Contents

[Overview 2](#_Toc388457064)

[Assumptions 3](#_Toc388457065)

[Environment 3](#_Toc388457066)

[Instructions 4](#_Toc388457067)

[JDeveloper Application 4](#_Toc388457068)

[Datasource configuration 4](#_Toc388457069)

[Database schema 5](#_Toc388457070)

[Oracle Sales Cloud connection Configuration 5](#_Toc388457071)

[Create credential for Oracle Sales Cloud connection in JCS credential store. 6](#_Toc388457072)

[Configuration SalesAccount custom fields in Oracle Sales Cloud 9](#_Toc388457073)

[Create a SalesAccount in Oracle Sales Cloud 15](#_Toc388457074)

[Set SalesAccount name in the testing Servlet 17](#_Toc388457075)

[Build the ear application 18](#_Toc388457076)

[Deploy the application into JCS 21](#_Toc388457077)

[Test the web application in JCS 26](#_Toc388457078)

[Samples 27](#_Toc388457079)

[Create operation 27](#_Toc388457080)

[Update operation 28](#_Toc388457081)

[Get operation 29](#_Toc388457082)

[Find operation 30](#_Toc388457083)

**Disclaimer**

All sample code is provided by Oracle for illustrative purposes only.

These sample code examples have not been thoroughly tested under all conditions. Oracle, therefore, cannot guarantee or imply security, reliability, serviceability, or function of the sample code.

All sample code contained herein are provided to you "AS IS" without any warranties of any kind. The implied warranties of non-infringement, merchantability and fitness for a particular purpose are expressly disclaimed.

# Overview

The purpose of this document is to guide you how to use this sample application to handle custom fields defined in Oracle Sales Cloud.

Oracle Sales Cloud provides tenants the ability to easily extend standard out-of-box Sales Cloud objects such as the Opportunity object etc. with additional custom fields.This extensibility feature is often used by tenants to implement their tenant-specific requirements and also leveraged by Oracle ISV partners to enrich the tenant's experience of Oracle Sales Cloud with additional capabilities. Each tenant may implement multiple partner integrations on a specific tenant instance - each of these partner integrations could extend Sales Cloud with different custom fields. When the standard objects are extended with custom fields (the sandbox with the additional fields is published), the additional fields are immediately visible to all web service clients of the tenant instance.    
   
As a specific example of the scenario, a tenant administrator extends the opportunity standard object with an additional custom field "OptyStrategy" in a sandbox. The tenant administrator then publishes the sandbox. The tenant-specific service catalog references the Opportunity Service wsdl URL  - this wsdl will now contain a reference in an imported Opportunity.xsd to the custom field "OptyStrategy" represented by an element "OptyStrategy\_c" as shown below.    
   
<https://tenant-id-crm.oracleoutsourcing.com/opptyMgmtOpportunities/OpportunityService?XSD=/oracle/apps/sales/opptyMgmt/opportunities/opportunityService/Opportunity.xsd>    
<xsd:element minOccurs="0" name="OptyStrategy\_c" nillable="true"    
type="xsd:string"/>    
   
All web service clients of the Opportunity Service for that tenant will receive the OptyStrategy\_c in the SOAP response on the next invocation of the Opportunity Service.  

The solution here is to leverage JAX-WS handler to manipulate custom fields without regenerating the static web service proxy client whenever custom fields are created/updated in Oracle Sales Cloud with the updated web service schema.

## Assumptions

* It is assumed that the reader is competent with Oracle JDeveloper and general Java development, as well as Oracle Sales Cloud extensibility concepts (i.e. Application Composer).
* Reader is aware of creating jar,war,ear deployment packages
* It’s also assumed that reader is aware of WebLogic deployment

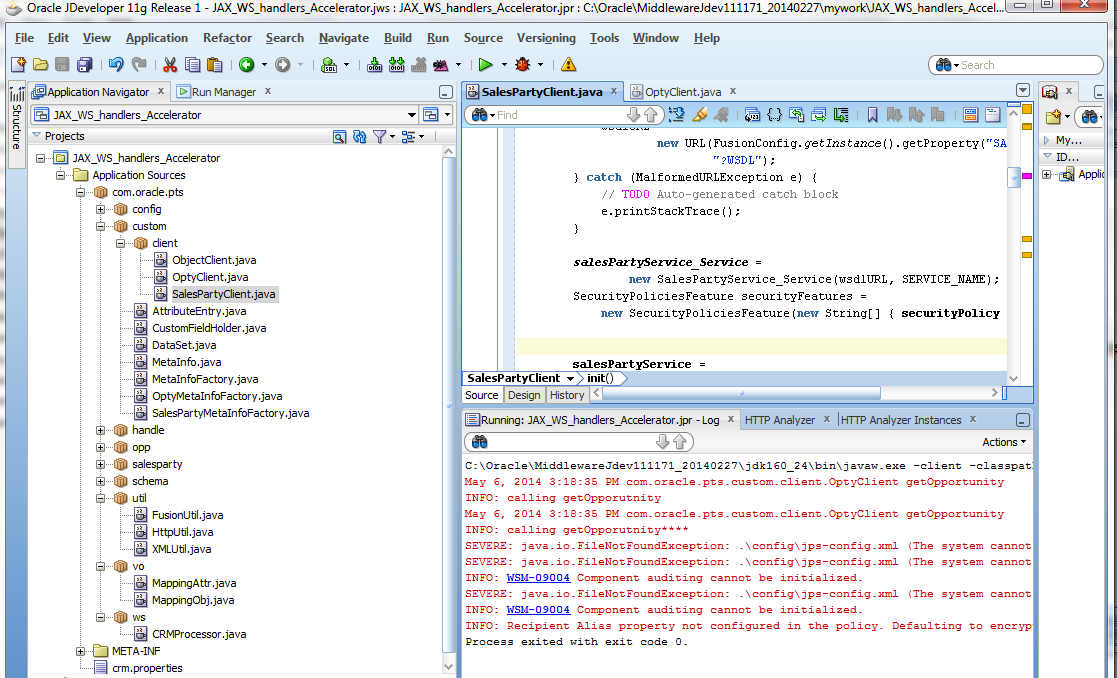
## Environment

* Oracle Sales Cloud Release 8
* Oracle Java Cloud Service 13.2
* Oracle JDeveloper 11.1.1.7.1

# Instructions

## JDeveloper Application

* Open JAX\_WS\_handlers\_Accelerator application and JAX\_WS\_handlers\_Accelerator project



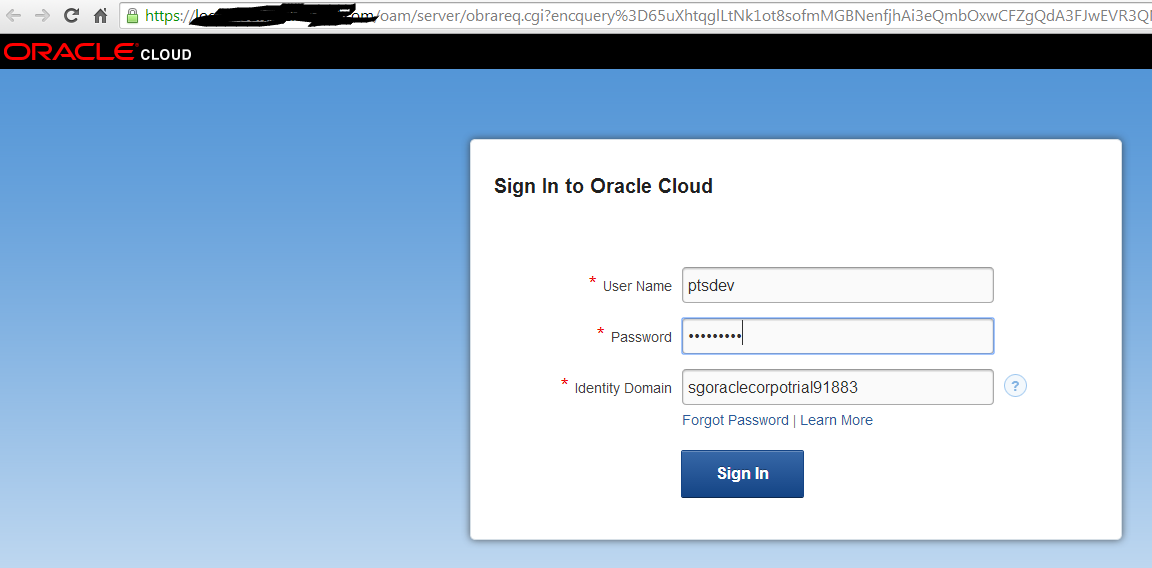
## Datasource configuration

Open crm.properties in JDeveloper and set database datasource name that is used for the JCS.

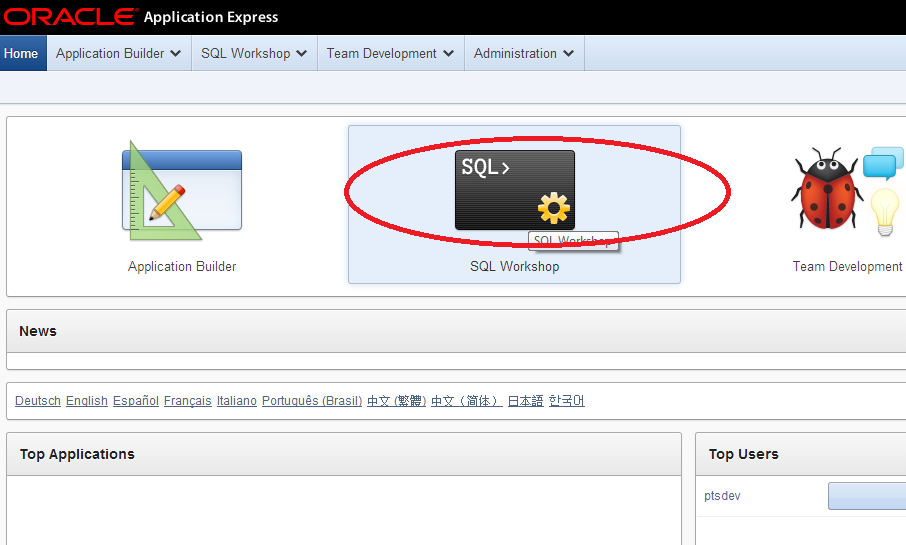
|  |
| --- |
| # Sample JCS datasoruce name  datasource=javatrial1316db |

## Database schema

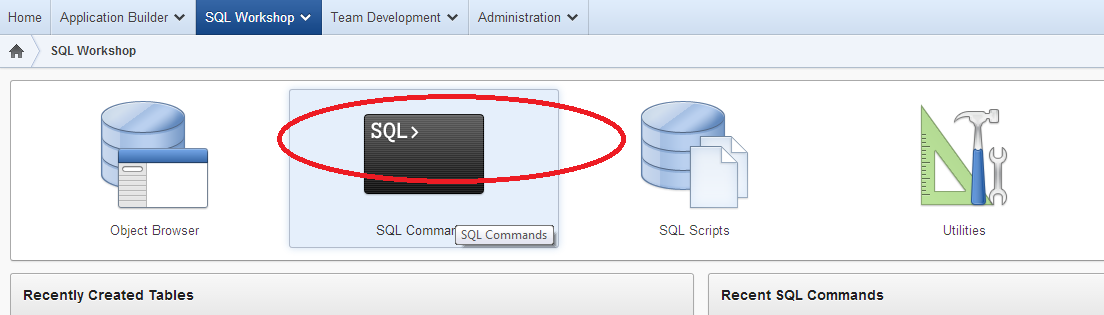
* Login to DBCS admin console



* Click SQL Workshop

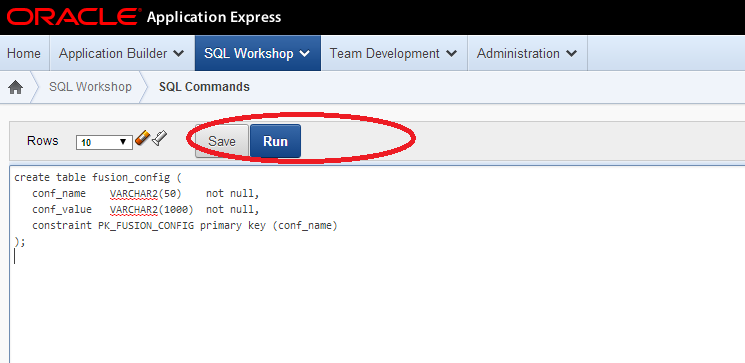


* Click SQL



* Run SQL command below to create fusion\_config database table

|  |
| --- |
| create table fusion\_config (  conf\_name VARCHAR2(50) not null,  conf\_value VARCHAR2(1000) not null,  constraint PK\_FUSION\_CONFIG primary key (conf\_name)  ); |



## Oracle Sales Cloud connection Configuration

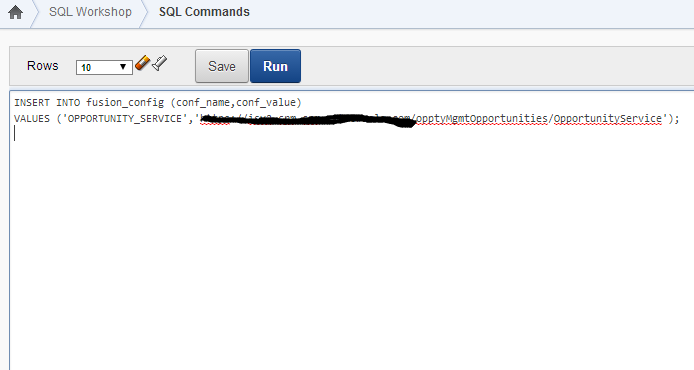
* Set Oracle Sales Cloud connection info in fusion\_config database where OSC\_HOME is Oracle Sales Cloud hostname

|  |  |
| --- | --- |
| OPPORTUNITY\_SERVICE | https://<OSC\_HOME>opptyMgmtOpportunities/OpportunityService |
| SALESPARTY\_SERVICE | https://<OSC\_HOME>/crmCommonSalesParties/SalesPartyService |
| FUSION\_USER | Osckey |
| FUSION\_SalesPartyService\_SCHEMA | https://<OSC\_HOME>/crmCommonSalesParties/SalesPartyService?XSD=/oracle/apps/crmCommon/salesParties/salesPartiesService/SalesParty.xsd |
| FUSION\_OptyService\_SCHEMA | https://<OSC\_HOME>/crmCommonSalesParties/SalesPartyService?XSD=/oracle/apps/crmCommon/salesParties/salesPartiesService/SalesParty.xsd |

For each row above, run SQL command

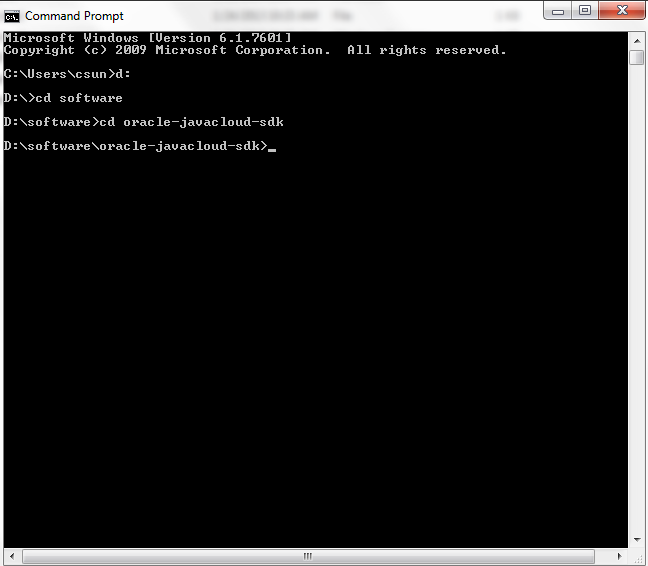
INSERT INTO fusion\_config (conf\_name,conf\_value)

VALUES ('name',' value');



## Create credential for Oracle Sales Cloud connection in JCS credential store.

* Download Oracle Java Cloud Service SDK (release 14.1.2.0) from <http://www.oracle.com/technetwork/middleware/weblogic/downloads/java-cloud-sdk-1848874.html>
* Unzip the downloaded SDK
* Open a command prompt and go to the SDK location



* Create a credential

Sample:

javacloud -id sgoraclecorpotrial91883 -si javatrial1316 -u ptsdev -p "jcs\_admin\_pass" -adminurl https://javaservices.us2.cloud.oracle.com -key osckey -keyuser Matt.Hooper -keypassword osc\_pass -set-credential

-id: JCS identity domain

-si: java instance name

-u: JCS admin user

-p: JCS admin password

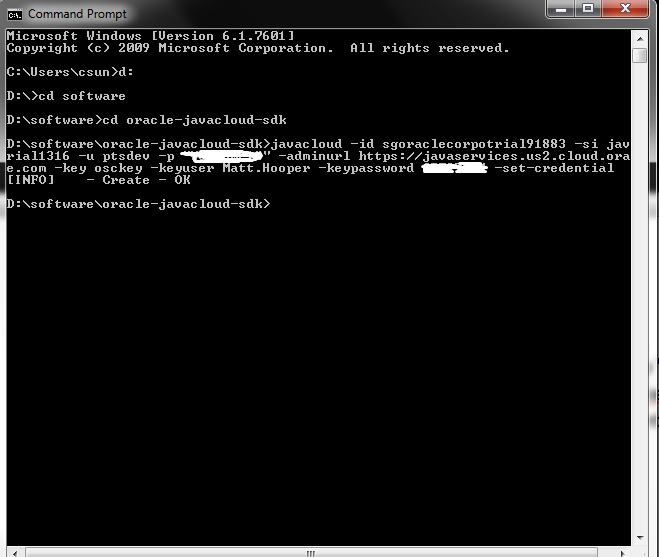
-adminurl: JCS admin url

-key: the name of this credential

-keyuser: the user used to connect to Oracle Sales Cloud

-keypassword: the associated password for the keyuser

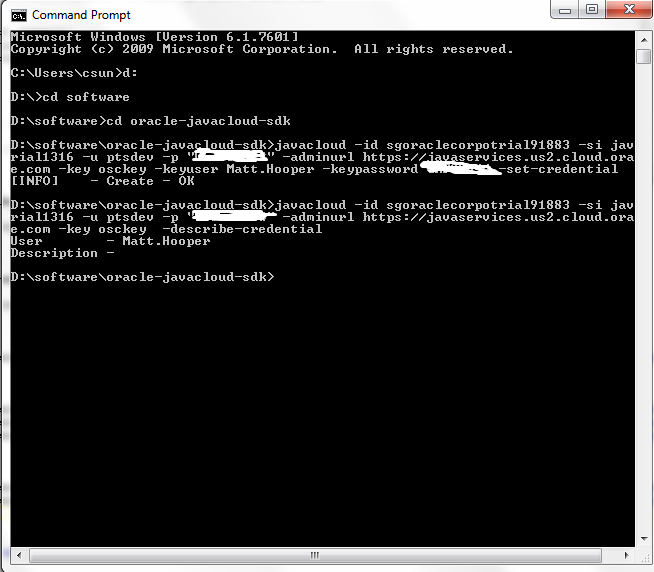
For complete command line reference, go to doc folder in SDK utility



* Check if the key is created

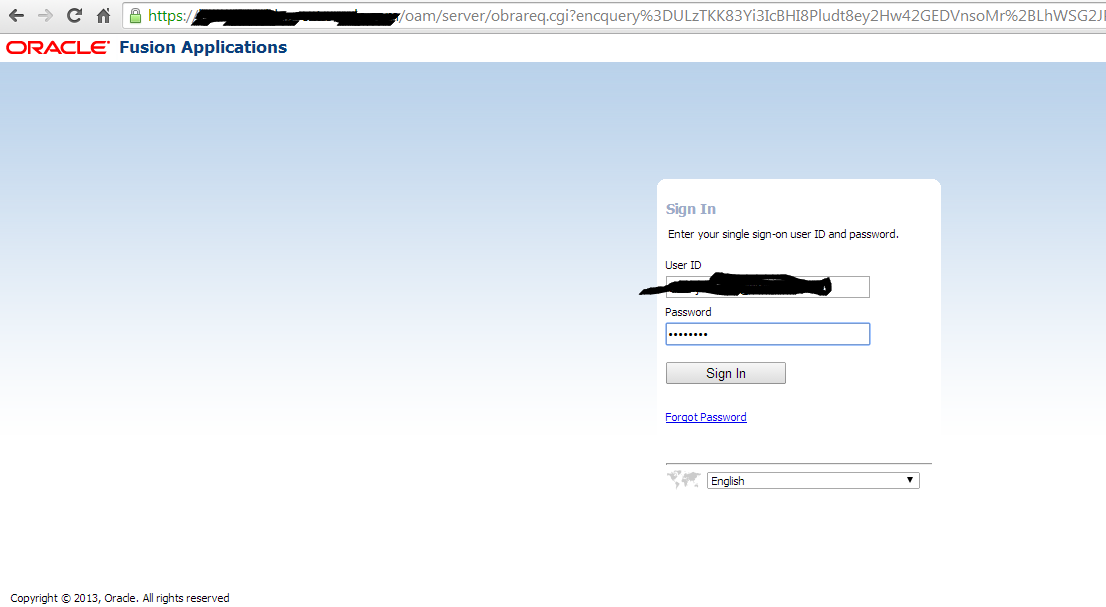
Sample:

javacloud -id sgoraclecorpotrial91883 -si javatrial1316 -u ptsdev -p " jcs\_admin\_pass " -adminurl https://javaservices.us2.cloud.oracle.com -key osckey -describe-credential

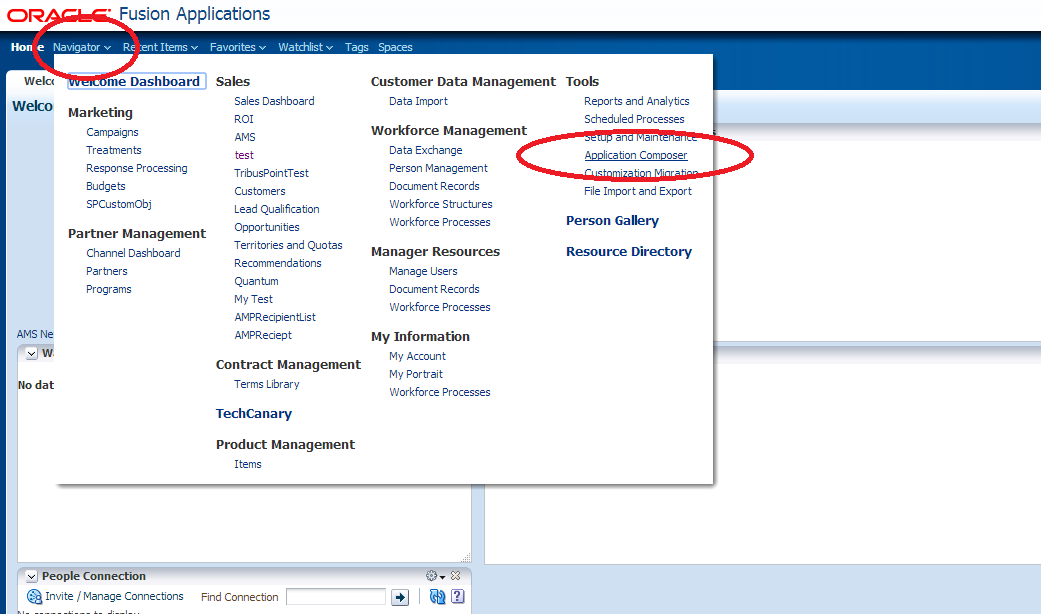


## Configuration SalesAccount custom fields in Oracle Sales Cloud

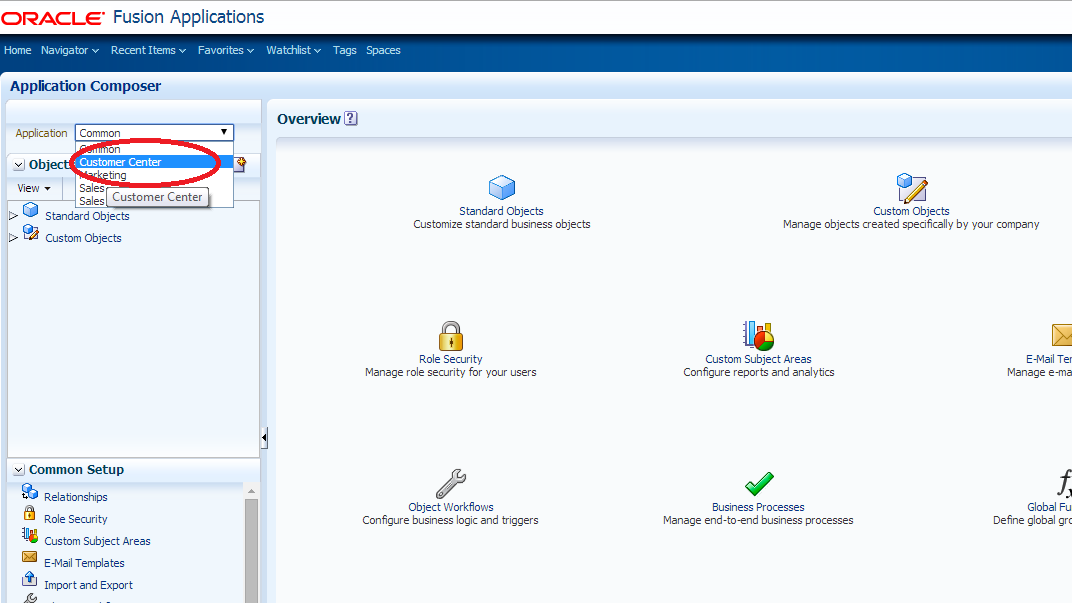
* Login to Oracle Sales Cloud



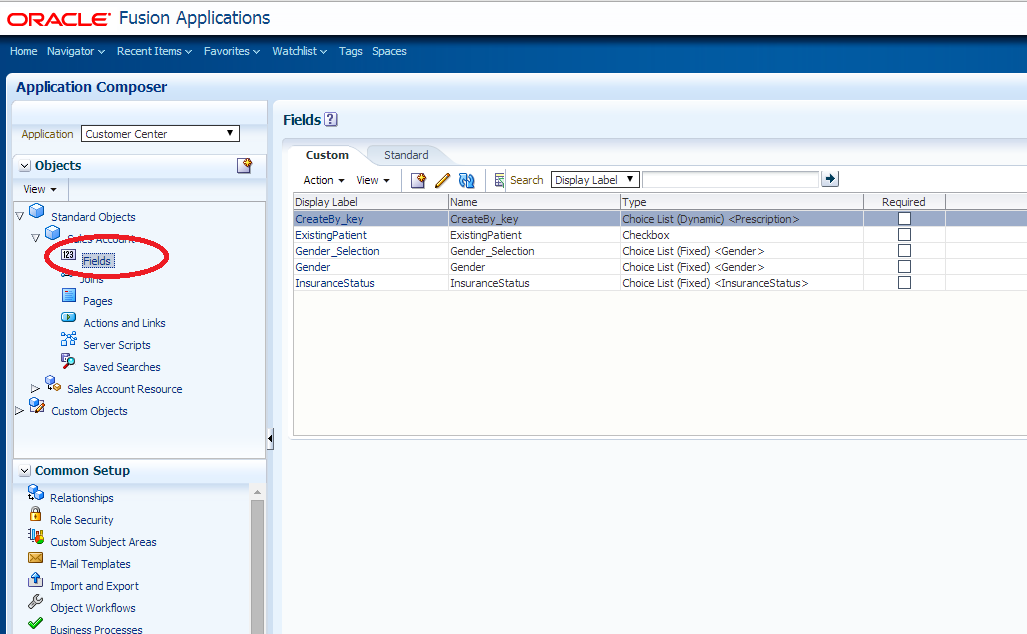
* Click Navigator -> Application Composer



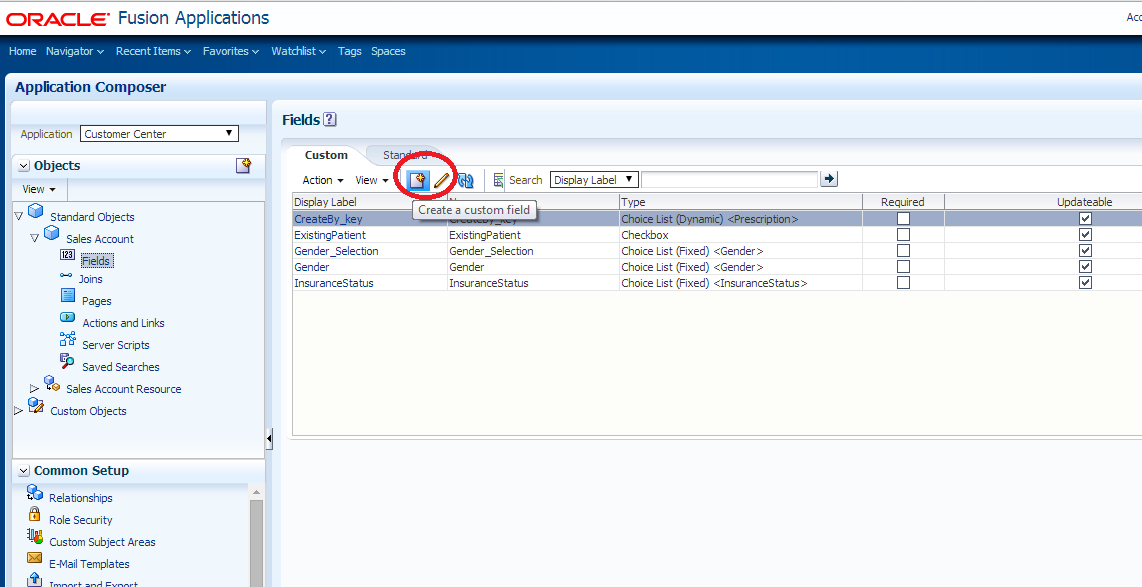
* Select Customer Center



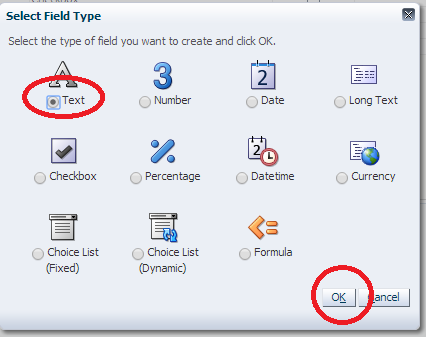
* Expand Standard Objects and SalesAccount, then click Fields



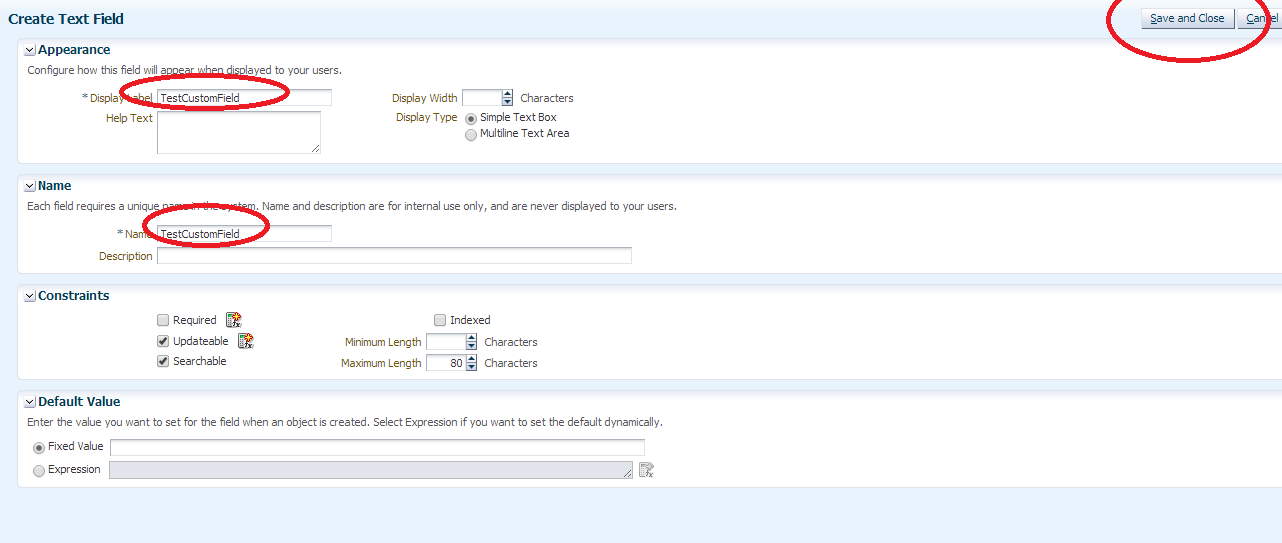
* Click ‘+’ icon to create a custom field



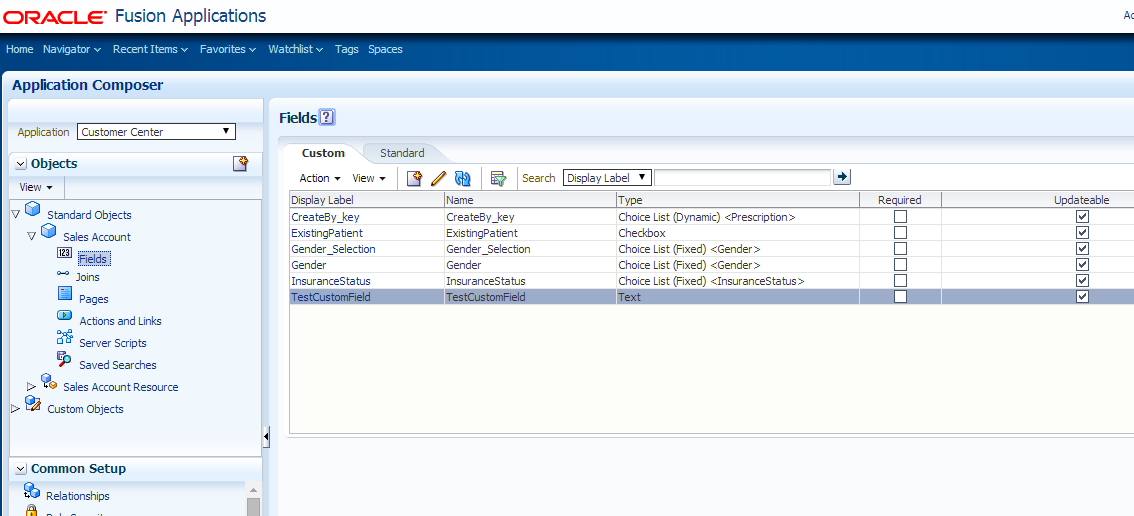
* Select Text and press OK



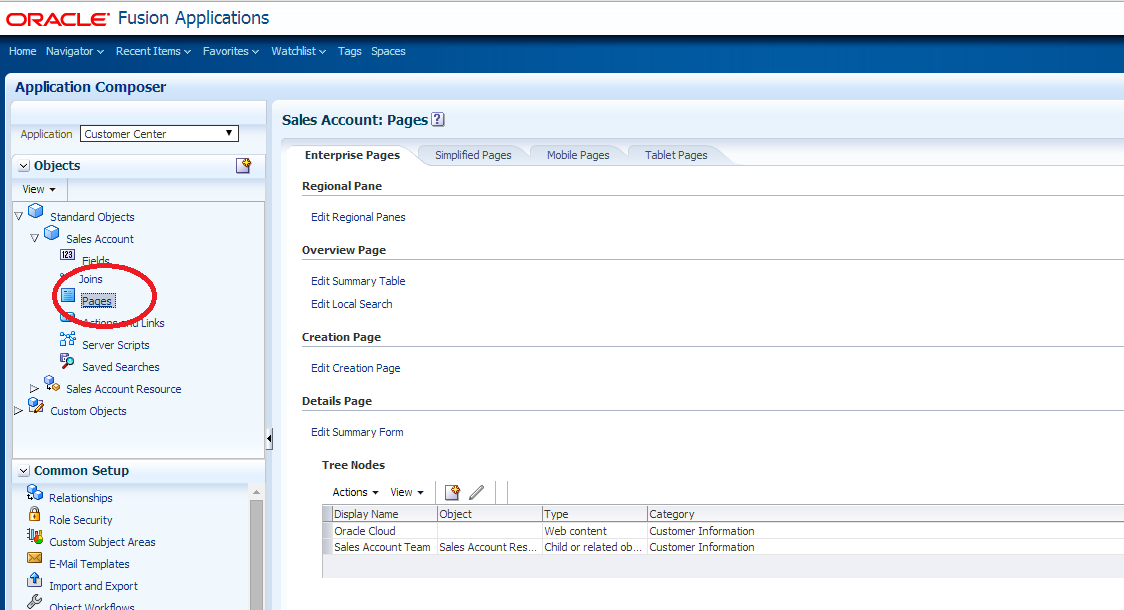
* Fill Display Label and Name as TestCustomField, then click Close and Save



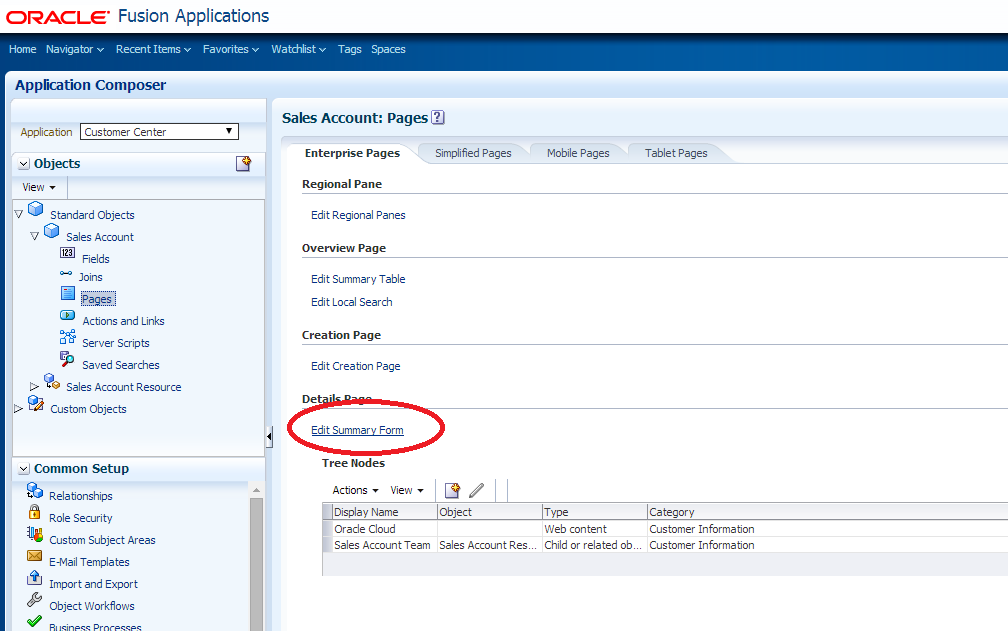
* TestCustomField is created



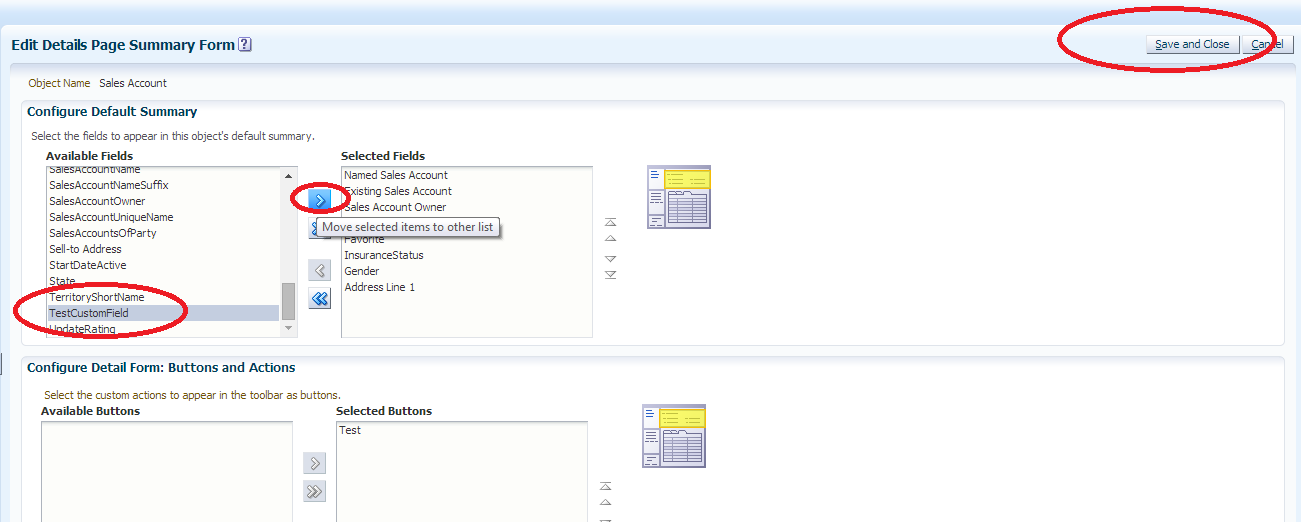
* Click Pages



* Click Edit Summary Form

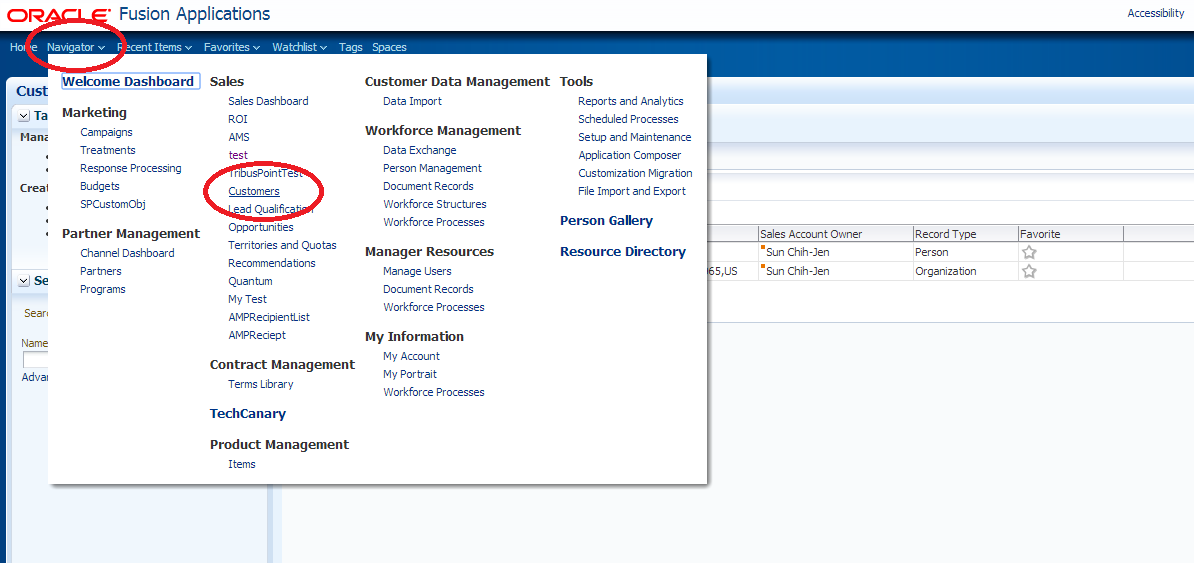


* Select TestCustomField and move the Selected Fields, then click Save and Close

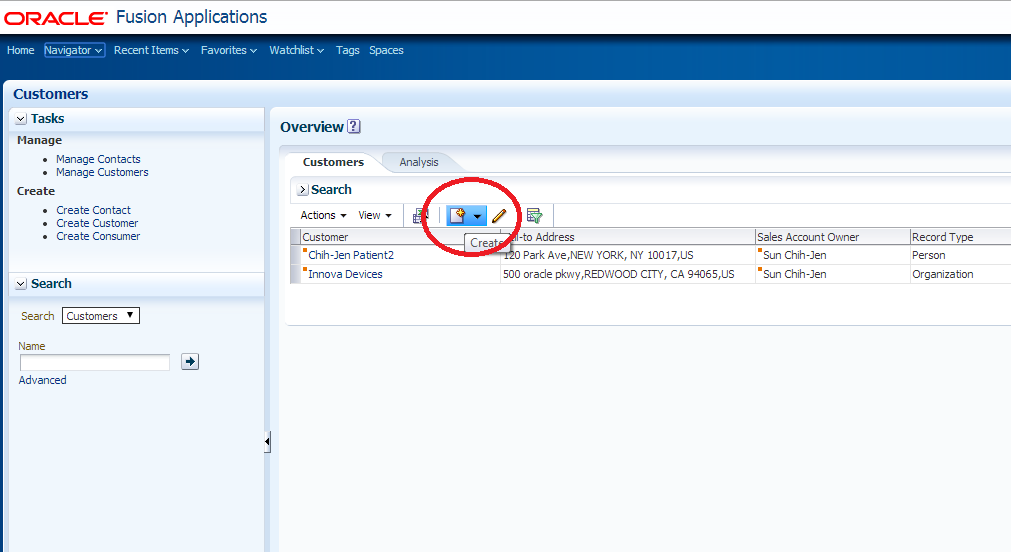


## Create a SalesAccount in Oracle Sales Cloud

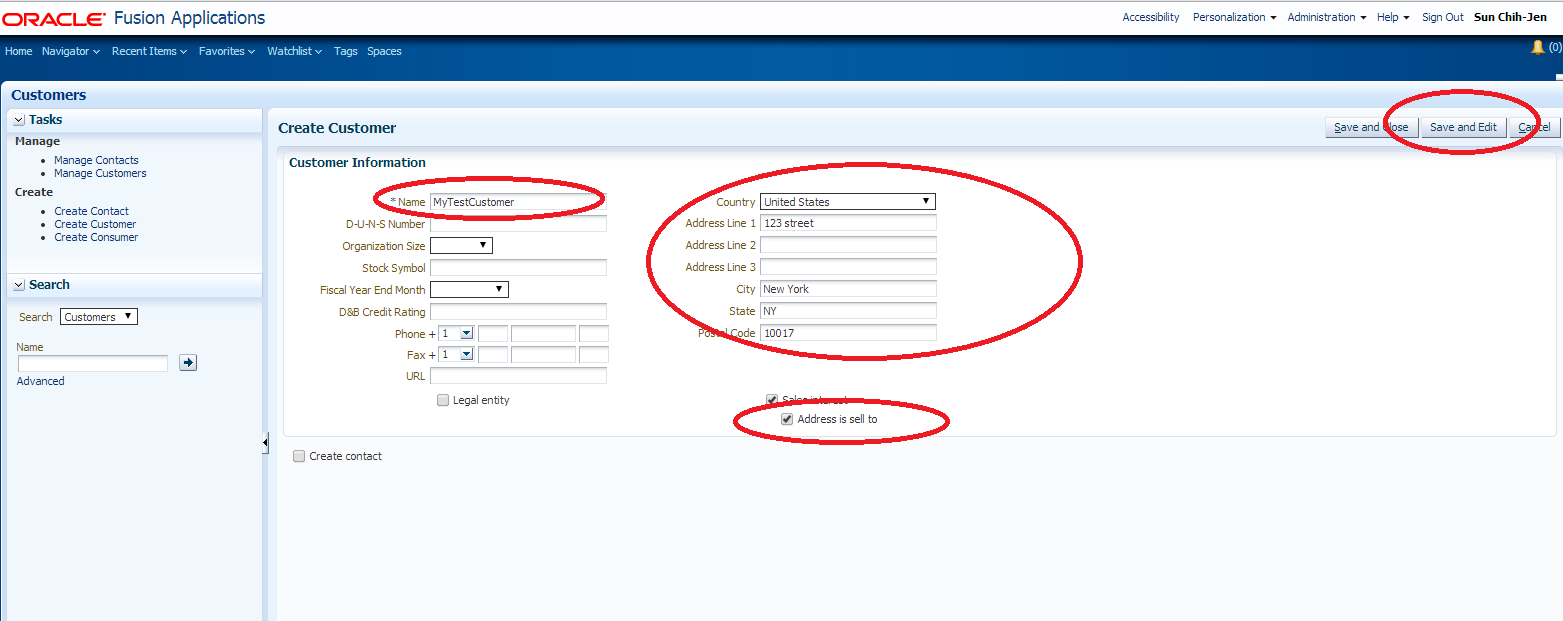
* Click Navigator -> Customers



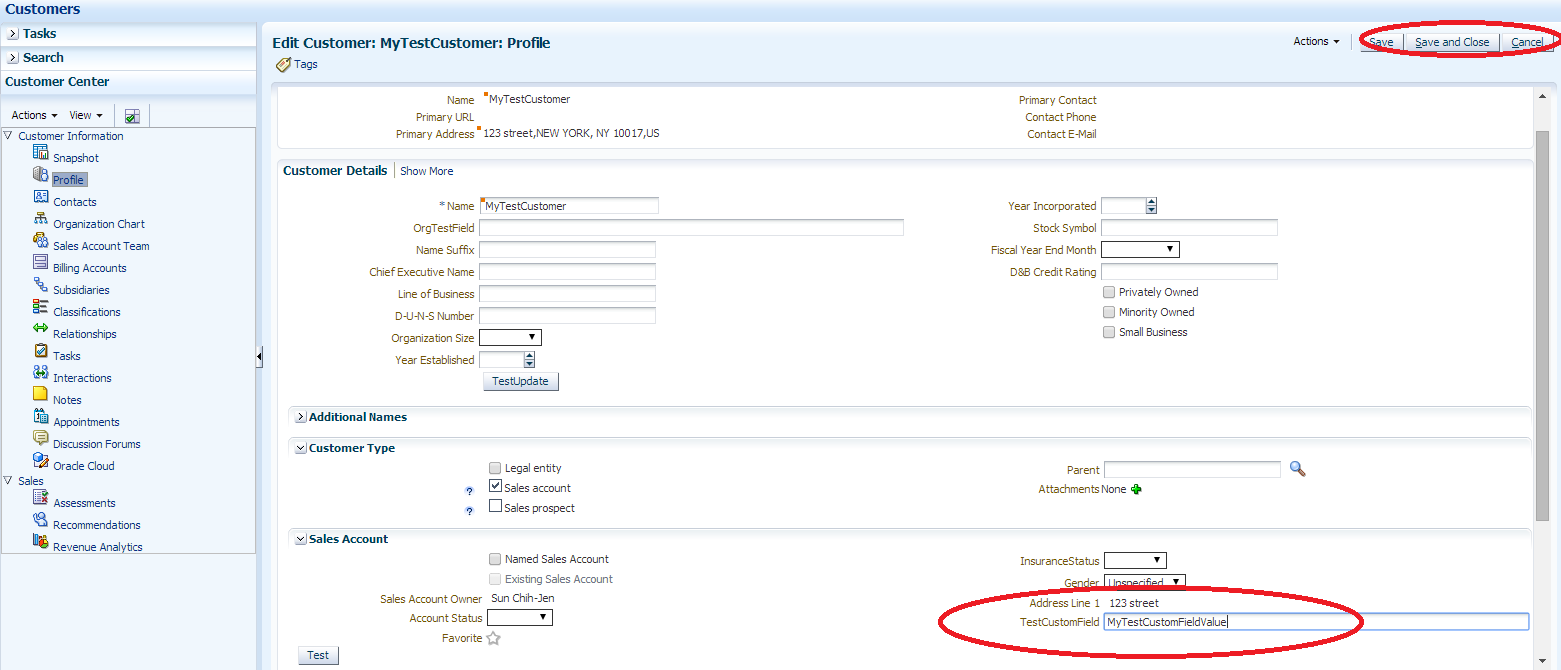
* Click ‘+’ icon to create a customer



* Set Name as MyTestCustomer, an address and check Address is sell to checkbox, then click Save and Edit

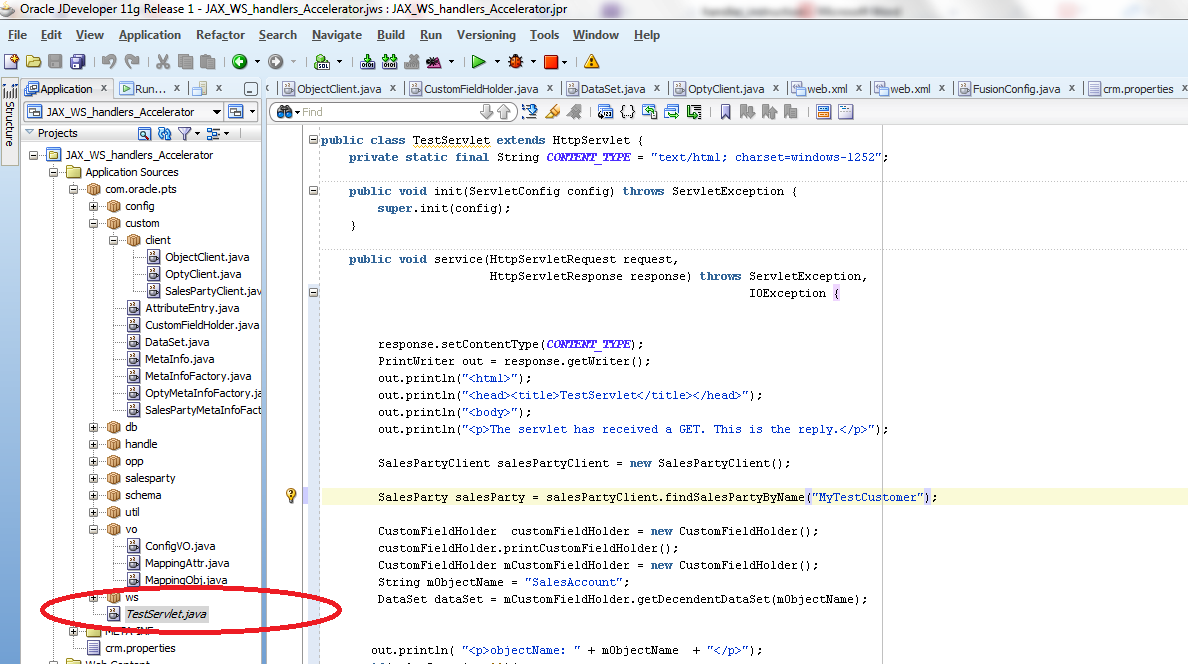


* Set TestCustomField value as MyTestCustomFieldValue



## Set SalesAccount name in the testing Servlet

* Open TestServlet java class in JDeveloper

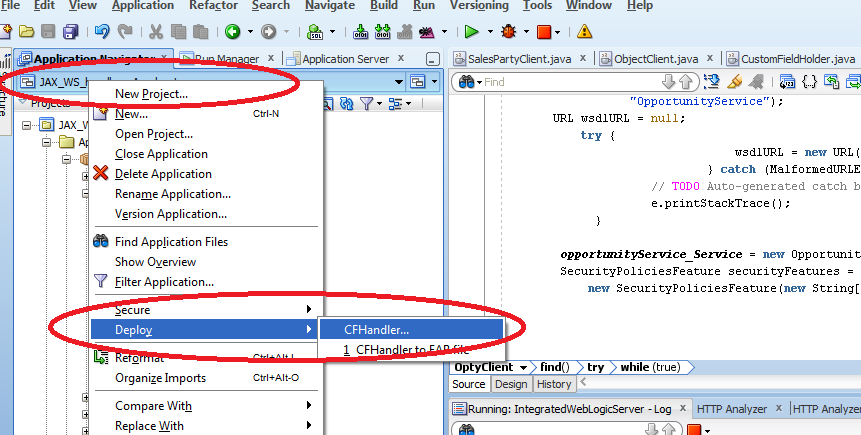


* Set SalesParty Name as MyTestCustomer and save it

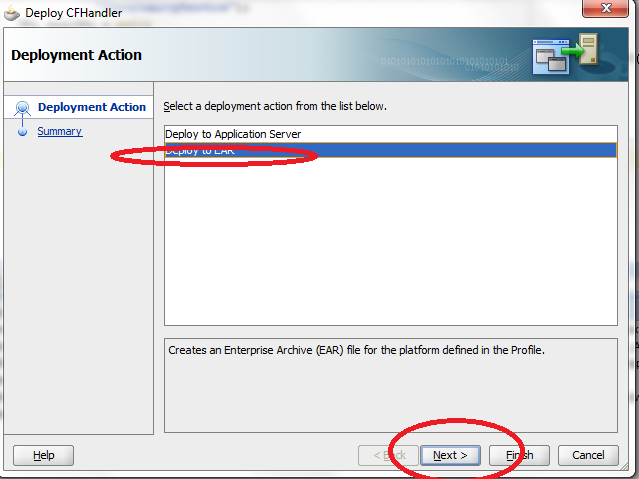


## Build the ear application

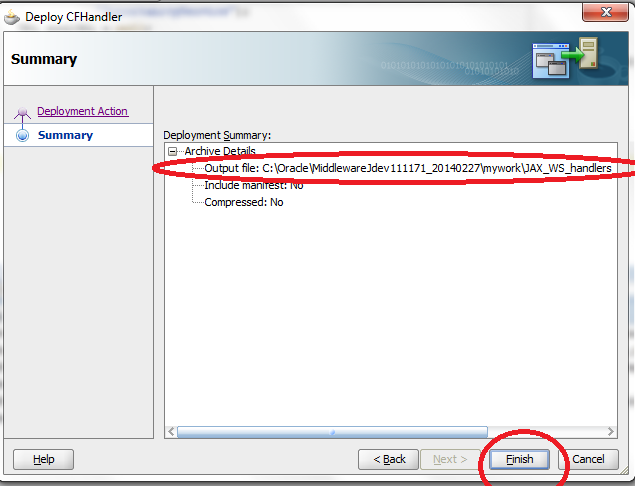
* Right click JAX\_WS\_handlers\_Accelerator application and click Deployment -> CFHandler



* Select Deploy ear and click next

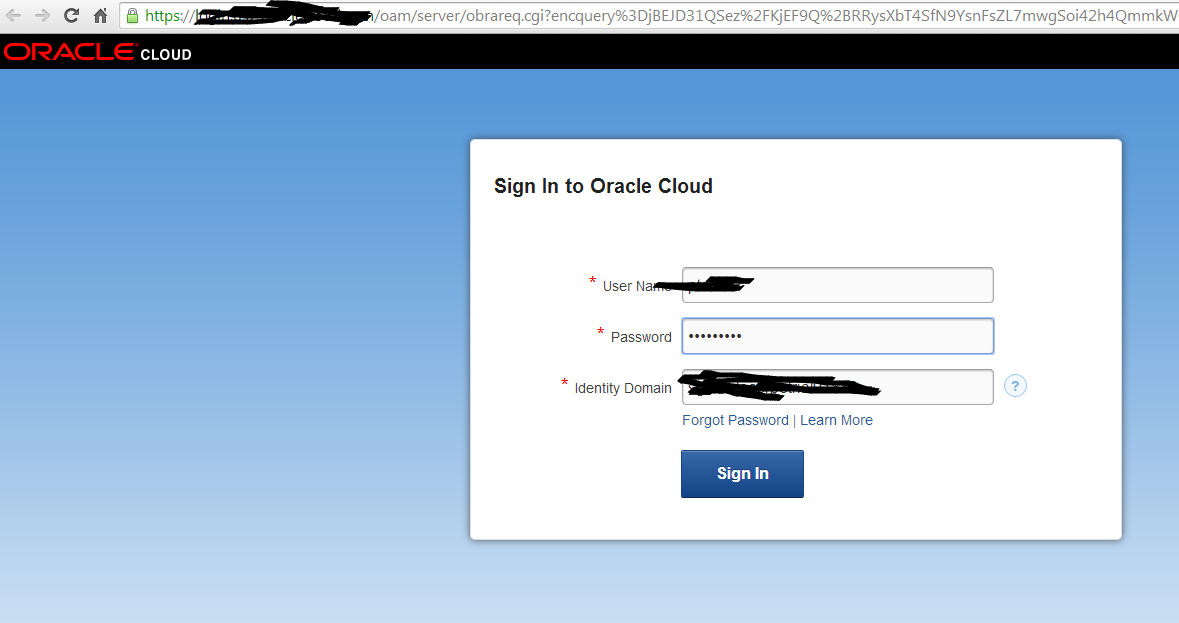


* Click Finish and remember output ear file location.

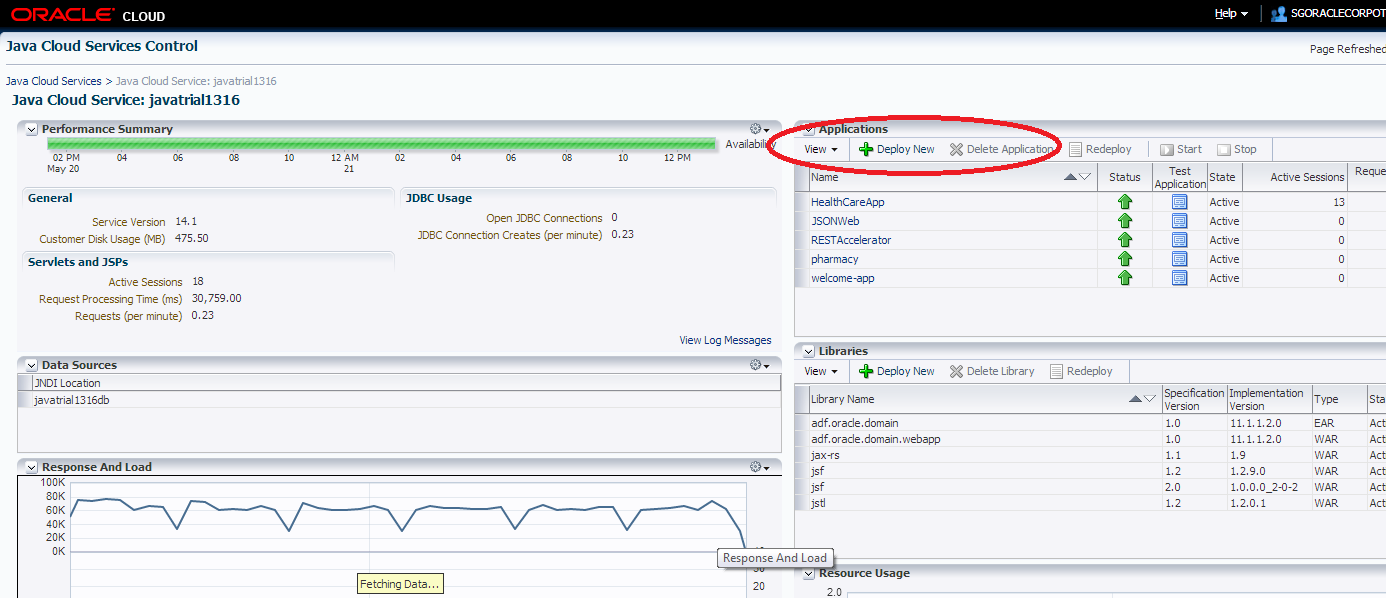


## Deploy the application into JCS

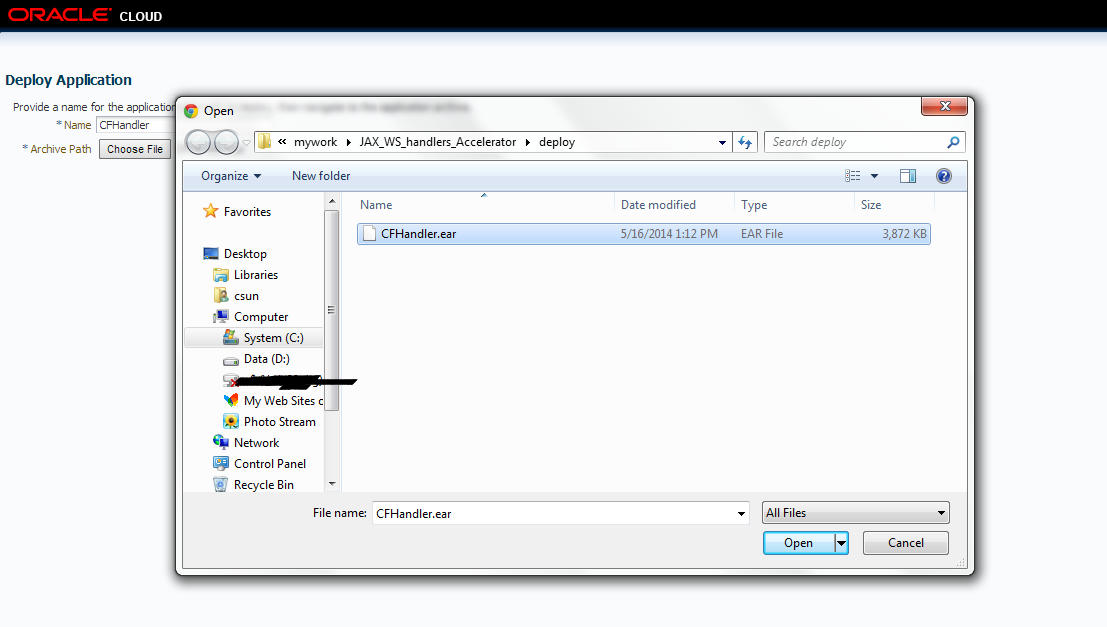
* Login to JCS admin console



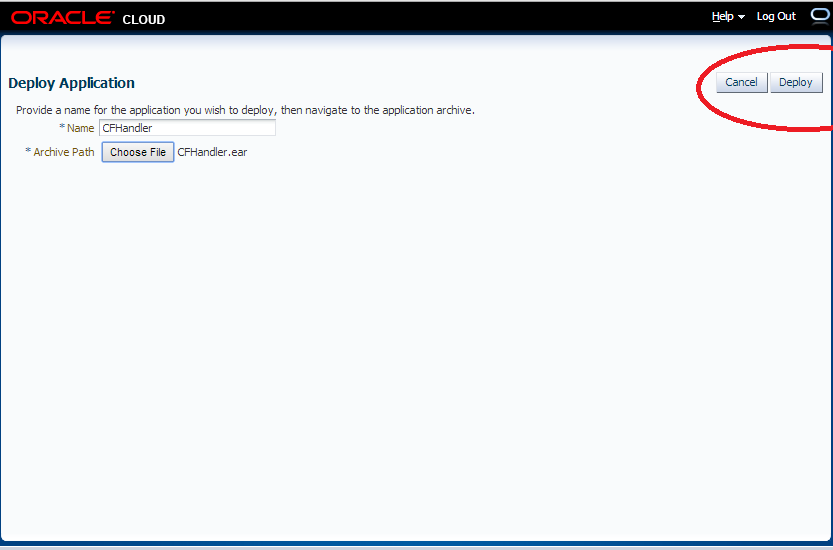
* Click Deploy New



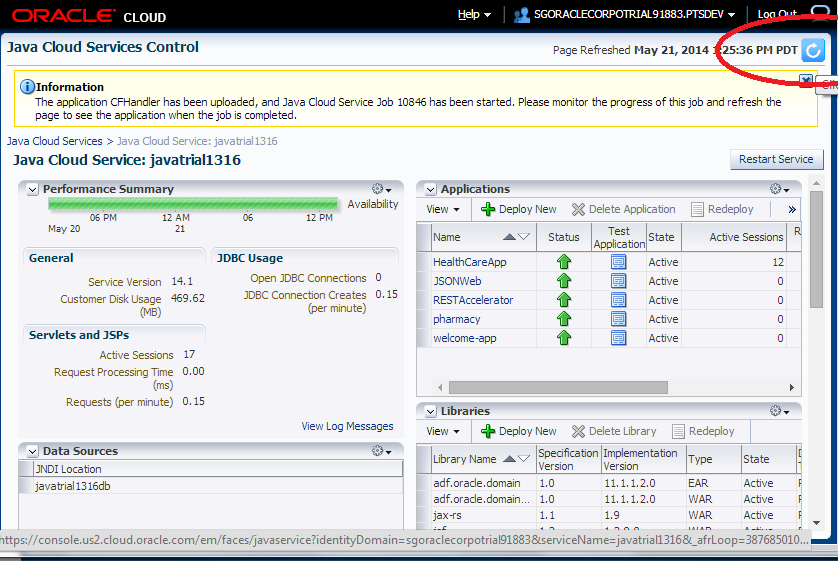
* Set Name as CFHandler and select ear application from the previous steps



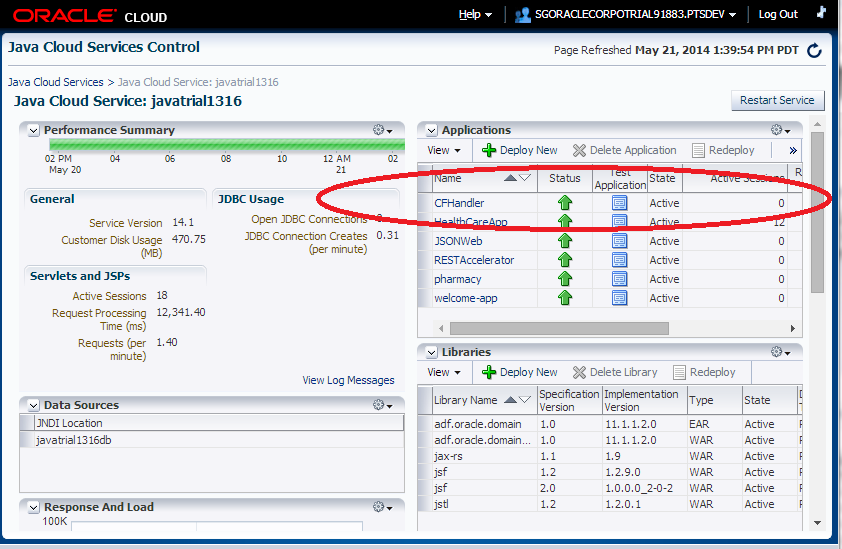
* Click Deploy



* Wait for a minute and click refresh



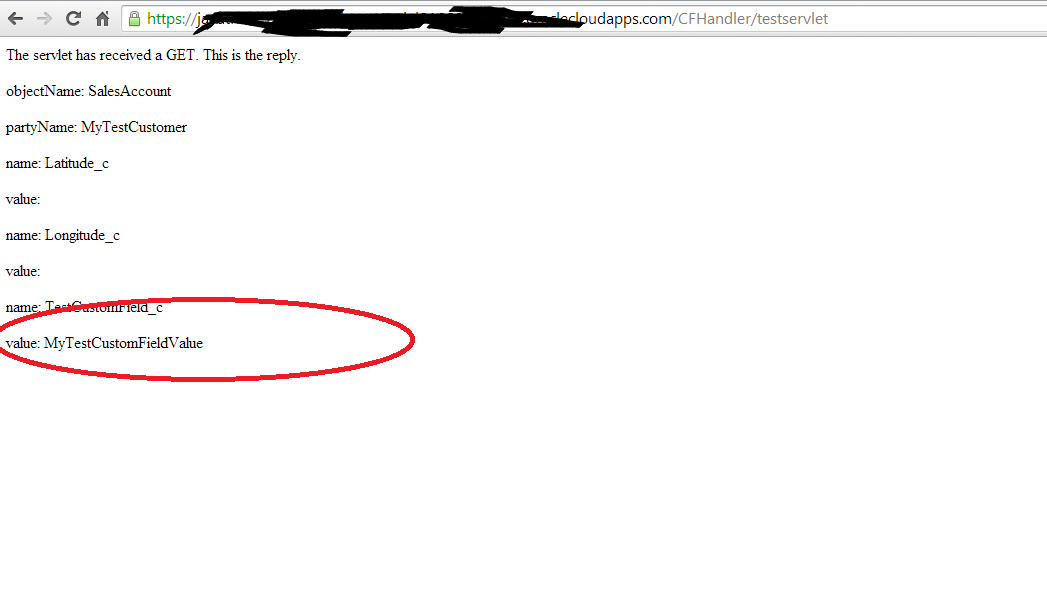
* CFHanlder application is deployed



## Test the web application in JCS

* Run TestServlet in JCS environment

URL: https://<JCS\_HOME>/CFHandler/testservlet



# Samples

## Create operation

|  |
| --- |
| public void testCreate() {  ObjectFactory objectFactory = new ObjectFactory();  Opportunity opportunity = objectFactory.createOpportunity();    String optName = "optName" + String.valueOf(System.currentTimeMillis());    opportunity.setName(optName);    logger.info("Creatinging " + optName );    try {  DataSet dataSet = new DataSet();  dataSet.setName(objectName);  AttributeEntry attr = new AttributeEntry();  attr.setName("customfield1\_c");  attr.setValue("customfield1Value");  dataSet.getAttributeList().add(attr);  attr = new AttributeEntry();  attr.setName("customfield2\_c");  attr.setValue("customfield2Value");  dataSet.getAttributeList().add(attr);  CustomFieldHolder customFieldHolder = new CustomFieldHolder();  customFieldHolder.setObjectName(objectName);  customFieldHolder.setDataSet(dataSet);    opportunityService.createOpportunity(opportunity);  } catch (ServiceException e) {  e.printStackTrace();  logger.info(e.getMessage());    }  } |

## Update operation

|  |
| --- |
| public void update(String optyId) {  long optyIdLong = Long.parseLong(optyId);  try {  ObjectFactory objectFactory = new ObjectFactory();  // Opportunity opportunity = opportunityService.getOpportunity(optyIdLong);  Opportunity opportunity = objectFactory.createOpportunity();  opportunity.setOptyId(optyIdLong);  DataSet dataSet = new DataSet();  dataSet.setName(objectName);  AttributeEntry attr = new AttributeEntry();  attr.setName("customfield1\_c");  attr.setValue("customfield3Value");  dataSet.getAttributeList().add(attr);  attr = new AttributeEntry();  attr.setName("customfield2\_c");  attr.setValue("customfield4Value");  dataSet.getAttributeList().add(attr);  CustomFieldHolder customFieldHolder = new CustomFieldHolder();  customFieldHolder.setObjectName(objectName);  customFieldHolder.setDataSet(dataSet);    opportunityService.updateOpportunity(opportunity);  } catch (ServiceException e) {  e.printStackTrace();  logger.info(e.getMessage());  }  }  public Opportunity getOpportunity(long optyId) {  CustomFieldHolder customFieldHolder = new CustomFieldHolder();  customFieldHolder.setObjectName(objectName);  Opportunity opportunity = null;  logger.info("calling getOpporutnity");  try {  logger.info("calling getOpporutnity\*\*\*\*");  opportunity = opportunityService.getOpportunity(optyId);    } catch (Exception e) {  logger.info("There is an errorE: " + e.getMessage());  e.printStackTrace();  }  customFieldHolder.printCustomFieldHolder();    return opportunity;  } |

## Get operation

|  |
| --- |
| public Opportunity getOpportunity(long optyId) {  CustomFieldHolder customFieldHolder = new CustomFieldHolder();  customFieldHolder.setObjectName(objectName);  Opportunity opportunity = null;  logger.info("calling getOpporutnity");  try {  logger.info("calling getOpporutnity\*\*\*\*");  opportunity = opportunityService.getOpportunity(optyId);    } catch (Exception e) {  logger.info("There is an errorE: " + e.getMessage());  e.printStackTrace();  }  customFieldHolder.printCustomFieldHolder();    return opportunity;  } |

## Find operation

|  |
| --- |
| public List<Opportunity> find() {  CustomFieldHolder customFieldHolder = new CustomFieldHolder();  customFieldHolder.setObjectName(objectName);  List<Opportunity> returnList = new ArrayList<Opportunity>();  try {  ObjectFactory factory = new ObjectFactory();  FindCriteria findCriteria = factory.createFindCriteria();  findCriteria.setExcludeAttribute(false);  findCriteria.getFindAttribute().add("OptyId");  findCriteria.getFindAttribute().add("TargetPartyName");  findCriteria.getFindAttribute().add("TargetPartyId");  findCriteria.getFindAttribute().add("Name");  findCriteria.getFindAttribute().add("PartyName1");  findCriteria.getFindAttribute().add("WinProb");  findCriteria.getFindAttribute().add("EffectiveDate");  findCriteria.getFindAttribute().add("Revenue");    SortOrder sortOrdersortOrder = factory.createSortOrder();  SortAttribute sortAttribute = factory.createSortAttribute();  sortAttribute.setName("Revenue");  sortAttribute.setDescending(true);  sortOrdersortOrder.getSortAttribute().add(sortAttribute);  findCriteria.setSortOrder(sortOrdersortOrder);      ViewCriteria viewCriteria= factory.createViewCriteria();      ViewCriteriaRow viewCriteriaRow = factory.createViewCriteriaRow();  viewCriteriaRow.setUpperCaseCompare(true);  viewCriteriaRow.setConjunction(Conjunction.AND);      ViewCriteriaItem viewCriteriaItem = factory.createViewCriteriaItem();  viewCriteriaItem.setUpperCaseCompare(true);  viewCriteriaItem.setAttribute("TargetPartyId");  viewCriteriaItem.setOperator("=");  viewCriteriaItem.getValue().add("300000000616599");  viewCriteriaRow.getItem().add(viewCriteriaItem);  viewCriteriaItem = factory.createViewCriteriaItem();  viewCriteriaItem.setUpperCaseCompare(true);  viewCriteriaItem.setAttribute("StatusCode");  viewCriteriaItem.setOperator("=");  viewCriteriaItem.getValue().add("OPEN");  viewCriteriaRow.getItem().add(viewCriteriaItem);  viewCriteria.getGroup().add(viewCriteriaRow);  findCriteria.setFilter(viewCriteria);  findCriteria.setFetchSize(-1);  FindControl findControl = factory.createFindControl();  findControl.setRetrieveAllTranslations(false);  int start = 0;  while (true) {  findCriteria.setFetchStart(start);  start += 100;  List<Opportunity> resultList =  opportunityService.findOpportunity(findCriteria,  findControl);  for (Opportunity opportunity : resultList) {  returnList.add(opportunity);    }    if (resultList.size() == 0) {  break;  }  break;  }  } catch (Exception e) {  e.printStackTrace();  logger.info(e.getMessage());  }    customFieldHolder.printCustomFieldHolder();    return returnList;  } |