Finish		
$\checkmark$	Datk	Cancer Philish		

- 12. Select the needed Categories for your projects.The available extensions of the selected category are displayed in the Extensions column.
- Select the needed Extensions for your projects.
   Double-click on the extension to select or deselect it. The selected extensions appear in the Selected column.
- 14. Click Finish.

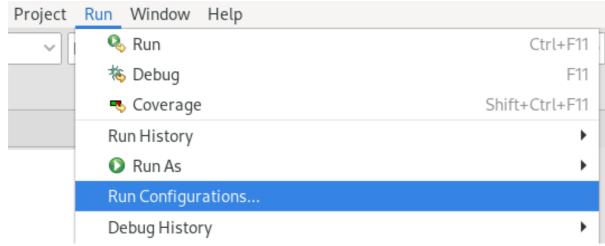
Your newly created Quarkus project is now listed in the **Project Explorer** view.

## **6.2. RUNNING A QUARKUS APPLICATION**

The following section describes how to run a Quarkus application in CodeReady Studio.

### Procedure

- 1. Start CodeReady Studio.
- 2. Click Run  $\rightarrow$  Run Configurations.



The Run Configurations window appears.

3. Scroll down to **Quarkus Application**.

**Run Configurations** 

×

#### Create, manage, and run configurations

📑 🖹 🐢 🗎 🗶 🖻	Configure launch settings from this dialog:
type filter text	$\square$ - Press the 'New Configuration' button to create a configuration of the selected type.
	- Press the 'New Prototype' button to createonfiguration prototype of the selected type.
HTTP Preview	🖚 - Press the 'Export' button to export the selected configurations.
🗄 J2EE Preview	📄 - Press the 'Duplicate' button to copy the selected configuration.
🜌 Java Applet	× - Press the 'Delete' button to remove the selected configuration.
Java Application Ju JUnit	$\overline{\gamma}$ - Press the 'Filter' button to configure filtering options.
📌 JUnit Plug-in Test	- Edit or view an existing configuration by selecting it.
🖶 Launch Group	- Select launch configuration(s) and then selink Prototype' menu item to link a prototype.
🚥 Launch NPM	- Select launch configuration(s) and then se Prototype' menu item to unlink a prototype.
😘 Local Camel Context	- Select launch configuration(s) and then ses' menu item to reset with prototype values.
m² Maven Build	Seccer adhen conngaration (s) and alen ses mena term to reset with prototype values.
Node.js application	Configure launch perspective settings from the <u>'Perspectives'</u> preference page.
<ul> <li>OSGi Framework</li> <li>Quarkus Application</li> </ul>	
Run Docker Image	The W Configuration
× XSL	🖻 New Prototype
Filter matched 26 of 26 iter	Se Export
	Duplicate
?	X Delete Close Run
	Link Prototype

4. Right-click Quarkus Application → New Configuration

Right-click Guarkus Ap	$p_{\text{incation}} \rightarrow \text{New Configuration}.$				
•	Run Configurations				
Create, manage, and run configurations					
	Name: New_configuration				
type filter text Local Camel Context Maven Build Node.js application OSGi Framework Call Camel Context Quarkus Application New_configuration Run Docker Image TestNG	Project Environment Common Project: (my-quarkus-project Profile	Browse			
→ XSL Filter matched 28 of 28 iter		Revert Apply			
0		Close Run			

5. Name your configuration.

6. Click **Browse** to locate your project.

### NOTE

It is possible to add environment variables to your Quarkus project. To add a new environment variable, click **Environment**  $\rightarrow$  add and select a name and value.

•		Run Configurations			
Create, manage, and ru	In configurations				
type filter text	Name: New_con Project C Enviro Environment vari	onment 🔲 <u>C</u> ommon			
Java Application	Variable	Value			<u>A</u> dd
🔊 JUnit Plug-in Test	• N	lew Environment Variat	ole		Se <u>l</u> ect.
Launch Group	Name:				E <u>d</u> it
🚥 Launch NPM 🦙 Local Camel Context	Value:		Variables		Remov
m² Maven Build					
Node.js application		Cancel	OK		
+ OSGi Framework		_	_	_	Paste
Quarkus Application     New_configuration     Run Docker Image     TestNG	• <u>Append</u> envir	onment to native enviro re environment with spe		nment	
≫ XSL				Revert	Apply
Filter matched 28 of 28 ite	1				
(?)				Close	Ru

- 7. Click Apply.
- 8. Click Run.

The **Console** view appears.

Your application will start after the built process.

## 6.3. DEBUGGING A QUARKUS APPLICATION

The following section describes how to debug a Quarkus application in CodeReady Studio.

- 1. Start CodeReady Studio.
- 2. Click  $Run \rightarrow Debug$  Configurations.

Project	Run Window Help	
nfiguratio	🗞 Run	Ctrl+F11
<del>(2)</del> 8	🎋 Debug	F11
₽ ŏ	🤜 Coverage	Shift+Ctrl+F11
	Run History	•
	🜔 Run As	•
	Run Configurations	
	Debug History	•
	🎋 Debug As	•
	Debug Configurations	
	Coverage History	•

The **Debug Configurations** window appears.

	Debug Configurations	×
Create, manage, and run	configurations	Ť
Image: Second Structure       Image: Second Structure         Image: Second Structure       Image: Second Structure </td <td>Name: New_configuration Project  my-new-quarkus-project</td> <td><u>B</u>rowse</td>	Name: New_configuration Project  my-new-quarkus-project	<u>B</u> rowse
Filter matched 32 of 32 item	Revert	Apply
0	Close	Debug

- 3. Expand Quarkus Application.
- 4. Select your configuration.
- 5. Click **Debug**.

### The **Console** view appears.

Your Quarkus application starts and connects to a remote JVM debug configuration. If you set breakpoints in your application source files, the execution automatically stops after reaching the breakpoint.

## 6.4. USING LANGUAGE SUPPORT IN CODEREADY STUDIO

Every Quarkus application is configured through an **application.properties** configuration file. The content of this configuration file is dependent on the set of Quarkus extensions that your application is using.

Quarkus Tools includes content assist, which provides code completion, validation, and documentation. Code completion allows you to quickly complete statements in your code. Multiple choices are available to you via popups. This language support is now available for Kubernetes, OpenShift, S2i, Docker properties, MicroProfile REST Client properties, and MicroProfile Health artifacts. Note that language support for MicroProfile REST Client properties needs to be enabled separately. For more information, see Section 6.4.2, "Enabling language support for MicroProfile REST Client properties".

### 6.4.1. Using Quarkus content assist

The following section describes how to use Quarkus **application.properties** content assist in CodeReady Studio.

### Prerequisites

An existing Quarkus project.
 For more information on how to create a Quarkus project, see Section 6.1, "Creating a new Quarkus project".

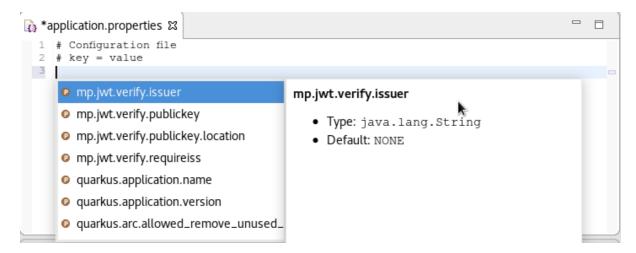
- 1. Start CodeReady Studio.
- 2. Start Project Explorer.
- 3. Expand your **Quarkus project** → src/main/resources.
- 4. Right-click **application.properties** → **Open With** → **Generic Text Editor**.

ြဲ Project Explorer 🕱	E 🕏 7 % -		
<ul> <li>♥ ➡ my-new-quarkus-project</li> <li>▶ ● ➡ src/main/java</li> <li>♥ ● src/main/resources</li> <li>▶ ▷ META-INF</li> </ul>			
<ul> <li>application.properties</li> <li>src/test/java</li> <li>JRE System Library [JavaS</li> <li>Maven Dependencies</li> </ul>	New Show In Open	Shift+Alt+W F3	
<ul> <li>▷ src</li> <li>▷ target</li> <li>⊇ mvnw</li> <li>⊇ mvnw.cmd</li> <li>⊇ pom.xml</li> <li>⊇ README.md</li> </ul>	Open With  Copy  Copy Qualified Name  Copy Qualified Name  Paste  Colore  Delete  Build Path Source	Ctrl+C Ctrl+V Shift+Alt+S	O 🖹 Text Editor

The Generic Text Editor window appears.

- 5. Navigate to an empty line.
- 6. Press Ctrl+Space to invoke code completion.

The code completion suggestions appear. Hover the mouse over the suggestions to display documentation.



If you enter a wrong value, the editor underlines the error with a red wavy line.

🚯 *application.properties 😫	
1 # Configuration file 2 # key = value	
😡 3 quarkus.http.port=false	

### Additional resources

• Language support for MicroProfile REST Client properties needs to be enabled separately. For more information, see Section 6.4.2, "Enabling language support for MicroProfile REST Client properties".

### 6.4.2. Enabling language support for MicroProfile REST Client properties

The following section describes how to enable language support for MicroProfile REST Client properties.

### Prerequisites

An existing Quarkus project.
 For more information on how to create a Quarkus project, see Section 6.1, "Creating a new Quarkus project".

- 1. Start CodeReady Studio.
- 2. Start Project Explorer.
- 3. Expand your **Quarkus project** → src/main/java.
- Right-click org.acme → New → Other. The Select wizard window appears.

		New		×
Select a wizard				-
Create a new file re	source			
Wizards:				
file				Ø
🔫 🗁 General				
📑 File				
貸 Untitled Te	ext File			
🕶 🗁 Batch				
Batch Job >	(ML File			
🕶 🗁 CDI (Contexts	and Dependency Inject	ction)		
🔐 beans.xml	File			
🛨 🔁 Hibernate				
Show All Wizard	ls.			
2	< Back	Next >	Cancel	Finish
$\odot$	Dack	Next >	Cancer	

- 5. Enter **file** in the search field.
- 6. Select File.
- 7. Click Next.

The **Create a new file resource**window appears.

Create New File	×
File	
Create a new file resource.	
Enter or select the parent folder:	
my-new-quarkus-project/src/main/java/org/acme	
🔻 🦢 main	
🗁 docker	
🔫 🗁 java	
← 🗁 org	
🗁 acme	
resources	
▶ 🗁 test	
▶ 🗁 target	
RemoteSystemsTempFiles	
File name: MyServiceClient.java	
<u>A</u> dvanced >>	
<pre></pre>	Finish

8. Name your new file.

### 9. Click Finish.

10. Paste the following content into your newly created file:

package org.acme;
<pre>import javax.ws.rs.GET; import javax.ws.rs.Path; import javax.ws.rs.core.Response;</pre>
import org.eclipse.microprofile.rest.client.inject.RegisterRestClient;
<pre>@RegisterRestClient public interface MyServiceClient {   @GET</pre>



11. Press **Ctrl+S** to save the changes.

### Additional resources

- For more information on how to use language support, see Section 6.4.1, "Using Quarkus content assist".
- For information on adjusting language support, see Quarkus Using the REST client, Create the interface.

## CHAPTER 7. HIBERNATE TOOLS BASICS IN CODEREADY STUDIO

Hibernate Tools is a collection of tools for projects related to Hibernate version 5 and earlier. The tools provide Eclipse plugins for reverse engineering, code generation, visualization, and interaction with Hibernate.

## 7.1. CREATING A NEW JPA PROJECT

The following section describes how to create a new JPA project in CodeReady Studio.

### Prerequisites

You need to start the Sakila database server before you can create a new JPA project in CodeReady Studio.

- 1. Download the h2 version of the Sakila database.
- 2. Navigate to the directory that contains the **runh2.sh** file.
- 3. Execute the **runh2.sh** file:

\$./runh2.sh

- 1. Start CodeReady Studio.
- Press Ctrl+N. The Select a Wizard window appears.

		New		×
Select a wizard				-
Create a JPA project				
Wizards:				
JPA				B
→ ▷JPA				
GIPA Entities from	n Tables			
📽 JPA Entity 🔐 JPA ORM Mappi	ing File			
JPA Project	ngrite			
Show All Wizards.				
?	< Back	Next >	Cancel	Finish

- 3. Enter **JPA** in the search field.
- 4. Select JPA Project.
- 5. Click Next.

The New JPA Project window appears.

PA Project Configure JPA project se	ttings.			
Project name: my-nev	v-JPA-project			
Project location				
✓ Use default location				
Location: /home/levi/e	clipse-workspace/my-n	ew-JPA-project		Brow <u>s</u> e
Target runtime				
java-11-openjdk-11.0.7	7.10-1.fc32.x86_64		•	New <u>R</u> untime
JPA version				
2.1				
Configuration				
<custom></custom>			•	Mod <u>i</u> fy
Hint: Get started quickly	/ by selecting one of the p	re-defined project	conf	
EAR membership	, , ,			5
<u>A</u> dd project to an EA	R			
EAR pr <u>oj</u> ect name:			•	New <u>P</u> roject
Working sets				
Add projec <u>t</u> to work	ing sets			Ne <u>w</u>
Working sets:	5		_	S <u>e</u> lect
working sets.			_	S <u>e</u> lect

- 6. Name your project.
- 7. Select the location for your project.
- 8. Click the down-arrow in the **Target runtime** field to select the runtime server.

- 9. Set the **JPA version** to 2.1.
- 10. Click Next.

The **Java** window appears.

	×			
<b>Java</b> Configure project for	building a Java a	pplication.		
Source folders on bui	ld path:			
🕮 src				Add Folder
				<u>E</u> dit
				<u>R</u> emove
Default output folder				
build/classes				]
?	< Back	Next >	Cancel	Finish

- 11. Select the source folder.
- 12. Click Next.

The **JPA Facet** window appears.

	New JPA Project	×
JPA Fac	:et	
	ry configuration is disabled. The user may need to configure further path changes later.	
Platfor	m	
Hibe	rnate (JPA 2.1)	•
JPA im	plementation	
Туре:	Disable Library Configuration	•
classp	A facet requires a JPA implementation library to be present on the p ath. By disabling library configuration, the user takes on the respons ng that the classpath is configured appropriately via alternate mean	ibility of
Connec	tion	
<no< td=""><td>ne&gt;</td><td>•</td></no<>	ne>	•
	Ac	dd connection
		<u>Connec</u>

- 13. Click the down-arrow in the **Platform** field and select **Hibernate (JPA 2.1)**.
- 14. Add user libraries or set the **JPA Implementation Type** to **Disable Library Configuration**. For more information on how to set up user libraries, see Section 7.2, "Adding libraries".
- 15. Click Add connection.

The **Connection Profile** window appears.

New Connection Profile	×
<b>Connection Profile</b> Create a Generic JDBC connection profile.	
Connection Profile Types: generic Generic JDBC	
Name:	
Description (optional):	
? < Back Next > Cancel F	Finish

- 16. Enter **Generic** in the search field.
- 17. Select Generic JDBC.
- 18. Enter **Sakila** in the **Name** field.
- Click Next.
   The Specify a Driver and Connection Details window appears.
- 20. Click the **New Driver Definition** icon.

•					
New Driver Definition					

The New Driver Definition window appears.

	New Driver Defin	ition	×		
Specify a Driver Template and Definition Name Oriver files not specified in driver definition.					
Name/Type JAR List	Properties				
Available driver templates	5:				
Name	System Vendor	System Version			
Generic JDBC Drive	r Generic JDBC	1.0			
Driver name:					
Generic JDBC Driver					
Driver type:					
Generic JDBC Driver					
?		Cancel	K		

- 21. Select the **Generic JDBC Driver**.
- 22. Click the **JAR List** tab.

New Driver Definition Specify a Driver Template and Definition Name Required property in driver definition missing value: Driver Class.		
Name/Type JAR List Properties		
Driver files:		
/home/levi/sakila-h2/h2-1.4.200.jar	Add JAR	/Zip
	Edit JAR	/Zip
	Remove J	AR/Zi
	Clear	All
	Cancel	)K

- 23. Click the **Add JAR/Zip** button.
- 24. Select the **.jar** file for the Sakila database.
- 25. Click the **Properties** tab.

**New Driver Definition** 

#### Specify a Driver Template and Definition Name

😣 Required property in driver definition missing value: Driver Class.

Name/Type JAR List Pro	perties	
Properties:		
Property	Value	
<ul> <li>✓ General</li> <li>Connection URL</li> <li>Database Name</li> <li>Driver Class</li> <li>User ID</li> </ul>	jdbc: SAMPLE	
?		Cancel OK

- 26. Add jdbc:h2:tcp://localhost/./sakila to the Connection URL field.
- 27. Click the **Driver Class** field.

×

28. Click the three dots icon at the end of the **Driver Class** field. The **Available Classes from Jar List**window appears.

Available Classes from Jar List ×	¢
Provide the name of the driver class or select a class from the available jar O Type class name	s.
O Browse for class org.h2.Driver	
Cancel OK	

- 29. Select the **Browse for Class** option.
- 30. Select org.h2.Driver.
- 31. Click **OK**.
- 32. Enter **sa** in the **User ID** field.

	nodify details in the fields below to provide a unique I set any available and applicable property values.	
Name/Type JAR List Pro	perties	
Properties:		
Property	Value	
<ul> <li>✓ General</li> <li>Connection URL</li> <li>Database Name</li> </ul>	jdbc:h2:tcp://localhost/./sakila SAMPLE	
Driver Class	org.h2.Driver	
User ID	sa	

#### 33. Click $OK \rightarrow Finish \rightarrow Finish$ .

Your newly created JPA project is now listed in the **Project Explorer** view.

## 7.2. ADDING LIBRARIES

The following section describes how to add libraries to your Hibernate project in CodeReady Studio.

#### Procedure

- 1. Download Hibernate ORM.
- 2. Extract the files.
- 3. Start CodeReady Studio.
- 4. Click **Window** → **Preferences**.

Search Project Run	Window Help	
- 😰 🐨 😕 😂	New Window	- *> + -
· · ·	Editor	•
	Appearance	•
	Show View	•
	Perspective	•
	Navigation	•
	Preferences	

The **Preferences** window appears.

	Preferences	×
libraries 🗨	User Libraries	
<ul> <li>✓ Java</li> <li>✓ Build Path</li> <li>User Libraries</li> <li>Installed JREs</li> </ul>	User libraries can be added to a Java Build path and bun external archives. System libraries will be added to the launched. Defined user libraries:	
✓ Web		<u>N</u> ew
<ul> <li>HTML Files</li> <li>Client-side JavaS</li> </ul>		<u>E</u> dit
		Add <u>J</u> ARs
User Libraı		Add External JARs
		Remove
		Шр
		Down
		I <u>m</u> port
		Export
? 🖻 🗹 🔘	Cancel	Apply and Close

- 5. Enter Libraries in the search field.
- 6. Select User Libraries under Java.
- 7. Click the **New** button.

## The New User Library window appears.

- 8. Name your user library.
- 9. Click **OK**.
- 10. Select your new user library.

	Preferences		×
libraries 🗨	User Libraries		⇔ - ⇒ - 8
<ul> <li>✓ Java</li> <li>✓ Build Path</li> <li>User Libraries</li> <li>Installed JREs</li> </ul>	User libraries can be added to a Java Bu external archives. System libraries will launched. <u>D</u> efined user libraries:		
✓ Web	🛋 my-new-library		<u>N</u> ew
<ul> <li>HTML Files</li> <li>Client-side JavaS</li> </ul>			Edit
			Add <u>J</u> ARs
User Libraı		Ad	d E <u>x</u> ternal JARs
			<u>R</u> emove
			<u>U</u> р
			D <u>o</u> wn
			I <u>m</u> port
			E <u>x</u> port
? 🎽 🛆 🔘		Cancel	Apply and Close

- 11. Click the **Add External JARs** button.
- 12. Select the directory you extracted the Hibernate ORM files into.
- 13. Navigate to the /lib/required/ directory.
- 14. Select a **.jar** file.
- Click **Open**.
   Your selected **.jar** file appears under your user library.

type filter text	User Libraries	⇔ ⇒ ⇒ ⇒ ∞
<ul> <li>General</li> <li>Ansi Console</li> <li>Ant</li> </ul>	User libraries can be added to a Java Build path and bundle a number of external arch will be added to the boot class path when launched. Defined user libraries:	ives. System librarie
Apache Camel	▼ ➡ my-new-library	New
Data Management	image: a constraint of the second	110.00
Docker	Image: A start and the star	Edit
Forge	Glassmate-1.5.1.jar - /home/levi/Downloads/hibernate-release-5.4.18.Find	Add JARs
Fuse Tooling	dom4j-2.1.3.jar - /home/levi/Downloads/hibernate-release-5.4.18.Final/li	Add JARS
Gradle	FastInfoset-1.2.15.jar - /home/levi/Downloads/hibernate-release-5.4.18.F	Add External JARs.
Help	in the second	5
HQL editor	bibernate-core-5.4.18.Final.jar - /home/levi/Downloads/hibernate-release	Remove
Install/Update	istack-commons-runtime-3.0.7.jar - /home/levi/Downloads/hibernate-rel	Up
Java	jandex-2.1.3.Final.jar - /home/levi/Downloads/hibernate-release-5.4.18.F	Op
Appearance	javassist-3.24.0-GA.jar - /home/levi/Downloads/hibernate-release-5.4.18	Down
✓ Build Path	javax.activation-api-1.2.0.jar - /home/levi/Downloads/hibernate-release-	
Classpath Variat	javax.persistence-api-2.2.jar - /home/levi/Downloads/hibernate-release-!	Import
User Libraries	ai jaxb-api-2.3.1.jar - /home/levi/Downloads/hibernate-release-5.4.18.Final/	Evport
Code Coverage	▶  jaxb-runtime-2.3.1.jar - /home/levi/Downloads/hibernate-release-5.4.18.1	Export
Code Style	jboss-logging-3.3.2.Final.jar - /home/levi/Downloads/hibernate-release-5	
Compiler	jboss-transaction-api_1.2_spec-1.1.1.Final.jar - /home/levi/Downloads/hib	
Debug	🕨 👼 stax-ex-1.8.jar - /home/levi/Downloads/hibernate-release-5.4.18.Final/lib	
Editor	Image: style="text-align: center;">	
Installed JREs		
JUnit		
Properties Files Edi		
Java EE		
Java Persistence		
JBoss Tools		
JSON		
JVM Monitor		
Language Servers		
Maven		
<ul> <li>Mylyn</li> </ul>		

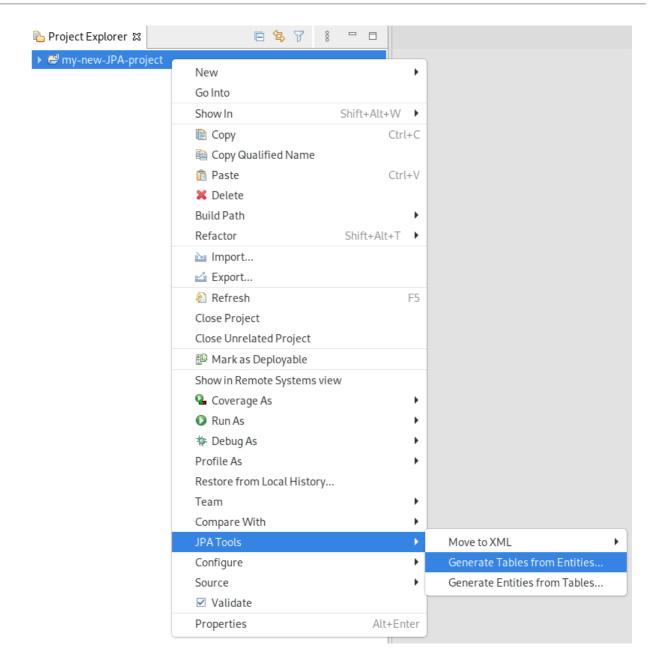
16. Click Apply and Close.

## 7.3. GENERATING TABLES FROM ENTITIES

The following section describes how to generate tables from entities for your Hibernate project in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Open Project Explorer.
- 3. Right-click your JPA project → JPA Tools → Generate Tables from Entities.



The Generate Tables from Entities window appears.

	Generate Tables from Entities	×
Use existing console	configuration or connection profile for database connection	
Output directory:	/my-new-JPA-project/src	Browse
File name	schema.ddl	
Export to Database		
🗹 Use Console Config	uration	
Console configuration:	my-new-JPA-project	•
Hibernate Version:	3.5	-
Database Settings		
Database Connection	sakila	•
Database dialect:	[Autodetect]	-
?	Cancel	Finish

- 4. Select the Use Console Configuration check box.
- 5. Click Finish.

## 7.4. CREATING A HIBERNATE MAPPING FILE

Hibernate mapping files specify how your objects relate to the database tables.

The following section describes how to create a Hibernate mapping file in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Press Ctrl+N.

The Select a wizard window appears.

		New		×
Select a wizard				-
Creates a new Hiberr	ate XML Mappin	g file		
Wizards:				
hibernate				∞
S Hibernate Co Hibernate Re	onfiguration File ( onsole Configurat everse Engineerii ML Mapping file (	tion ng File (reveng.xm	1()	
Show All Wizards.				
?	< Back	Next >	Cancel	Finish

- 3. Enter **Hibernate** in the search field.
- 4. Select Hibernate XML Mapping file (hbm.xml)
- 5. Click Next.

The Create Hibernate XML Mapping filewindow appears.

Add classes and	packages or wizard wi	ll create an empty m	apping file	
				Add Class Add Package
				Remove
				depth cont

- 6. Click the **Add Class** button to add classes.
- Click the Add Package button to add packages.
   Alternatively, you can create an empty .hbm.xml file by not selecting any packages or classes.
- 8. Select the **depth control** check box to define the dependency depth used when choosing classes.
- 9. Click **Next**.

The  $\ensuremath{\textit{New Hibernate XML Mapping files}}\xspace$  window appears.

New Hibernate XML Mapping files (hbm.xml) ×
New Hibernate XML Mapping files (hbm.xml)
Create a new empty XML Mapping files
Enter or select the parent folder:
my-new-JPA-project
▶ ≓my-new-JPA-project
➢ RemoteSystemsTempFiles
File name: hibernate.hbm.xml
Advanced >>
O Kart Next > Cancel Finish

- 10. Select the parent directory.
- 11. Name your **.hbm.xml** file .
- 12. Click Finish.

## 7.5. CREATING A HIBERNATE CONFIGURATION FILE

For reverse engineering, prototype queries, or Hibernate Core usage, a **hibernate.properties** or a **hibernate.cfg.xml** file is required. CodeReady Studio provides a wizard to generate the configuration file **hibernate.cfg.xml**.

The following section describes how to create a Hibernate configuration file in CodeReady Studio.

#### Procedure

1. Start CodeReady Studio.

#### 2. Press Ctrl+N.

The Select a wizard window appears.

		New		×
Select a wizard Create a new hibern	ate.cfg.xml file (He	elping with the init	ial JDBC setup etc.	)
Wizards:				
hibernate				Ø
🕶 🗁 Hibernate				
🖏 Hibernate G 🖏 Hibernate F	Configuration File ( Console Configurati Reverse Engineerin KML Mapping file (h	ion Ig File (reveng.xm	ι)	
?	< Back	Next >	Cancel	Finish

- 3. Enter **Hibernate** in the search field.
- 4. Select Hibernate Configuration file (cfg.xml).
- 5. Click Next.

The Create Hibernate Configuration file (cfg.xml) window appears.

				×
Create Hibernate Configuration file	(cfg.xml)			
A To load Configuration from 'hibernate. Configuration.configure() method	cfg.xml' you'll ha	ve to pass file nar	ne to	
Enter or select the parent folder:				
my-new-JPA-project/src				
h 🗘 🗘				
🗁 .settings				
🕨 🗁 build				
▶ 🗁 src				
BemoteSystemsTempFiles				
File name: hibernate.cfg.xml				
Advanced >>				
•	< Back	Next >	Cancel	Finish

- 6. Select the parent directory.
- 7. Click Next.

The Hibernate Configuration File (cfg.xml) window appears.

Hibernate Configur	ation Eile (cfg yn	nl)			×		
This wizard creates a			oernate.				
<u>C</u> ontainer:	/my-new-JPA-proj	ect/src					
Fi <u>l</u> e name:	hibernate.cfg.xml						
Hibernate version:	5.4 💌	5.4 🔻					
Session factory name:							
Get values from Conne	ection						
<u>D</u> atabase dialect:	MySQL				-		
D <u>r</u> iver class:	org.gjt.mm.mysql.Driver						
Connection <u>U</u> RL:	jdbc:mysql:// <hostname>/<database></database></hostname>						
Default Schema:							
Default Catalog:							
User <u>n</u> ame:							
Password:							
Create a console configuration							
?		< Back	Next >	Cancel	Finish		

- 8. Click the down-arrow in the **Database dialect** field to select the database.
- 9. Click the down-arrow in the **Driver class** field to select the driver.
- 10. Click the down-arrow in the **Connection URL** field to select the URL.
- 11. Click Finish.

## 7.6. CREATING A HIBERNATE CONSOLE CONFIGURATION FILE

A console configuration file describes how the Hibernate plugin configures Hibernate. It also describes the configuration files and classpaths needed to load the POJOs, JDBC drivers, and so on. It is required to make use of query prototyping, reverse engineering and code generation. You can have multiple console configurations per project, however, one configuration is sufficient.

The following section describes how to create a Hibernate console configuration file in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Press Ctrl+N.

The **Select a wizard** window appears.

		New		×
Select a wizard				-
Create a new Hibern	ate Console Confi	guration		
Wizards:				
hibernate				×
🖏 Hibernate Co	onfiguration File (	(cfg.xml)		
S Hibernate C	onsole Configurat	tion		
S Hibernate R	everse Engineerii	ng File (reveng.xm	l)	
S Hibernate X	ML Mapping file (	hbm.xml)		
Show All Wizards	. <b>S</b>			
(?)	< Back	Next >	Cancel	Finish
•	Duck		cuncer	

- 3. Enter **Hibernate** in the search field.
- 4. Select Hibernate Console Configuration.

#### 5. Click Next.

The Create Hibernate Console Configuration window appears.

Unable to create requested service [or	g.hibernate.engine.jd	lbc.env.spi.JdbcEr	wironment]
ame: my-new-console-config			
🕽 Main 🛛 🔲 Options 😽 Classpath 🖸	Mappings 🔲 Comn	non	
ype: O Core O Annotations (jdk 1.5+) O Hibernate Version: 5.4 •	JPA (jdk 1.5+)		
Project:			
my-new-JPA-project			Browse
Database connection:			
[Hibernate configured connection]			▼ New Edit
Property file:			
			Setup
Configuration file:			
			Setup
Persistence unit:			Browse

- 6. Name your configuration file .
- 7. Ensure that the **Type** is set to **Core**.
- 8. Select the correct Hibernate version.
- 9. Click **Browse** to locate your project.
- Click New to configure a new Database connection.
   The New Connection Profile window appears.
- 11. Select the **Database Connection** or create a new one.
- 12. Click **Setup** to set up the **Property file**. The **Setup property file** window appears.
- Click Create new.
   The Create Hibernate Properties file (.properties) window appears.

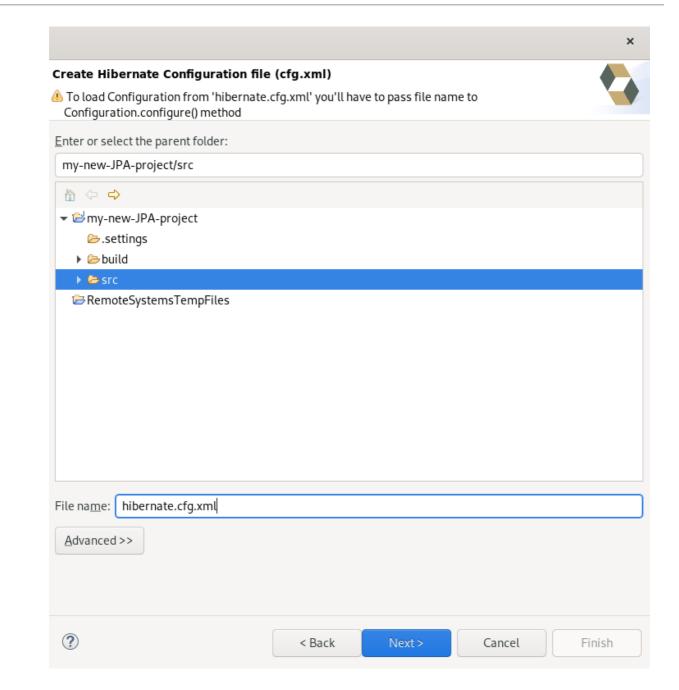


### Create Hibernate Properties file (.properties)

#### Create a new properties file

Enter or select the parent folder:		
my-new-JPA-project		
▶ ể my-new-JPA-project		
⇐ RemoteSystemsTempFiles		
File name: hibernate.properties		
Advanced >>		
?	Cancel	Finish

- 14. Select the parent directory.
- 15. Name your **.properties** file.
- 16. Click Finish.
- 17. Click **Setup** to set up the **Configuration file**.
- Select the path to the target .cfg.xml file.
   The Setup configuration file window appears.
- 19. Click Create new.The Create Hibernate Configuration file (cfg.xml) window appears.



- 20. Select the parent directory.
- 21. Click Next.

The Hibernate Configuration File (cfg.xml) window appears.

_	Hibernate Configuration File (cfg.xml)							
This wizard creates a new configuration file to use with Hibernate.								
<u>C</u> ontainer:	/my-new-JPA-project/src							
File name:	hibernate.cfg.xml							
Hibernate version:	5.4 🔻							
Session factory name:								
Get values from Conne	ection							
<u>D</u> atabase dialect:	MySQL				•			
D <u>r</u> iver class:	org.gjt.mm.mysql.Driver							
Connection <u>U</u> RL:	jdbc:mysql:// <hostname>/<database></database></hostname>							
Default Schema:								
Default Catalog:								
User <u>n</u> ame:	name:							
Password:	Password:							
Create a console configuration								
?		< Back	Next >	Cancel	Finish			

- 22. Click the down-arrow in the **Database dialect** field to select the database.
- 23. Click the down-arrow in the **Driver class** field to select the driver.
- 24. Click the down-arrow in the **Connection URL** field to select the URL.
- 25. Click **Finish**. The **Create Hibernate Console Configuration** window appears.

ame: my-new-console-config		
Main 🔲 Options 🍫 Classpath	• Mappings 🔲 Common	
Гуре:		
O Core O Annotations (jdk 1.5+)	○ JPA (jdk 1.5+)	
Hibernate Version: 5.4 💌		
Project:		
my-new-JPA-project		Browse
Database connection:		
sakila		▼ New Edit
Property file:		
/my-new-JPA-project/hibernate.p	roperties	Setup
Configuration file:		
/my-new-JPA-project/src/hiberna	te.cfg.xml	Setup
Persistence unit:		
		Browse

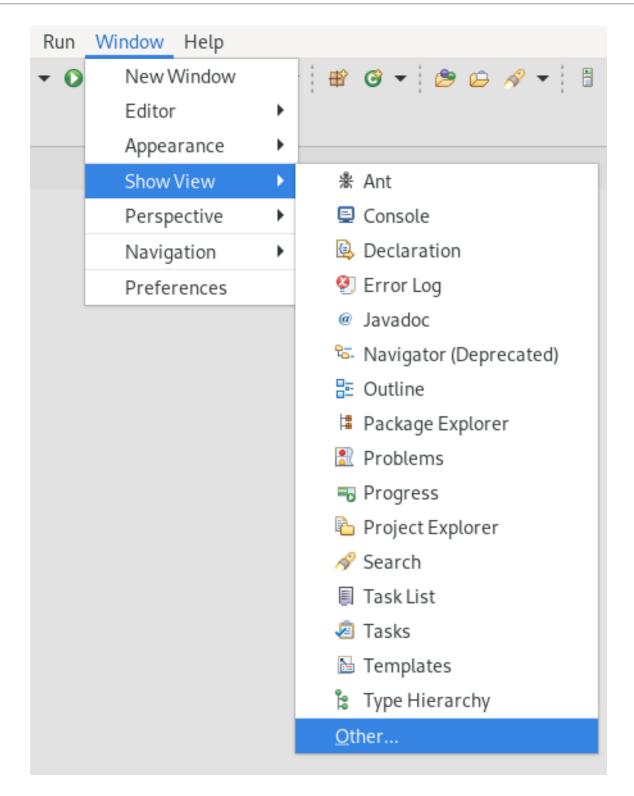
- 26. Set the database connection to **sakila**.
- 27. Click Finish.

# 7.7. EDITING HIBERNATE PROJECT CONFIGURATIONS

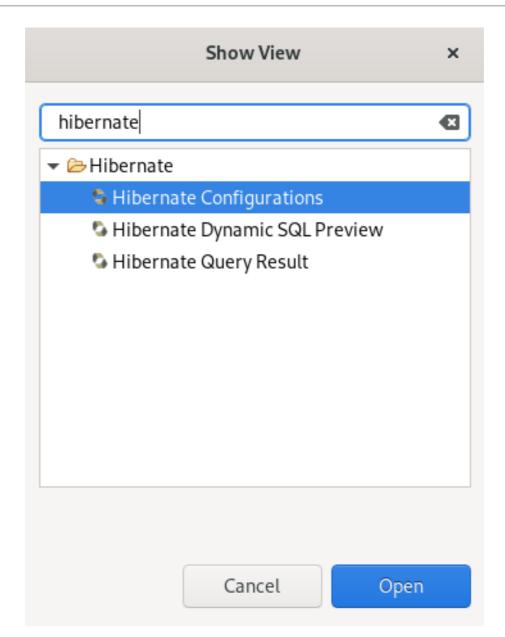
The following section describes how to edit configurations for your Hibernate project in CodeReady Studio.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click Window  $\rightarrow$  Show View  $\rightarrow$  Other.



The **Show View** window appears.



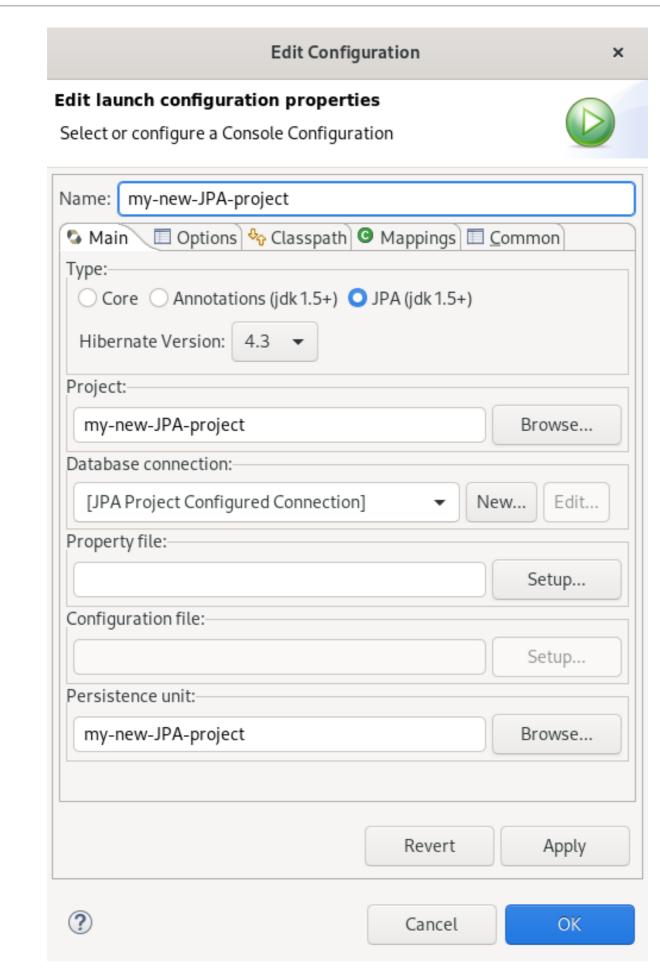
- 3. Enter **Hibernate** in the search field.
- 4. Select Hibernate Configurations.

#### 5. Click **Open**.

The Hibernate Configurations view appears.

S Hibernate Configurations	2
🕨 🗟 my-new-console-config	
▶ 🗟 my-new-JPA-project	<ul> <li>HQL Editor</li> <li>Hibernate Criteria Editor</li> <li>Mapping Diagram</li> <li>Add Configuration</li> <li>Rebuild configuration</li> </ul>
	Edit Configuration
	🗱 Close Configuration
	Delete Configuration
	Refresh Run SchemaExport

Right-click your project → Edit Configuration.
 The Edit Configuration window appears.



- 7. Edit your configurations.
- 8. Click Apply.

9. Click **OK**.

# CHAPTER 8. MOBILE WEB TOOLS BASICS IN CODEREADY STUDIO

Mobile Web Tools provide an **HTML5 Project** wizard that enables you to create web applications optimized for mobile devices. The **HTML5 Project** wizard is a useful starting point for creating all new HTML5 web applications in CodeReady Studio. The wizard generates a sample ready-to-deploy HTML5 mobile application with REST resources from a Maven archetype.

You can customize the application using the built-in editor, and deploy and view the application with the built-in browser.

CodeReady Studio provides the **Mobile Web** palette that allows the user to make interactive web applications. This palette offers a wide range of features including drag-and-drop widgets for adding common web interface framework features such as HTML5, jQuery Mobile, and Ionic tags to html files. It also contains widgets like **Panels**, **Pages**, **Lists**, and **Buttons** to make your applications more user friendly and efficient.

# 8.1. CREATING AN HTML5 PROJECT

The **HTML5 Project** wizard generates a sample project based on a Maven archetype and the project and application identifiers provided by you. The Maven archetype version is indicated in the **Description** field on the first page of the wizard. You can change the version, and therefore the project look and dependencies, by selecting either an enterprise or non-enterprise target runtime within the wizard.

The following section describes how to create an HTML5 project in CodeReady Studio.

#### Prerequisites

 A configured server.
 For information on configuring a local runtime server and deploying applications to it, see Section 3.1, "Configuring a local server".

CodeReady Studio must be configured for any servers to which you want to deploy your application, including the location and type of the application server and any custom configuration or management settings.

#### Procedure

- 1. Start CodeReady Studio.
- Press Ctrl+N. The Select a wizard window appears.

		New		×
Select a wizard				Ď
Wizards:				
html5				₿
🔫 🗁 Red Hat Central				
명 HTML5 Proje	ct			
Show All Wizards.				
?	< Back	Next >	Cancel	Finish

- 3. Enter **HTML5** in the search field.
- 4. Select HTML5 Project.
- 5. Click Next.

The New Project Example window appears.

	New Project Example	
HTML5 Proj	ject	1
Create a Jav	a EE 6 HTML5 Mobile web application project	Z
Description:		
	e that generates a Java EE 6 HTML5 Mobile Webapp project f	or JBoss Enterpri
Application F	Platform 6 or JBoss Application Server 7.1	
	ed on the org.wildfly.archetype:wildfly-html5-mobile-archety	pe:8.2.0.Final Ma
archetype		
Create a	blank project	
Target Runt	time Red Hat JBoss EAP 7.3 Runtime	
Requirement	ts and Recommendations	
Туре	Description Sta	Install
	itime Requires JBoss EAP 6, JBoss AS 7.1 or WildFly 🛕	Instatt
	Requires m2e >= 1.0.	wnload and Insta
plugin		
	•	
plugin plugin	Requires m2e-wtp >= 0.16.0.	
plugin plugin	Requires m2e-wtp >= 0.16.0.	
plugin plugin	Requires m2e-wtp >= 0.16.0.	
plugin plugin	Requires m2e-wtp >= 0.16.0.	Finish

- 6. Click the down-arrow in the Target Runtime field.
- 7. Select your server.
- 8. Click Next.
- 9. Name your project and your package.
- 10. Select the location for your project.
- Click **Finish**.
   Note that it might take some time for the project to generate.

The **New Project Example** window appears.

12. Click Finish.

Your newly created project is now listed in the **Project Explorer** view.

## 8.2. ADDING A NEW HTML5 JQUERY MOBILE FILE

The HTML5 **jQuery Mobile** file template consists of JavaScript and CSS library references that are inserted in the file's HTML header. The template also inserts a skeleton of the **jQuery Mobile** page and **listview** widgets in the file's HTML body.

The following section describes how to add a new HTML5 jQuery Mobile file to an existing project.

#### Prerequisites

• A configured server. For information on configuring a local runtime server and deploying applications to it, see Section 3.1, "Configuring a local server".

CodeReady Studio must be configured for any servers to which you want to deploy your application, including the location and type of the application server and any custom configuration or management settings.

#### Procedure

- 1. Start CodeReady Studio.
- 2. Press Ctrl+N.

The **Select a wizard** window appears.

		New			×
Select a wizard					-
Create a new HTML file					
Wizards:					
html					Ø
← ➢ Red Hat Central					
😴 HTML5 Project 👻 🗁 Web					
🚰 HTML File					
Show All Wizards.					
?	< Back	Next	t>	Cancel	Finish

3. Enter **HTML** in the search field.

- 4. Select HTML File.
- 5. Click Next.

The New HTML File window appears.

	Ν	lew HTML File		×
HTML				
Create a new HTML fi	le.			~
Enter or select the par	rent folder:			
jboss-as-kitchensink	-html5-mobile/s	rc/main/webapp		
☆ ⇔				
🕶 🔛 jboss-as-kitchei	nsink-html5-mob	ile		
🗁 .settings				
🔻 🗁 src				
🔻 🗁 main				
🕨 🗁 java				
🕨 🗁 resourc	es			
🕨 🗁 webapp	)			
▶ 🗁 test				
▶ 🗁 target				
🗁 RemoteSystems	sTempFiles			
File name: NewFile.	html			
Advanced >>				
?	< Back	Next >	Cancel	Finish

- 6. Select the location for your file.
- 7. Name your file.
- 8. Click Next.

The **Select HTML Template** window appears.

Select a template as initial content in the HTML page.						
✔ <u>U</u> se HTML Template <u>T</u> emplates:						
Name		Description				
Facelets XHTML Page		Facelets XHTML Page Template				
HTML5 jQuery Mobile Pag	je (1.3)	HTML5 jQuery Mobile 1.3 Template				
HTML5 jQuery Mobile Pag	je (1.4)	HTML5 jQuery Mobile 1.4 Template				
New Facelet Template New HTML File (4.01 fram New HTML File (4.01 stric New HTML File (4.01 trans New HTML File (5) New XHTML File (1.0 fram New XHTML File (1.0 stric New XHTML File (1.0 trans	t) sitional) eset) t)	Creates a header for use with the Facelet ten Creates a basic header/content/footer Facele html 4.01 frameset html 4.01 strict html 4.01 transitional html 5 xhtml 1.0 frameset xhtml 1.0 strict xhtml 1.0 transitional				
Preview:						
<pre>- <!DOCTYPE html>     <html> <head>     <title>jQuery Mob     &lt;meta http-equiv=     &lt;meta name="viewp     content="widt&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;"Content-Type"&lt;br&gt;ort"&lt;br&gt;b-dowico width&lt;/td&gt;&lt;td&gt;/title&gt; " content="text/html; charset=utf-8" binitial_ccale=1"/&gt; ates found in the &lt;u&gt;HTML Templates&lt;/u&gt; preference&lt;/td&gt;&lt;td&gt;&lt;/td&gt;&lt;/tr&gt;&lt;/tbody&gt;&lt;/table&gt;</title></head></html></pre>						

- 9. Select a template.
- 10. Click Finish.

The newly created HTML file is now displayed in the CodeReady Studio editor.

## 8.3. ADDING A NEW MOBILE PAGE

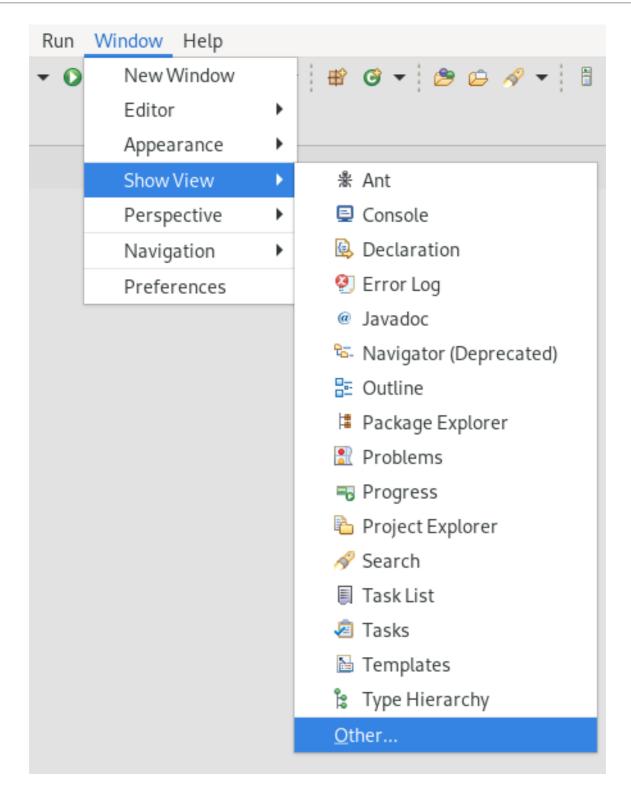
The following section describes how to add a new jQuery Mobile Page to an existing web application.

#### Prerequisites

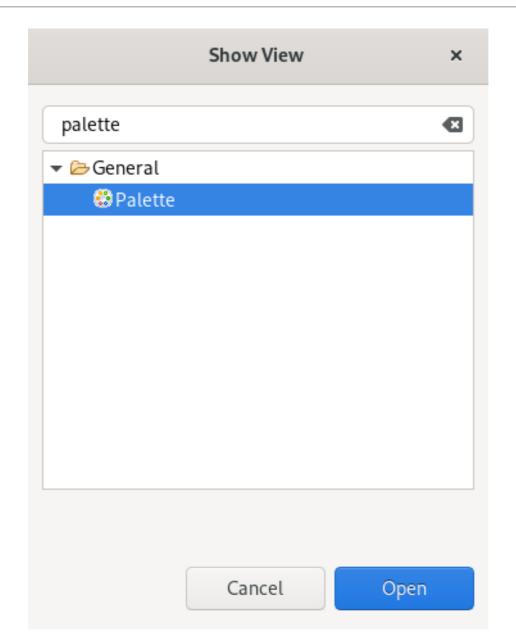
An HTML5 project.
 For more information on how to create an HTML5 project, see Section 8.1, "Creating an HTML5 Project".

#### Procedure

- 1. Start CodeReady Studio.
- 2. Click Window → Show view → Other.



The **Show View** window appears.



- 3. Enter **Palette** in the search field.
- 4. Select Palette.

## 5. Click **Open**.

The **Palette** view appears.

<b>\</b> type filt	er text										
6										25 🔻	Most popular 🔻
🗖 jQuery M	1obile										1.4 🔻
JS/CSS	Page	Dialog	Popup	Grid	Panel	Table	Collapsible	Collapsible Set	Tabs	Heading	
Header Bar	Footer Bar	Navbar	Select	Listview							
Button	Buttons	Submit Form Button	Link	On Toggle	Radio	Checkbox	Checkboxes	Slider	Text_ Text Input		

6. Click the **Page** icon.

#### The Insert Tag window appears.

		Insert Tag	×				
<b>ew Page</b> Create a new	jQuery Mobile page widget.	ĺ					
✓ Header	Page Title	<pre><div data-role="page" id="page-1">     <div data-role="header"></div></div></pre>					
Veric Footer	Page Footer Generate	 <div data-role="content"> Page content goes here.</div>					
Back Bu	tton:	<div data-role="footer"></div>					
Label:	Back	Page Title					
Icon: Icon only:		Page content goes here.					
ineme:	•	Page Footer					
🛃 Add refe	rences to JS/CSS Hide Preview	Preview may not support all available features					
?		< Back Next > Cancel Finish	h				

- 7. Name the **Header**.
- 8. Name the **Footer**.
- 9. Click Finish.

Your newly added page is now displayed in the CodeReady Studio editor.



#### NOTE

You can use the same workflow to customize the pages of your web application by selecting widgets from the **Palette** view.