



Red Hat Process Automation Manager 7.0

**Running the employee roster starter
application for Red Hat Business Optimizer on
Red Hat OpenShift Container Platform**

Red Hat Process Automation Manager 7.0 Running the employee roster starter application for Red Hat Business Optimizer on Red Hat OpenShift Container Platform

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Copyright © 2018 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, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This document describes how to run and use the OptaShift Employee Rostering example included as an add-on in Red Hat Process Automation Manager 7.0.

Table of Contents

PREFACE	3
CHAPTER 1. INSTALLING AND RUNNING THE EMPLOYEE ROSTERING DEMONSTRATION ON OPENSIFT	4
1.1. DEPLOYING USING THE PROVIDED SCRIPT	4
1.2. DEPLOYING USING THE SOURCE TO IMAGE (S2I) TEMPLATE	6
CHAPTER 2. OVERVIEW OF THE EMPLOYEE ROSTERING STARTER APPLICATION	9
2.1. ENTERING SOURCE INFORMATION FOR EMPLOYEE ROSTERING	9
2.2. CREATING AN OPTIMAL EMPLOYEE ROSTER	14
2.3. PUBLISHING THE EMPLOYEE ROSTER AND ENABLING GENERATION FOR SUBSEQUENT DATES	14
2.4. SETTING EMPLOYEE AVAILABILITY	15
2.5. CONSTRAINTS	16
APPENDIX A. VERSIONING INFORMATION	17

PREFACE

As a business rules developer, you can test and interact with the Red Hat Business Optimizer functionality by quickly deploying the **employee-rostering** starter project included in the Red Hat Process Automation Manager distribution to OpenShift.

Prerequisites

- You have downloaded the Red Hat Process Automation Manager add-ons distribution, which includes the OptaShift Employee Rostering starter application.
- You have access to a deployed OpenShift environment. For details, see the documentation for the OpenShift product that you use.

CHAPTER 1. INSTALLING AND RUNNING THE EMPLOYEE ROSTERING DEMONSTRATION ON OPENSIFT

You can deploy the Red Hat Business Optimizer Employee Rostering starter application to an OpenShift instance.

Preparing deployment files

1. Unzip the add-ons distribution (**rhcam-7.0.0-add-ons.zip**).
2. Unzip the employee rostering zip file (**rhcam-7.0-employee-rostering.zip**) that is extracted from the add-ons archive.



NOTE

When you unzip the employee rostering zip file, the **optashift-employee-rostering-7.7.0.Final-redhat-7** folder is created. This folder is the base folder in subsequent steps. File and folder names may have higher version numbers than specifically noted in this document.

There are two options for quickly deploying this application to OpenShift:

1. Using the **provision.sh** shell script that is provided in the add-ons distribution. In this case, the application source code is built and packaged locally and uploaded to the OpenShift environment for deployment. Use this preferred method when Java developer tools (Java Development Kit and Maven) and a bash shell command line are available.
2. Using the provided Source to Image (S2I) template. This template pulls the source for the application from a Git repository and builds it during the deployment. In this case, the compilation and packaging of the application occurs within the OpenShift environment. Use this option when the prerequisites for using **provision.sh** are not available.



NOTE

The S2I approach requires significantly more OpenShift resources to complete than the shell script deployment. If your OpenShift environment has less than 2GB of memory available, you must follow the shell script deployment approach. The application will not deploy successfully with any method to the OpenShift Online Starter tier at the time of this writing because of memory requirements.

1.1. DEPLOYING USING THE PROVIDED SCRIPT

Prerequisite

- You have logged in to the target OpenShift environment using the **oc** command line tool.

Procedure

1. Running the script
 - a. Using the command line, change to the **optashift-employee-rostering-7.7.0.Final-redhat-7/sources** folder.

- b. Run the provision script to build and deploy the application:

```
./provision.sh setup employee-rostering --binary
```



NOTE

If the current OpenShift user name contains characters that are not letters or numbers, the deployment might fail. This can be fixed by providing an additional parameter: **./provision.sh setup employee-rostering - -binary --project-suffix optashift**. This example uses "optashift", but the actual value has no impact on the deployment.

2. Verify the compilation status

- a. Compilation and packaging may take several minutes to complete, and should continually show progress on the command line output.

- b. Once completed, the following message is displayed: **Uploading file "target/R00T.war" as binary input for the build ...**

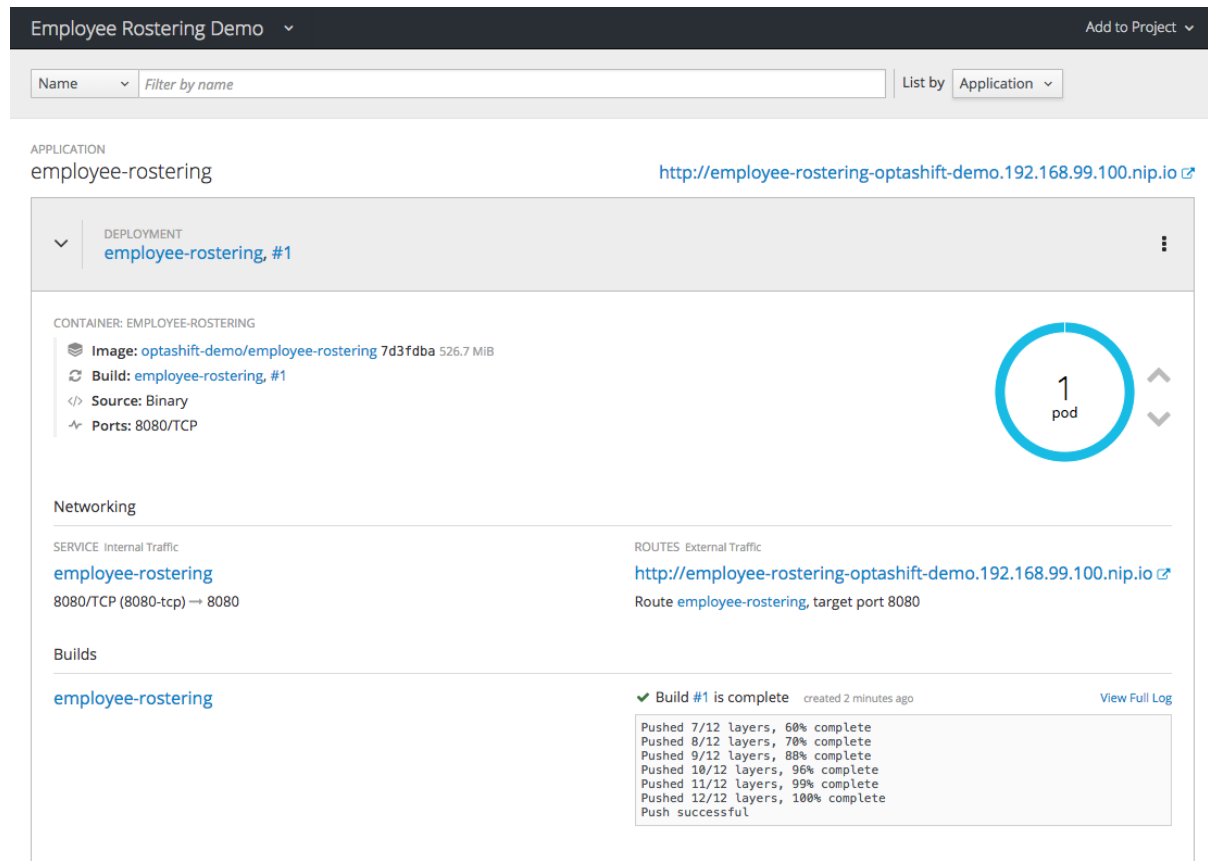
3. Use the OpenShift web UI to view the details for the deployed application. Click the link in the Routes section to open the starter application.

The build can take up to 30 seconds to complete. The application container can take an additional 30 seconds to start up completely.



NOTE

Perform a hard refresh of your browser page if the web app does not open after clicking the link.

Figure 1.1. OpenShift web console with deployed Employee Rostering starter application

4. Optionally, you can use other actions provided by the provision script:
 - a. Deploy updates to the code to an existing deployment: `./provision.sh deploy employee-rostering --binary`
 - b. Remove the application: `./provision.sh delete employee-rostering`

1.2. DEPLOYING USING THE SOURCE TO IMAGE (S2I) TEMPLATE

Procedure

1. Log in to the OpenShift web console and click **Import YAML/JSON** in the upper right corner.
2. Select the project to add to from the **Add to Project** drop down menu, or select **Create Project** to create a new one. If creating a new project, enter a new project name, display name, and description.
3. Click the **Browse** button and navigate to the **optashift-employee-rostering-template.yaml** file. Starting from the **optashift-employee-rostering-7.7.0.Final-redhat-7** folder, access the file by clicking **sources > openshift > templates**
4. Click **Create**.

Figure 1.2. Upload template window

Import YAML / JSON

YAML / JSON Results

1 2

Create or replace resources from their YAML or JSON definitions. If adding a template, you'll have the option to process the template.

optashift-employee-rostering-template.yaml Browse...

Upload a file by dragging & dropping, selecting it, or pasting from the clipboard.

```

1 apiVersion: v1
2 kind: Template
3 labels:
4   app: employee-rostering
5 metadata:
6   annotations:
7     description: OptaPlanner Employee Rostering on OpenShift
8     iconClass: icon-java
9     tags: optaplanner,jboss
10  name: employee-rostering-builds
11 objects:
12 - apiVersion: v1
13   kind: ImageStream
14   metadata:
15     name: employee-rostering
16   spec:
17     image:

```

Cancel < Back Create

5. In the Add Template dialog box, click **Continue** to accept the default settings.

Figure 1.3. Add template window

Add Template

What would you like to do?

☒ **Process the template**

Create the objects defined in the template. You will have an opportunity to fill in template parameters.

☐ **Save template**

Save the template to the project. This will make the template available to anyone who can view the project.

Continue Cancel

6. In the Import YAML/JSON dialog box, click **Create** to accept the default settings.

Figure 1.4. Configure template window

7. In the Success dialog box, click **Close**.
8. Wait for the build and deployment to complete. This might take several minutes. When the deployment completes, click the link in the upper right corner above the shaded bar to open the application.

Figure 1.5. Completed deployment and resulting link



NOTE

Perform a hard refresh of your browser page if the web app does not open after clicking the link.

CHAPTER 2. OVERVIEW OF THE EMPLOYEE ROSTERING STARTER APPLICATION

The employee rostering starter application assigns employees to shifts on various positions in an organization. For example, you can use the application to distribute shifts in a hospital between nurses, guard duty shifts across a number of locations, or shifts on an assembly line between workers.

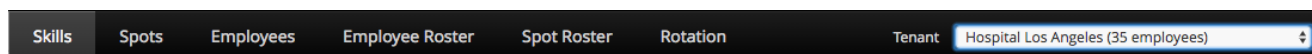
Optimal employee rostering must take a number of variables into account. For example, different skills can be necessary for shifts in different positions. Also, some employees might be unavailable for some time slots or might prefer a particular time slot. You can specify inputs for these variables using the tabs along the top of the application window.



IMPORTANT

The current version of the starter application is limited to working with generated test data. You can use it to evaluate the Red Hat Business Optimizer capabilities. It is not suitable for use in a production environment.

Figure 2.1. Employee rostering toolbar



2.1. ENTERING SOURCE INFORMATION FOR EMPLOYEE ROSTERING

In order to produce an optimal employee roster, a set of input data is required: skills, spots and employees. Additional data, such as employee availability and rotation orders, can be used to further optimize the generated roster output.

Tenants

In the **TENANT** field at the top right corner of the application window, you can select one of the predefined tenants. Each tenant represents an independent set of data (inputs and roster outputs); any change in the data for one tenant does not affect any other tenants.

You can switch between tenants to use several independent data sets, for example, to prepare employee rosters for different locations.

Skills

In the skills section, you can set all skills that can be necessary in any position within the roster. For example, a 24-hour diner would require cooking, serving, bussing, and hosting skills, in addition to skills such as general human resources and restaurant operations.

To enter these skills, click the **Skills** tab.

Figure 2.2. Initial skills list for the factory tenant

Skills

Search x

1-2 of 2

Skill Name
Electrical
Mechanical

Editing the skills list:

- To add a new skill:
 - Click **Add Skill**.
 - Type the name of the new skill into the text box that appears.
 - Click **Save Skill**.
- Edit skill names by clicking the **Edit Skill** icon (pencil shape).
- Delete skills by clicking the **Delete Skill** icon (trashcan shape).



NOTE

Within each tenant, skill names must be unique. Skills can not be deleted when they are associated with an employee or spot.

Spots

Spots refer to the various positions at the business. For a diner, spots include the bar, the bussing stations, the front counter, the various kitchen stations, the serving areas, and the office. For each spot, you can select one or more required skills from the list that you entered in the Skills tab. The following figure shows a sample list of spots and skills for the factory tenant example:

Figure 2.3. Excerpt of spot definitions for the factory tenant

Spots

Search x

1-10 of 10

Spot Name	Skill Set
Battery	Electrical
Bumper	Mechanical
Chassis	Mechanical, Electrical

Editing the spots list:

- To add a new spot:

- Click **Add Spot**.
- Type the name of the new spot into the text box that appears.
- Select one or more skills from the combo-drop down list that appears.
- Click **Save Spot**.
- Edit spot names and associated skills by clicking the **Edit Spot** icon (pencil shape).
- Delete spots by clicking the **Delete Spot** icon (trashcan shape).



NOTE

Within each tenant, spot names must be unique. Spots can not be deleted when they are associated with an employee roster.

Employees

The Employees tab contains a list of all people employed by the business and the skills they possess. The following figure contains an excerpt of the employees in the factory tenant example:

Figure 2.4. Excerpt of employees for factory tenant

Employees

Search <input type="text"/> x <input type="button" value="Q"/>	1-10 of 35 <input type="button" value="←"/> <input type="button" value="→"/>	<input type="button" value="Refresh"/> <input type="button" value="Add Employee"/>
Employee Name	Skill Set	
Amy Cole	Mechanical, Electrical	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Amy Fox	Mechanical, Electrical	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Amy Green	Mechanical	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Editing the employee list:

- To add a new employee:
 - Click **Add Employee**
 - Type the name of the new employee into the text box that appears.
 - Select one or more skills from the combo-drop down list that appears.
 - Click **Save Employee**.
- Edit employee names and associated skills by clicking the **Edit Employee** icon (pencil shape).
- Delete employees by clicking the **Delete Employee** icon (trashcan shape).



NOTE

Within each tenant, employee names must be unique. Employees can not be deleted when they are associated with an employee roster.

Spot Roster

The Spot Roster is a table of all shifts (time slots) during which the business is open and all spots that must be manned during each shift. Each shift is represented by a rectangle at the intersection of a spot (row) and time span (column).

Spots can support multiple employees per shift. If a spot is not required for a shift, you can remove all of the shifts completely from the roster for that time span.

Figure 2.5. Spot roster for factory tenant example

Spot Roster Publish Solve Refresh < >

	2018 ...	2018 April 22, Sunday					2018 April 23, Monday					2018 April 24, Tuesday					2018 April 25, Wednesd			
	20:00	04:00	08:00	12:00	16:00	20:00	04:00	08:00	12:00	16:00	20:00	04:00	08:00	12:00	16:00	20:00	04:00	08:00	12:00	16:00
Battery	Amy Fox	Ivy Smith	Beth Green	Amy Fox			Jay Watt	Beth Green	Amy Fox			Jay Watt	Beth Green	Amy Fox			Jay Watt	Ivy Smith		
Bumper	Amy Jo...	Ivy Cole	Beth King	Amy Jones			Jay Fox	Beth King	Amy Jones			Jay Fox	Beth King	Amy Jones			Jay Fox	Ivy Cole		

To view the spot roster for future days, use the horizontal scroll bar at the bottom of the window.

Double-click an open area in the schedule to add a new shift to the schedule or double-click an existing shift to edit it. The value for the spot type and time frame is set by the initial click and is not editable. By default, the employee is not editable either, as the planning engine determines the employee for every shift. A particular employee can be "pinned" to a shift by selecting the **Pinned** check box and selecting the employee from the **Employee** drop down.

Figure 2.6. Shift details for pinning and deleting

The screenshot shows a calendar interface with a modal dialog titled "Edit shift". The calendar background displays dates for 2018 April 23 (Monday) and 2018 April 24 (Tuesday). The dialog box contains the following fields and options:

- From:** Date: APRIL 22, Time: 22:00
- To:** Date: APRIL 23, Time: 06:00
- Spot:** Battery
- Employee:** Amy Fox
- Pinned:** ☒ Pinned
- Rotation employee:** -
- Buttons:** Delete shift, Cancel, Apply

Rotation Patterns

The **Rotation** tab allows the schedule administrator to specify the default shift patterns for new weeks, without adding shifts manually. You can also apply default employees to the shifts and drag and drop the shift borders in this tab to change the length of shifts. The default assignments are "low priority" and can be superseded by "higher priority" constraints such as employee availability and pinned assignments in the spot roster.

Figure 2.7. Rotation configuration for factory tenant example

Rotation [Save](#) [Refresh](#)

	Day 1										Day 2									
	04:00	08:00	12:00	16:00	20:00						04:00	08:00	12:00	16:00	20:00					
Battery	Jay Watt	Ivy Smith	Amy Fox	Jay Watt	Ivy Smith	Amy Fox	Jay Watt	Ivy Smith	Amy Fox	Jay Watt	Ivy Smith	Amy Fox	Jay Watt	Ivy Smith	Amy Fox	Jay Watt	Ivy Smith	Amy Fox	Jay Watt	Ivy Smith
Bumper	Jay Fox	Ivy Cole	Amy Jones	Jay Fox	Ivy Cole	Amy Jones	Jay Fox	Ivy Cole	Amy Jones	Jay Fox	Ivy Cole	Amy Jones	Jay Fox	Ivy Cole	Amy Jones	Jay Fox	Ivy Cole	Amy Jones	Jay Fox	Ivy Cole
Chassis	Beth Fox	Amy Cole	Chad Green	Beth Fox	Amy Cole	Chad Green	Beth Fox	Amy Cole	Chad Green	Beth Fox	Amy Cole	Chad Green	Beth Fox	Amy Cole	Chad Green	Beth Fox	Amy Cole	Chad Green	Beth Fox	Amy Cole
Doors	Flo Rye	Elsa Poe	Gus Smith	Flo Rye	Elsa Poe	Gus Smith	Flo Rye	Elsa Poe	Gus Smith	Flo Rye	Elsa Poe	Gus Smith	Flo Rye	Elsa Poe	Gus Smith	Flo Rye	Elsa Poe	Gus Smith	Flo Rye	Elsa Poe
Engine	Flo Li	Elsa King	Gus Poe	Flo Li	Elsa King	Gus Poe	Flo Li	Elsa King	Gus Poe	Flo Li	Elsa King	Gus Poe	Flo Li	Elsa King	Gus Poe	Flo Li	Elsa King	Gus Poe	Flo Li	Elsa King
Lights	Beth Jones	Amy Green	Chad King	Beth Jones	Amy Green	Chad King	Beth Jones	Amy Green	Chad King	Beth Jones	Amy Green	Chad King	Beth Jones	Amy Green	Chad King	Beth Jones	Amy Green	Chad King	Beth Jones	Amy Green

**NOTE**

In the current version, the rotation data is essentially pre-set in the example data sets. In a subsequent version, these rotation preferences will be fully editable from the user interface.

2.2. CREATING AN OPTIMAL EMPLOYEE ROSTER

Procedure

1. Adjust the skills, employees, and spot types.
2. Move to the **Spot Roster** tab:
 - a. Adjust the pre-populated shifts by adding/removing as necessary. (This change might not work in the current version of the application.)
 - b. Pin specific employees to shifts as necessary.
 - c. Click the Solve button to create an optimized solution.

The engine takes up to 30 seconds to create a fully optimal result.

When the optimal result is available, the Spot Roster displays the name of the assigned employee for each spot and shift.

2.3. PUBLISHING THE EMPLOYEE ROSTER AND ENABLING GENERATION FOR SUBSEQUENT DATES

After generating an employee roster for a period of time, you can "publish" the roster. Publishing means that the roster can no longer be changed in the automatic rostering process; therefore, you can safely distribute rostering information to employees.

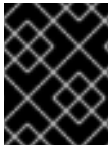
At the time you publish a roster, a new period (normally one week) is added to the **Spot Roster** tab. When you click the **Solve** button, this new period is filled in, but the roster that was already published is not changed.

Procedure

1. Move to the **Spot Roster** tab:

2. Ensure that the displayed roster is fully acceptable.

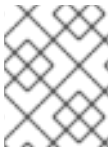
- Click the **Publish** button at the top right corner



IMPORTANT

In the generated test data, the first three weeks are already solved and published. You can solve the rostering for the subsequent period and then press the **Publish** button.

2.4. SETTING EMPLOYEE AVAILABILITY



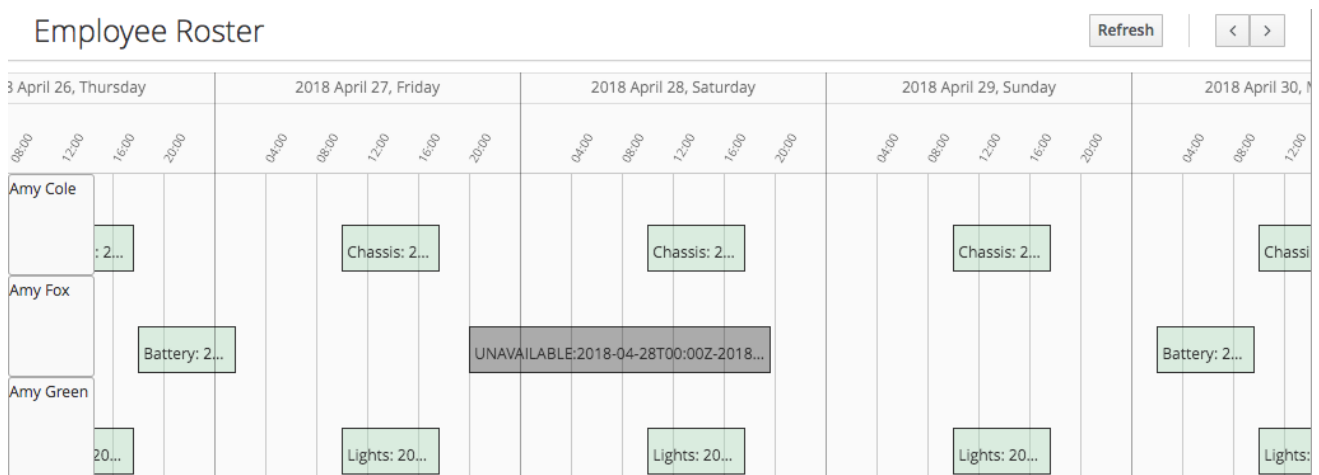
NOTE

This section is not fully functional in the current release and will be enabled in a subsequent release.

Employee Roster

The employee roster tab displays an employee-centric view of the schedule with a row for each employee depicting their assigned shifts.

Figure 2.8. Excerpt of employee roster for factory tenant



You can also use the employee roster view to specify the availability for particular employees:

- Double click on the schedule within the row of an employee to open the **Edit availability** pop-up.
- Select the availability preference (unavailable, undesired, desired). If an employee is unavailable for a particular shift, the employee can never be assigned to this shift. "Undesired" and "desired" are preferences that the rostering engine accommodates when possible.
- Click **Apply** to save.

Figure 2.9. Employee availability preferences example

UNAVAILABLE:2018-04-23T00:00Z-2018...

tery: 2... Batt

mp: 2... Bum

Chasie: 2

U nassie: 2 U nassie: 2

Edit availability ✕

Day **From**

To

Employee

Availability

- ✓ UNAVAILABLE
- UNDESIRE
- DESIRED

[Delete availability](#)

After making changes to employee availability, click the **Solve** button on the Spot Roster tab to update the schedule.

**NOTE**

The Unavailable option also applies for employees who call in sick or are on vacation. If an employee calls out just before a shift, a manager can indicate the scheduled employee's unavailability and then create a new schedule.

2.5. CONSTRAINTS

The Red Hat Business Optimizer rules for this starter application use both hard and soft constraints. During an optimization, the engine may not violate hard constraints, for example, if an employee is unavailable (out sick), or that an employee cannot work two spots in a single shift. The engine tries to adhere to soft constraints, such as an employee's preference to not work a specific shift, but can violate them if the optimal solution requires it.

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Monday, October 1, 2018.