



Develop, Test and Deploy Java Applications on OCI – Project

Project Guide – Requirements

S1102203GC10

Learn more from Oracle University at education.oracle.com

O

Copyright © 2021, Oracle and/or its affiliates.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

This documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

1003252021

Table of Contents

Java SE on OCI SLS Project: Requirements Document.....	5
Project Details.....	6

Neela Priyadarshini Venkatachalam (priyavenky1811@gmail.com)
has a non-transferable license to use this Guide.

Neela Priyadarshini Venkatachalam (priyavenny1811@gmail.com)
has a non-transferable license to use this Guide.

Java SE on OCI SLS Project: Requirements Document

Project Details

Overview

You will create an application to process and validate warranty claims. Set up a database of existing warranties. Read in list of warranty claims. If the claim is valid, store it in the database. If it's not valid, log this information in a local file.

Objectives:

- Analyze requirements and develop the design for a Java integration application.
- Open, read and print files.
- Connect to and read/write to an Autonomous Database.
- Implement application logic to check record formats and check database records for validity.
- Connect to and capture data from a REST service.

Background:

- Your company is looking to create a warranty claim application that will process new warranty claims against existing warranty records.
- The application should check that the submitted batch of claim records meet the required format. Wrong format records are to be logged.
- Correctly formatted claim records should be validated against a Warranty table.
- Claims with a valid warranty are augmented with additional warranty details read from a database table and a REST service that has been provided for you.
- These augmented claim records are stored in the database with status 'new'.
- You will provision an OCI VM to host your code and Autonomous database to store the warranties that the claims will be processed against.
- Review the technical requirement listed below and get started.

Duration: 8 hours

Requirements

- Create an application to process and validate warranty claims.
- Set up a database of existing warranties.
- Create a CSV file of warranty claims to be processed.
- Read in list of warranty claims.
- If the claim is valid, store it in the database.
- If it's not valid, log this information in a local file.
- Review the next section for details on tasks to be performed to create this application.

Tasks to be performed

It is recommended that you work iteratively and incrementally. Write code for one logical section of the program. Test that your code works before moving to the next section.

Overview

- Provision and set up the necessary OCI assets and connections.
- Create the project in an IDE like NetBeans.
- Create a class to read the input file and write to the output files.
- Create classes to connect to the database, to perform queries to the database, and write database records.
- Create a class to make the REST calls and return the result.
- Create a class to load a properties file with the application configuration.
 - Creating a properties file is a good practice to store and isolate configurable data for your application, such as input and output directories, data format, file names, database credentials, and the REST endpoint URL.
- Create a class with a `main()` method that will bring the other classes together and provide the logic for the application.

Provision OCI Resources

1. You'll need an Oracle Cloud account to complete this lab. Instructions to request an account can be found below.in the section titled: **Instructions to Request Lab Environment.**
2. You may create an SSH key pair if you don't already have one. You'll need this later to provision your compute instance.

ssh-key-creation.pdf

3. Login to your Oracle University provided Cloud account.
4. Create a VM Compute instance and connect to the VM. The rest of your work on this project can be completed in the VM and Oracle Cloud.

vm-dev-provisioning.pdf

5. Install NetBeans or an IDE of your choice.

netbeans-vm-install.pdf

6. From the VM, log in to your Oracle University provided Cloud account.
7. Provision an Autonomous Transaction Processing (ATP) database instance.

atp-provisioning.pdf

8. Download the ATP wallet and connect with SQL Developer.

atp-wallet.pdf

9. Create a new database user. You'll need this user's credentials later within the properties file that stores your application configuration. You'll also run a few SQL commands, which prepare database tables so you can work with warranties and claims.

sql-scripts.pdf

Read a CSV File with Claim Records

CSV stands for Comma Separated Values. This file is used to batch process many warranty claims. You'll need to create and test a Java class that reads the CSV input file and writes to the output files in the following manner:

1. Create a CSV file. This has a .csv file extension. You will make up your own test data. Some records should be invalid and some should be valid.
2. Code a filter in Java to parse claim record objects read from the CSV file.
3. Produce a local file to log any claims that are incorrectly formatted.

The values in the CSV reflect the SQL you executed earlier to set up the database table for claims. Not all values are represented in the CSV file. The remaining values are supplied later as each warranty claim is validated. Values in the CSV file include:

- customer_id
- customer_firstname
- customer_lastname
- customer_email
- product_id
- product_name
- serial_number
- claim_date
- subject
- summary

This is the header for the CSV file:

```
customer_id, customer_firstname, customer_lastname, customer_email,
product_id, product_name, serial_number, claim_date, subject, summary
```

This is an example of an entry. You'll need to add more to the CSV file:

```
1896, John.Doe, john.d@company.com, 100, iPhone X, PVQXT01-
A20015R, 06/11/2020, Broken display, Customer noticed the display was
broken after unpacking the product
```

4. Specify properties for CSV batch processing within a properties file in your project, for example, the location of the CSV file to read and its data format:

```
#BATCH processing
#Batch CSV input file with claims

#Data format from CSV - expecting 10 fields, where productId is a
#Number, and claim date has a short date format
#-! spaces matter when using message format patterns in Java
```

- Also, in the properties file, specify the output location for claims that are incorrectly formatted. You may wish to include a file name prefix to identify the output file.

```
#Output location for claims records with bad format in CSV
```

Validate Claim Objects against a Warranty Table

- Create and test a class or a series of classes that connect to the database and read and write database records in the following manner:
 - Query against the warranty `product_id` and `serial_number`. Both fields must match the claim.
 - If a warranty exists for a claim, validate the warranty is not expired and attach the valid `warranty_number` to the claim.
 - Produce a local file to log any claims that do not have a warranty, have an invalid warranty, or an expired warranty.
- In the properties file, specify the wallet and JAVA_DEV user credentials. Use this example to reverse-engineer what you need:

```
#DATABASE interaction
#ATP details including ATP wallet location
#ATP Wallet can be pulled from resources or from any other path such as
mounted file systems
app.db.url=jdbc:oracle:thin:@<tns-alias>?TNS_ADMIN=<wallet-
location>/<wallet-directory>
#connection string sample, ofscd20200526_medium is the TNS alias
#app.db.url=jdbc:oracle:thin:@ofscd20200526_medium?TNS_ADMIN=resources/Wa
llet_OFSCD20200526

app.db.username=JAVA_DEV
app.db.password=Developer2020
```

- `<tns-alias>` is within the `tnsnames.ora` of your unzipped ATP wallet, which usually includes different aliases such as low, medium, among others
- `<wallet-location>` is the path to your wallet directory
- `<wallet-directory>` is the directory name of the unzipped wallet

Note: To establish a connection to the ATP database using the previous connection string, you need to download and set up the `ojdbc10` full library in your Java project and use the connection factory class name: `oracle.jdbc.pool.OracleDataSource`. The detailed steps are in the **add_jdbc_libs_to_netbeans.pdf** file.

3. The warranty values you should anticipate can be derived from the SQL command you made earlier that set up the database table and created records for warranties in the database. Values include:
 - product_id
 - serial_number
 - warranty_number
 - date_opened
 - expiry_date

4. In the properties file, specify the output location for the log that details claims without a warranty, an invalid warranty, or an expired warranty. You may wish to include a file name prefix to identify the output file.

```
#Output location for claims records with an invalid warranty
```

Augment Claims that have a Valid Warranty with REST data

1. Create and test a class to make the REST call and return the result:
 - a. Call a REST endpoint to get additional data for a claim, such as the country_code and country_region. Consider this a black-box REST API service that your company maintains to generate the proper codes for staff. This API reviews information about the product and produces appropriate results for you.
 - b. Add this data to each claim.
 - c. In the properties file, specify the product manufacturing API:

```
#REST API interaction
#The following API provides information such as the country code,
region, sub-region, among other data
app.api.endpoint=https://restcountries.eu/rest/v2/alpha/usa
```

- d. The REST endpoint returns data in JSON format. You'll need to deserialize the JSON data into a Java object to extract the country code and country region.

Note: To create the REST client, you need to set up the proper libraries such as JAX-RS and JAX-WS within your Java project.

Store Augmented Claims in the Database

1. Until this point, claim objects have only existed in Java code in memory. Now you must store them in the database. You should already have a database connection, or a connection pool that you set up earlier by using the ATP wallet:
 - a. Add the claims to the database.
 - b. Set their STATUS to new.
 - c. Set their CLAIM_ID. You may do this using the SQL CLAIM_SQL sequence you created earlier.

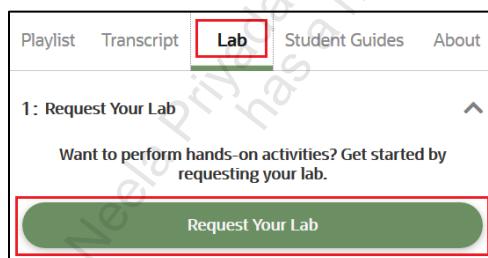
Required Project End State

- Expected output:
 - A file with invalid format records
 - A file with invalid warranty records
 - Claim records with valid warranty in ATP database (if any)
 - At the end, the Claim record in the database should have information from the CSV initial record, warranty, and data retrieved from the REST endpoint.
- The following image represents the processed claim record within the database. Note that this is the sample CSV record from earlier sections but it now includes augmented data such as the warranty number, country code, and country region:

CLAIM_ID	CU...	CU...	CU...	CUSTOMER_EMAIL	PR...	PRODUC...	SERIAL_NUMBER	WARRANTY_NUMBER	COUNTRY_CODE	COUNTRY_REGION	CLAIM_DATE	STATUS	SUBJECT	SUMMARY
50	1896	John	Doe	john.d@company.com	100	iPhone X	PVQXT01-A20015R	129850	USA	Americas	11/06/20	NEW	Broken...	Customer notice...

Instructions to Request Lab Environment

1. Request for a lab environment, i.e. OCI account using the course **Lab** tab.



You will get an email like the one below, confirming that the lab request has been placed. You will receive another email when the lab account is ready.

The image shows an email confirmation for a digital lab booking. The header features the Oracle University logo and a background photo of a person using a tablet. The main content area starts with a greeting "Steve," followed by a message stating "Your lab request has been received." Below this, it specifies the course as "Oracle Cloud Infrastructure Architect Associate Workshop (Sales Promotion)" and the booking period as "Lab Booked From: 07-Jan-21 to 14-Jan-21". A note indicates that an email will be sent when the lab environment is ready. At the bottom, there is a link to "Open a support ticket" and the Oracle University footer with copyright information.

Steve,

Your lab request has been received.

Course: Oracle Cloud Infrastructure Architect Associate Workshop (Sales Promotion)
Lab Booked From: 07-Jan-21 to 14-Jan-21

You will receive an email when the lab environment is ready.

For any lab related issues, [Open a support ticket](#)

ORACLE
UNIVERSITY

© 2019 Oracle University
[Your Privacy Rights](#) | [Terms and Conditions](#)

2. You will get a second email like the one below, once the lab account is ready.
Note: This is only a confirmation email to let you know that your lab environment is ready. This email will not have the account information. You will need to go back to the Lab Tab of your Subscription Course and refresh your page to see the lab account details.

The image shows a second email confirmation for a digital lab booking. It follows a similar structure to the first, with the Oracle University logo and a background photo of a person using a tablet. The greeting is "Steve," and the message informs him that his lab environment is now ready. It reiterates the course name and booking period. A note about instructions for accessing the lab is included. The footer contains a support link and the standard Oracle University copyright information.

Steve,

This is to inform you that your Lab environment is now ready.

Course: Oracle Cloud Infrastructure Architect Associate Workshop (Sales Promotion)
Lab Booked From: 07-Jan-21 to 14-Jan-21

Instructions to access your Lab are now available. Navigate to your Course in the Subscription, and view these instructions under Lab --> Access Your Lab Section.

For any lab related issues, [Open a support ticket](#)

ORACLE
UNIVERSITY

© 2019 Oracle University
[Your Privacy Rights](#) | [Terms and Conditions](#)

3. Here is a sample screen that shows what the account details will look like.

Note: Refresh the page if the lab details are not showing up.

Playlist Transcript **Lab** Student Guides About

1: Request Your Lab

Your lab environment is ready for use, and will be available to you until 01/14/2021 18:00hrs America/Los_Angeles .

2: Access Your Lab

OCI Credentials

URL : <https://console.us-ashburn-1.oraclecloud.com/?tenant=gtmhol18>

User Name: 99527325-lab.user01

Password: 2JFqh]-N8ee{Z&\$!fhAQ

Compartment: 99527325-C01

Region: Ashburn

4. Launch a web browser on your local computer and navigate to the URL link listed as part of your account details.

Playlist Transcript **Lab** Student Guides About

1: Request Your Lab

Your lab environment is ready for use, and will be available to you until 01/14/2021 18:00hrs America/Los_Angeles .

2: Access Your Lab

OCI Credentials

URL : <https://console.us-ashburn-1.oraclecloud.com/?tenant=gtmhol18>

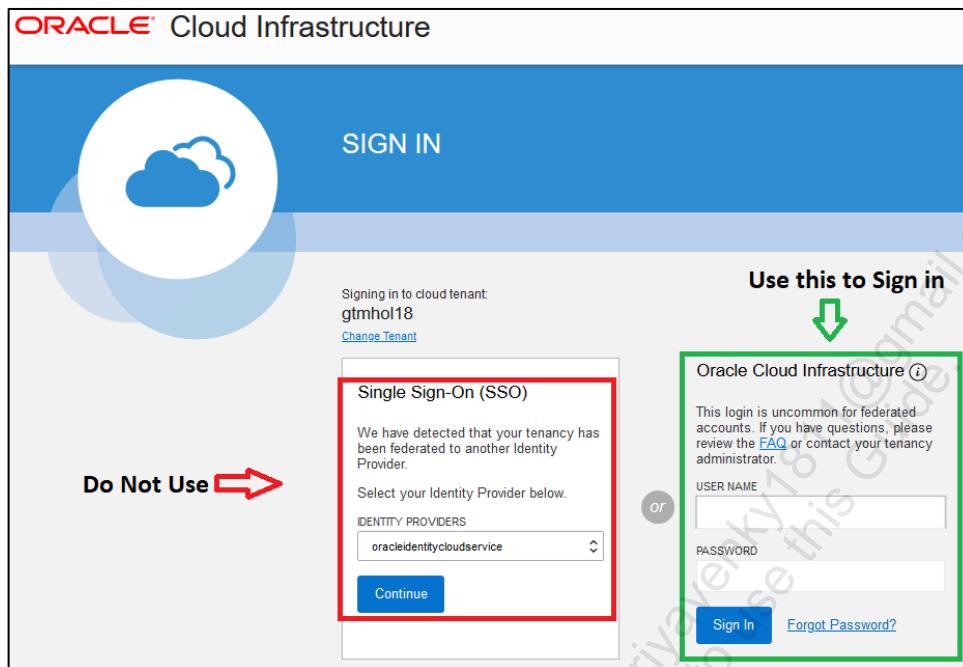
User Name: 99527325-lab.user01

Password: 2JFqh]-N8ee{Z&\$!fhAQ

Compartment: 99527325-C01

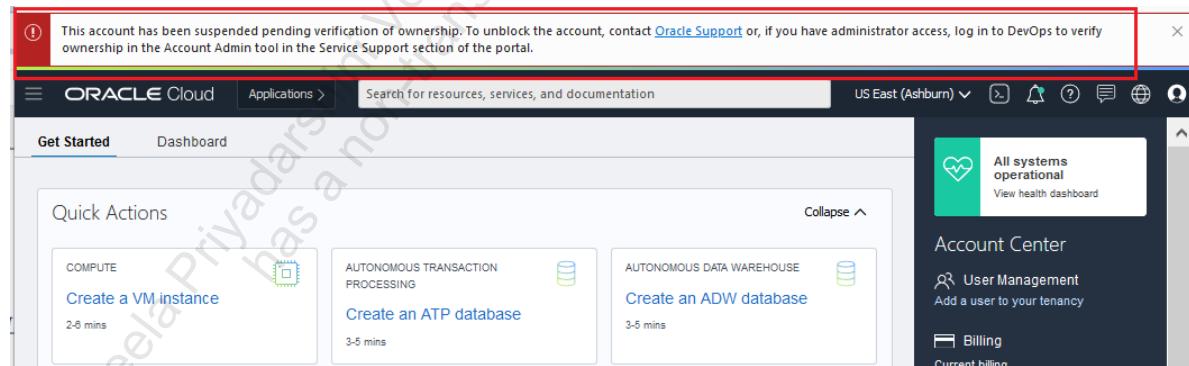
Region: Ashburn

- In the **Oracle Cloud Infrastructure** section highlighted below in green, enter the cloud account **User Name** and **Password** assigned to you and click “**Sign In**”. *Do not use the SSO option.*

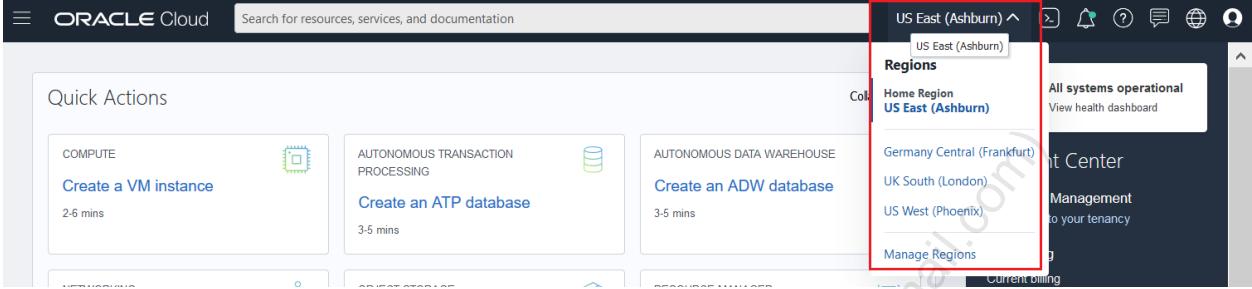


Note: You will be prompted to change the password on your first login; change the password and continue.

- At this point, you should be logged in to Oracle Cloud Infrastructure (OCI) Console, also referred to as the OCI home page. If you see a **warning message** as shown below, you can ignore it and close the message.



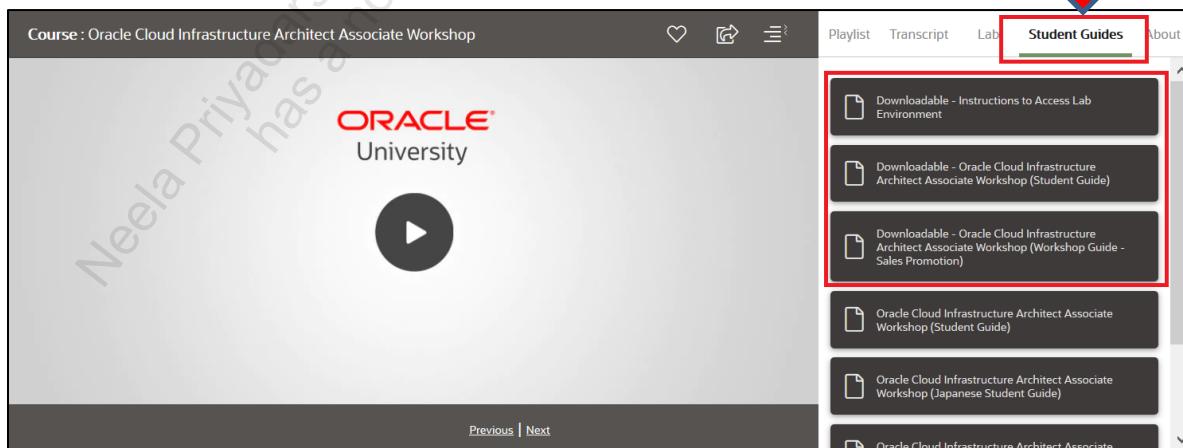
- You are now logged in to your account.

8. Verify and change **OCI Region** if required based on your account details.
 - a. In the following example the home region is **Ashburn** and it is selected in the OCI Console.
 
 - b. If the region assigned to you as part of your account details is not selected as shown above, then you need to change the region listed in your account details before you get started with the labs.
 - c. To change the region, while logged in to the OCI Console, click the region drop down menu option at the top, **US East (Ashburn) ▾** which is highlighted in the screenshot above.
 - d. This will list the available **Regions** for the OCI Tenant; select your region by clicking the corresponding name. This step is explained in detail in the very first practice of the activity guide.
 - e. Ensure you have set OCI Region to your assigned region and proceed further.

9. Proceed with project work.

Instructions to Download Student Guide

1. The details of each lab is available in the Workshop guide that is listed in the **Student Guides** tab.
2. To download PDF guides, click the **Student Guides** tab, which is a part of the subscription course UI.



Course : Oracle Cloud Infrastructure Architect Associate Workshop

Student Guides

- Downloadable - Instructions to Access Lab Environment
- Downloadable - Oracle Cloud Infrastructure Architect Associate Workshop (Student Guide)
- Downloadable - Oracle Cloud Infrastructure Architect Associate Workshop (Workshop Guide - Sales Promotion)
- Oracle Cloud Infrastructure Architect Associate Workshop (Student Guide)
- Oracle Cloud Infrastructure Architect Associate Workshop (Japanese Student Guide)

3. Under this tab you will find guides with the title starting with the key word “**Downloadable...**” as shown in the screenshot. Click them to download the PDF document.
4. Based on your browser setting, the guide will either get opened as a PDF document in a new tab or get downloaded into your default download location on your system.
5. By default, the file name will be **pdf.pdf**.
6. Rename the files as “*OCI SLS Project (Student Guide).pdf*.”

Neela Priyadarshini Venkatachalam (priyavenky1811@gmail.com)
has a non-transferable license to use this Guide.