
GLOC
Great Lakes Oracle Conference
Web Service Implementation of Oracle EBS Using Apache, Tomcat, ORDS, & APEX
MAY 15 & 16, 2019
CLEVELAND PUBLIC AUDITORIUM, CLEVELAND, OHIO
WWW.NEOOUG.ORG/GLOC

About the speakers

- **(David) Brian Hartwick**
  - Former IT Delivery Manager at TTI Floor Care NA
  - Currently looking for next opportunity due to Cleveland office closure
  - Hands-on technical IT manager who has been working with Oracle technology since 1998
  - E-mail: dbhartwick@gmail.com
  - http://www.linkedin.com/in/brianhartwick

- **Evelyn Bolton**
  - IT Tech Lead at TTI Floor Care NA
  - IT professional with 25 years of Oracle development experience
  - Currently working with Oracle EBS R12 and APEX v18
  - E-mail: evelyn.bolton@ttifloorcare.com
  - https://www.linkedin.com/in/evelyn-bolton-aa66953b

- Extra thanks to **Juan Rojas**, IT Tech Lead at TTI Floor Care NA for his contributions to the project and presentation!

GLOC
Great Lakes Oracle Conference

Agenda

- Project Overview
- Architecture and Solution
- Documentation and Development
- Testing Process
- Additional Notes and Reference
- Q&A

Principals and Partners

- **About TTI Floor Care North America**
  - The largest floor care business in North America, owning three of the biggest names in floor care (Hoover, Dirt Devil, and Oreck)
- **About Hyland**
  - Industry-leading content services platform and services
  - Hyland Software's OnBase product integrates document management, business process automation and records management
- **About Rackspace**
  - Leading provider of expertise and managed services across all the major public and private cloud technologies

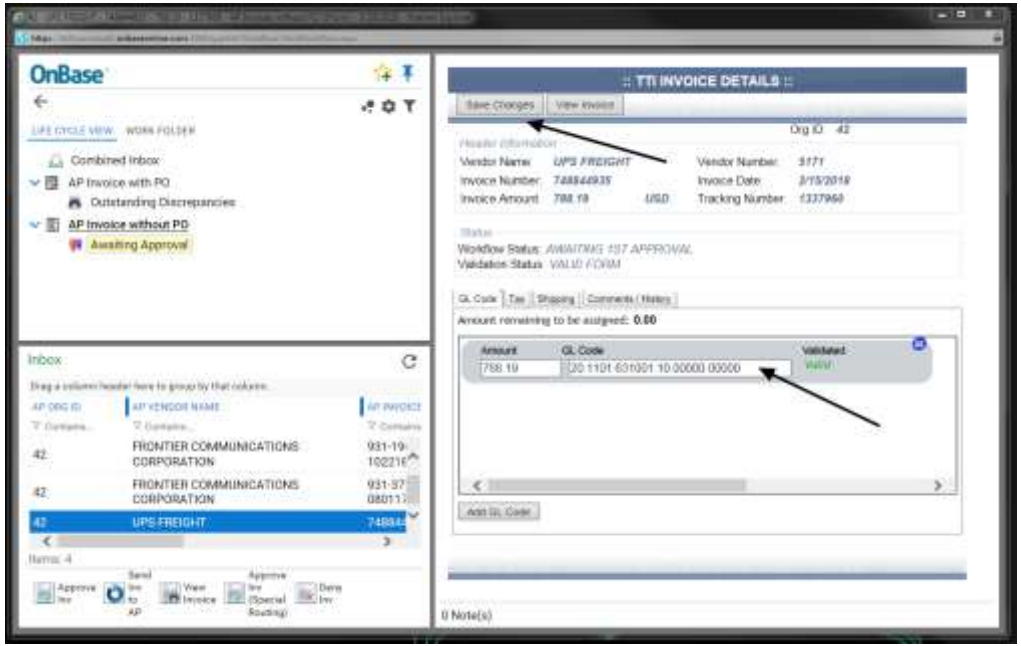
GLOC
Great Lakes Oracle Conference

How does TTI use Hyland OnBase?

- Electronic Document Management
  - Document Retention and Lookup
    - Shipping Documentation (Bill of Lading, Pick Tickets)
    - AR Customer Invoices
    - Bank Reconciliation Files
    - Historical Oracle eBusiness Suite Report Archiving
- Enterprise Enabled Workflow
  - AP Invoice Approval
  - Price Exception Approval

OnBase Workflow AP Invoice Approval – Web Client

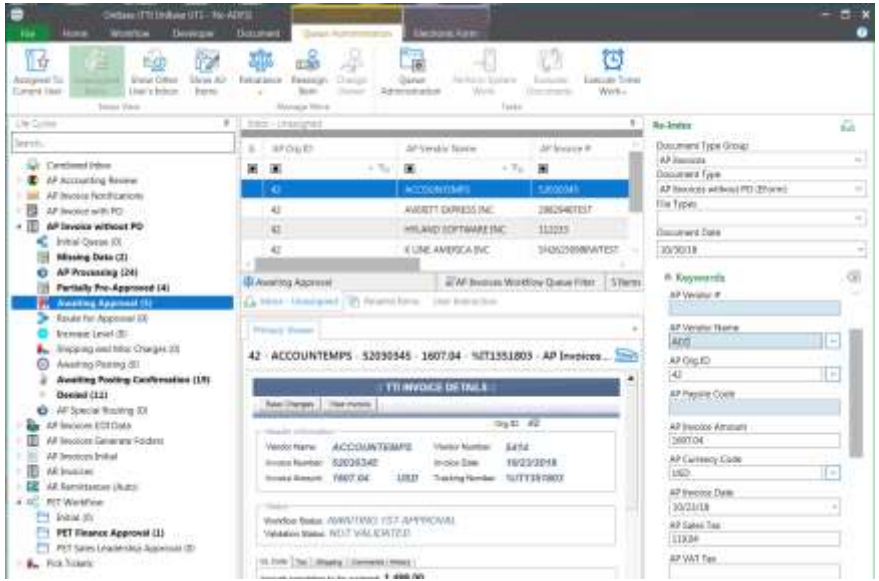
- Users code invoices to correct GL account

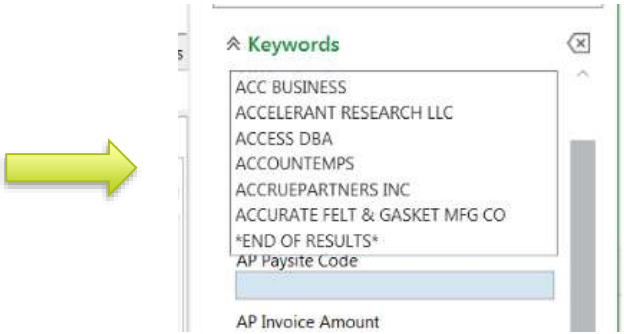


GLOC
Great Lakes Oracle Conference

OnBase Workflow AP Invoice Approval – Unity Client

- Users select vendor and web services GET retrieves vendor information from Oracle EBS





Hyland OnBase Upgrade Project Overview

- Update and host an aged on-premises Hyland OnBase environment (v9) to the latest release of the OnBase application stack (v18)
- The project including migrating the application database from Oracle to Microsoft SQL Server, moving the application and data to the Hyland Cloud, and implementing web services (REST/JSON) with connectivity to our Oracle eBusiness Suite environment using Oracle Rest Data Services (ORDS) with Apache, Apache Tomcat, and Oracle Application Express (APEX)
- The project was successfully implemented on-time and under budget in November 2018

Moving to the Hyland Cloud Documentation

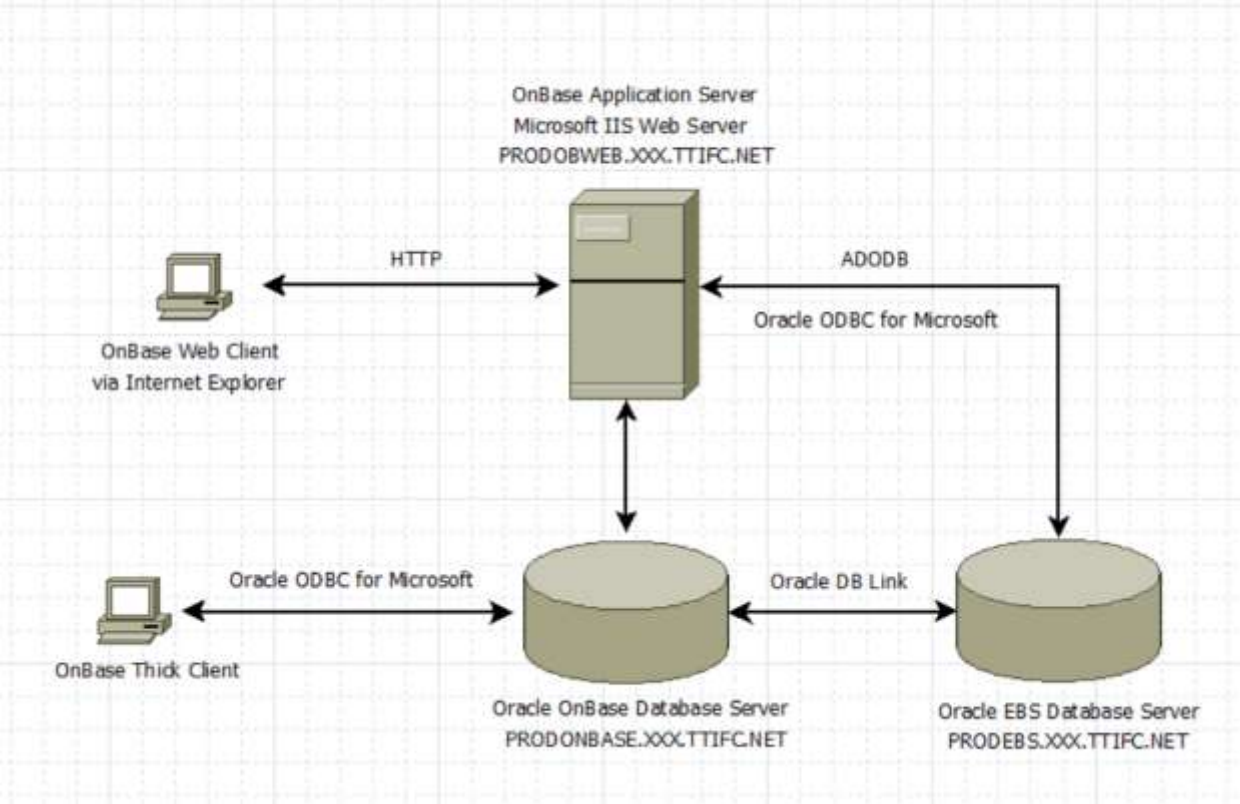
One company
Unlimited possibilities

Hyland

Integrations and Custom Code

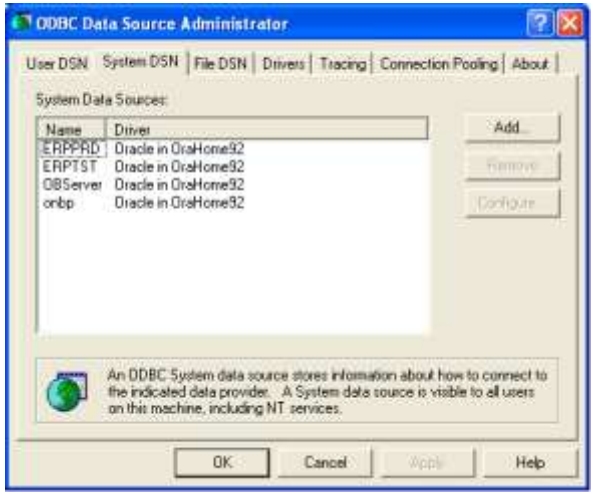
Integrations with other applications are key components of many OnBase systems. Integrations can be challenging when migrating to the cloud. Generally, the biggest challenge is how to communicate between an on-premises application and OnBase in the Hyland Cloud. This will take some planning and testing. Ensure that application integrations have adequate testing plans. OnBase supports web services, and Hyland can host custom applications so most integrations can be accommodated with a hosted OnBase solution. For details on best practices when writing code for the Hyland Cloud, see the GCS Custom Code Guide.

Initial State Architecture

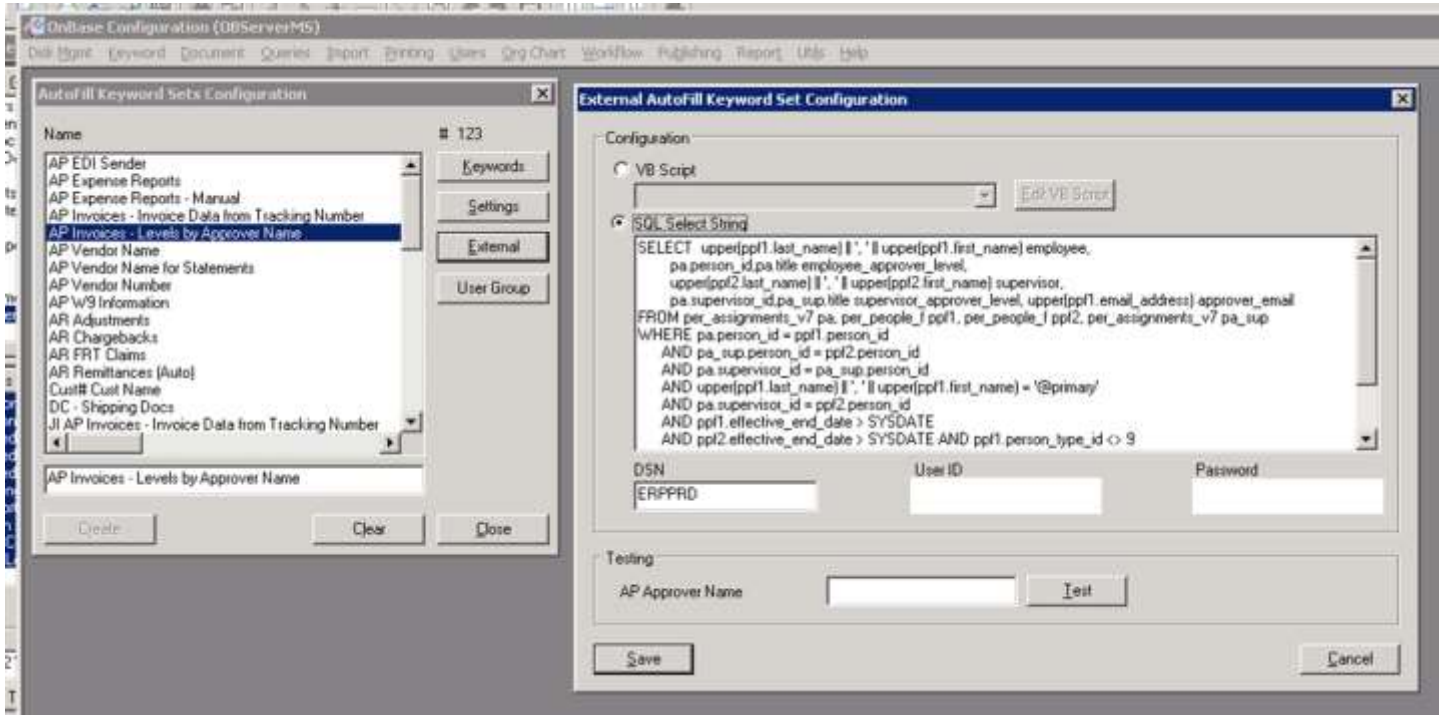


Existing Communication – ODBC and ADODB Connections

- Open Database Connectivity (**ODBC**)
  - OBDC Oracle EBS database called from OnBase Thick Client
- ActiveX Data Objects DB (**ADODB**)
  - ADODB via Microsoft VB Scripts called from OnBase Thick Client
  - ADODB calls from Microsoft ASP Pages called from the Web Client and Server



OBDC to Oracle EBS database via Client



GLOC
Great Lakes Oracle Conference

ADODB via VB Scripts- External Files

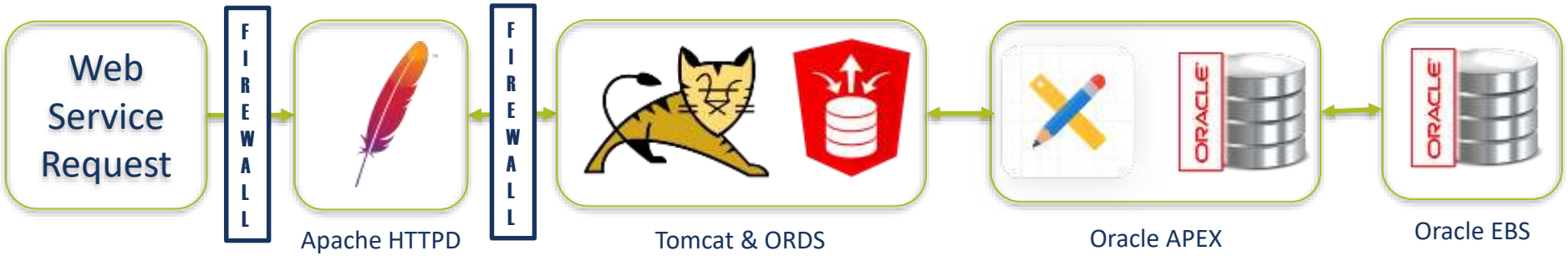
```
Option Explicit

Sub Main35()
    Dim objApp
    Set objApp = CreateObject("OnBase.Application")

    Dim TheForm
    Set TheForm = objApp.CurrentDocument

    Dim TheFormKeys
    Set TheFormKeys = TheForm.Keywords

    Dim FormKeyCount
    FormKeyCount = TheFormKeys.Count()

    Dim conn, rs
    Set conn = CreateObject("ADODB.Connection")
    Set rs = CreateObject("ADODB.recordset")

    conn.Open "DRIVER={Microsoft ODBC for Oracle};SERVER=PTTII;User Id=XXXXXXXXXX;Password=XXXXXXXXXX;"

    Dim SQLText

    Dim VendorNum, InvoiceNum
    Dim i

    For i=0 To FormKeyCount-1
        Select Case TheFormKeys.Item(i).Name
        Case "AP Vendor #"
            VendorNum = TheFormKeys.Item(i).Value
        Case "AP Invoice #"
            InvoiceNum = TheFormKeys.Item(i).Value
        End Select
    Next

    ' Get the zitmid code

    SQLText = "SELECT aia.invoice_num invoice_number, pv.segment1 vendor_number FROM ap.ap_invoices_all aia, ap.ap_suppliers pv WHERE aia.vendor_id =
    pv.vendor_id And pv.segment1 = (VendorNum) And aia.invoice_num = '(InvoiceNum)'"

    SQLText = Replace(SQLText, "(VendorNum)", VendorNum)
    SQLText = Replace(SQLText, "(InvoiceNum)", InvoiceNum)


    rs.Open SQLText, conn
    Dim ResultCount
```

ASP Web Pages – External Web Server

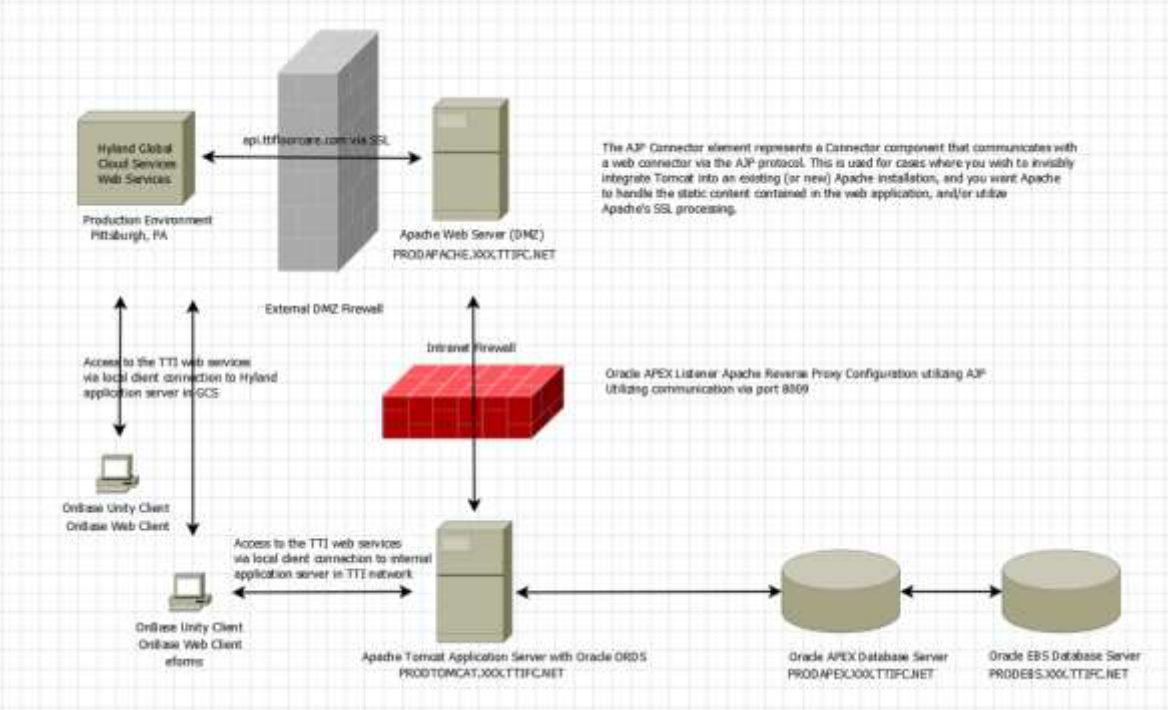
```
<%
response.expires=-1

set conn=Server.CreateObject("ADODB.Connection")
conn.Open "EBSPRD", "XXXXXXXXXX", "XXXXXXXXXX"

set rs=Server.CreateObject("ADODB.recordset")

Dim CodeArray, CodeParts, i, OutText

CodeArray = Split(request.querystring("codes"),"|")

For i=0 to UBound(CodeArray) - 1

    Dim RecordCount
    RecordCount = 0

    CodeParts = Split(CodeArray(i+1), ".")

    if UBound(CodeParts) = 8 Then
        Dim SQLText
        SQLText = "select * from gl.gl_code_combinations where enabled_flag = 'Y' and end_date_active is NULL and segment1='" & CodeParts(0) & "'"
        SQLText = SQLText & " AND segment2='" & CodeParts(1) & "'"
        SQLText = SQLText & " AND segment3='" & CodeParts(2) & "'"
        SQLText = SQLText & " AND segment4='" & CodeParts(3) & "'"
        SQLText = SQLText & " AND segment5='" & CodeParts(4) & "'"
        SQLText = SQLText & " AND segment6='" & CodeParts(5) & "'"
        SQLText = SQLText & " AND segment7='" & CodeParts(6) & "'"
        SQLText = SQLText & " AND segment8='" & CodeParts(7) & "'"
        SQLText = SQLText & " AND segment9='" & CodeParts(8) & "'"


        rs.Open SQLText, conn

        do until rs.EOF
            RecordCount = RecordCount + 1
            rs.MoveNext
        loop
```

How do we move forward?

- Oracle REST Data Services (ORDS)
  - A mid-tier Java application, ORDS maps HTTP(S) verbs (GET, POST, PUT, DELETE, etc.) to database transactions and returns any results formatted using JSON



Final State Architecture



Apache Web Server

- Dedicated Front End Web Server
- Serve static and transfer request
  to Tomcat
- Certificates and network rules
  to secure channel
- Created individual config files
  - HTTPS (SSL) -> Apache (external & internal)
  - HTTP -> Apache & Tomcat
    (internal for initial testing)

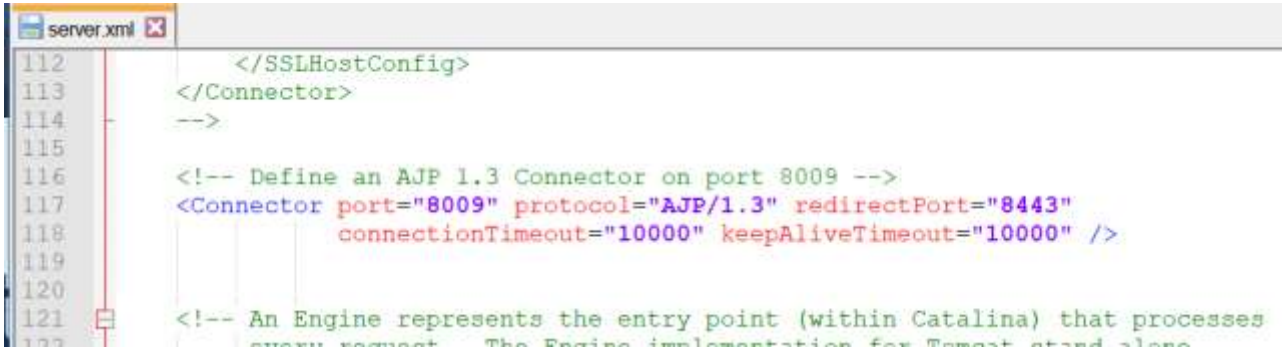
10-apex-ap443.conf

```
# trying to forward SSL to tomcat as well
<VirtualHost *:443>
    ServerName api.ttifloorcare.com
    DocumentRoot /home/tti_apache/www

    SSLEngine on
    SSLCertificateFile "/home/tti_apache/cert/api_mastercert.cer"
    SSLCertificateKeyFile "/home/tti_apache/cert/api.ttifloorcare.com.key"

    # alias for apex image files
    Alias "/i" "/var/www/apex/images/"

    # proxy ORDS requests to tomcat
    ProxyRequests off
    <Location "/api">
        ProxyPass "ajp://PRODTOMCAT.XXX.ttifc.net:8009/ords/api" timeout=1200
        ProxyPassReverse "ajp://PRODTOMCAT.XXX.ttifc.net:8009/ords/api"
    </Location>
</VirtualHost>
```

- Connecting to Tomcat via AJP
  - AJP (Apache JServ Protocol) is basically a binary protocol that allows to reverse proxying requests from a FE Web Server to a BE Application Server, effectively propagating all the needed information to make the Req-Res flow continuing successfully

Apache Tomcat

- Dedicated Application Server
- Manage the HTTP request to JDBC calls
- Minimal configuration changes are required
  - Ensure the AJP is enabled in the server.xml configuration file



Oracle Rest Data Services (ORDS)

- ORDS is a Java application that enables developers with SQL and database skills to develop REST APIs for the Oracle Database
  - Representational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services
- ORDS maps HTTP(S) verbs (GET, POST, PUT, DELETE, etc.) to database transactions and returns any results formatted using JSON
  - **JSON** (JavaScript Object Notation) is a lightweight data-interchange format, language and data independent
- ORDS is included (and for no additional cost) with any Oracle Database license

Apache Tomcat – ORDS Installation - Overview

- The Oracle Rest Data Services (ORDS) installation consists of unzipping the downloaded archive, running the configuration command, and then deploying the ords.war file into the Tomcat web applications folder
  - 1. Unzip ORDS zip file
  - 2. Run installation command line: java -jar ords.jar install advanced
  - 3. In the prompts, select to not create the ORDS schema yet
  - 4. Copy ords.war into webapps
  - 5. Run ORDS schema installation (this step requires the SYS password)
  - 6. If the APEX passwords have not been setup in step 3, run java -jar ords.war setup
  - 7. Unzip images in webapps/i on the Tomcat server
  - 8. Fix Tomcat conf/server.xml to allow some special characters on the URL

```
-->
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           relaxedQueryChars="[,],|" />
```

  - 9. Create users via command line: java –jar ords.war user DEVONBASE “TTI API Client”

Apache Tomcat – ORDS Installation – Part One

- ORDS Installation Log Output Example

```
[tc@PRODTOMCAT ords]$ java -jar ords.war install advanced
This Oracle REST Data Services instance has not yet been configured.
Please complete the following prompts

Enter the location to store configuration data:config
Enter the name of the database server [localhost]:psmlttiapx01.tcs.ttifc.net
Enter the database listen port [1521]:1521
Enter 1 to specify the database service name, or 2 to specify the database SID [1]:2
Enter the database SID [xe]:PTTIAPX
Enter 1 if you want to verify/install Oracle REST Data Services schema or 2 to skip this step [1]:2
Enter 1 if you want to use PL/SQL Gateway or 2 to skip this step.
If using Oracle Application Express or migrating from mod_plsql then you must enter 1 [1]:2
Sep 25, 2018 8:34:10 AM
INFO: Updated configurations: defaults
Enter 1 if you wish to start in standalone mode or 2 to exit [1]:2
```

Apache Tomcat – ORDS Installation – Part Two

- ORDS Schema Installation Log Output Example

```
[tc@PRODTOMCAT ords]$ java -jar ords.war schema
Enter the name of the database server [psmlttiapx01.tcs.ttifc.net]:psmlttiapx01.tcs.ttifc.net
Enter the database listen port [1521]:
Enter 1 to specify the database service name, or 2 to specify the database SID [1]:2
Enter the database SID [PTTIAPX]:
Requires SYS AS SYSDBA to verify Oracle REST Data Services schema.

Enter the database password for SYS AS SYSDBA:
Confirm password:

Retrieving information.
Enter the default tablespace for ORDS_METADATA [SYSAUX]:
Enter the temporary tablespace for ORDS_METADATA [TEMP]:
Enter the default tablespace for ORDS_PUBLIC_USER [USERS]:
Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP]:
Enter the database password for ORDS_PUBLIC_USER:
Confirm password:
Installing Oracle REST Data Services version 18.2.0.r1831332
... Log file written to /opt/tomcat/ords_install_core_2018-09-25_090317_00671.log
... Verified database prerequisites
... Created Oracle REST Data Services schema
... Created Oracle REST Data Services proxy user
... Granted privileges to Oracle REST Data Services
... Created Oracle REST Data Services database objects
... Log file written to /opt/tomcat/ords_install_datamodel_2018-09-25_090334_00961.log
... Log file written to /opt/tomcat/ords_install_apex_2018-09-25_090337_00222.log
Completed installation for Oracle REST Data Services version 18.2.0.r1831332. Elapsed time: 00:00:21.528
```

GLOC
Great Lakes Oracle Conference

Apache Tomcat – ORDS Installation – Part Three

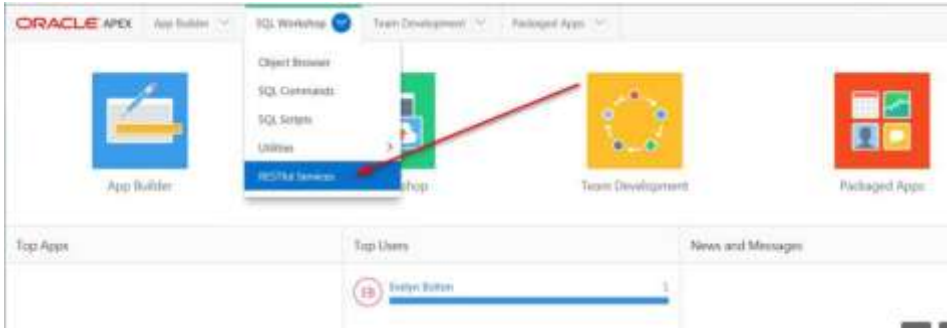
- Database Setup Log Output Example

```
[tc@PRODTOMCAT ords]$ java -jar ords.war setup
Enter the name of the database server [psmlttiapx01.tcs.ttifc.net]:
Enter the database listen port [1521]:
Enter 1 to specify the database service name, or 2 to specify the database SID [1]:2
Enter the database SID [PTTIAPX]:
Enter 1 if you want to verify/install Oracle REST Data Services schema or 2 to skip this step [1]:2
Enter 1 if you want to use PL/SQL Gateway or 2 to skip this step.
If using Oracle Application Express or migrating from mod_plsql then you must enter 1 [1]:1
Enter the PL/SQL Gateway database user name [APEX_PUBLIC_USER]:
Enter the database password for APEX_PUBLIC_USER:
Confirm password:
Enter 1 to specify passwords for Application Express RESTful Services database users (APEX_LISTENER, APEX_REST_PUBLIC_USER) or 2
to skip this step [1]:1
Enter the database password for APEX_LISTENER:
Confirm password:
Enter the database password for APEX_REST_PUBLIC_USER:
Confirm password:
Sep 25, 2018 9:06:19 AM
INFO: Updated configurations: defaults, apex, apex_al, apex_rt
```

Oracle APEX – Stand Alone Applications

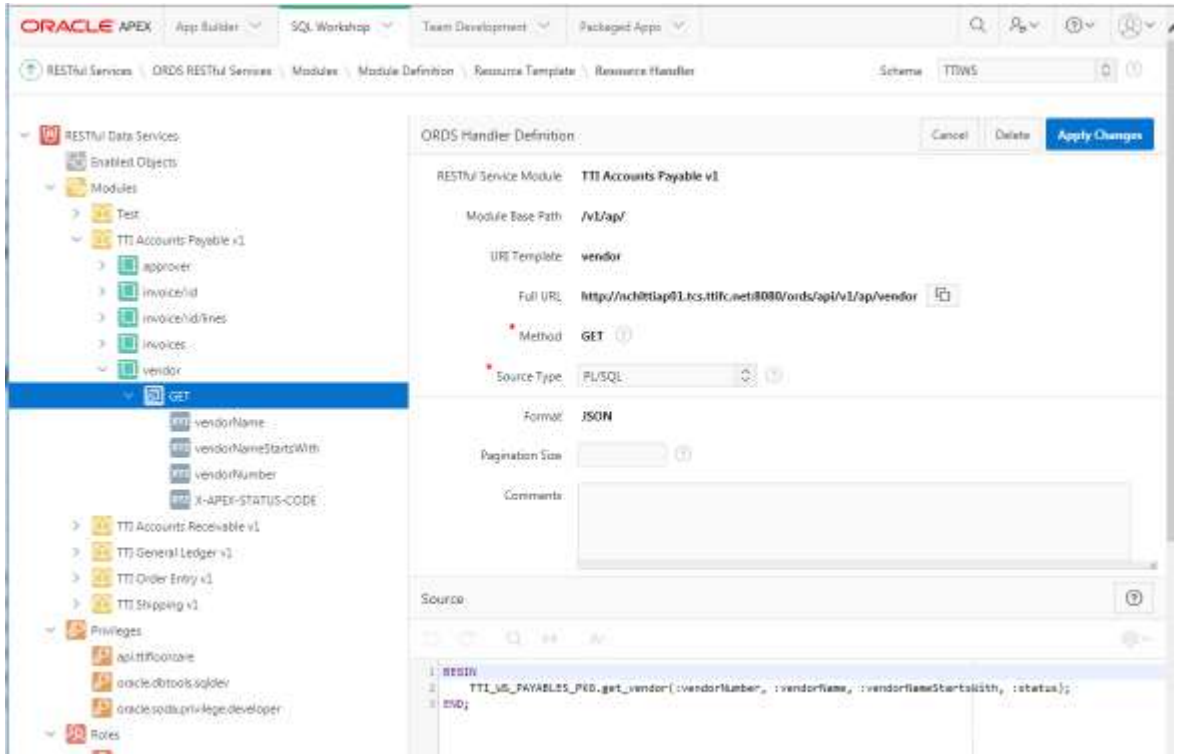
- Leverage an existing APEX environment for connectivity
  - User maintenance APEX application for a sales commission application
  - Reports and data extracts for finance and accounting teams
  - “Peanut Butter Factory” – Sales allocation web application

Oracle APEX – Restful Data Services



GLOC
Great Lakes Oracle Conference

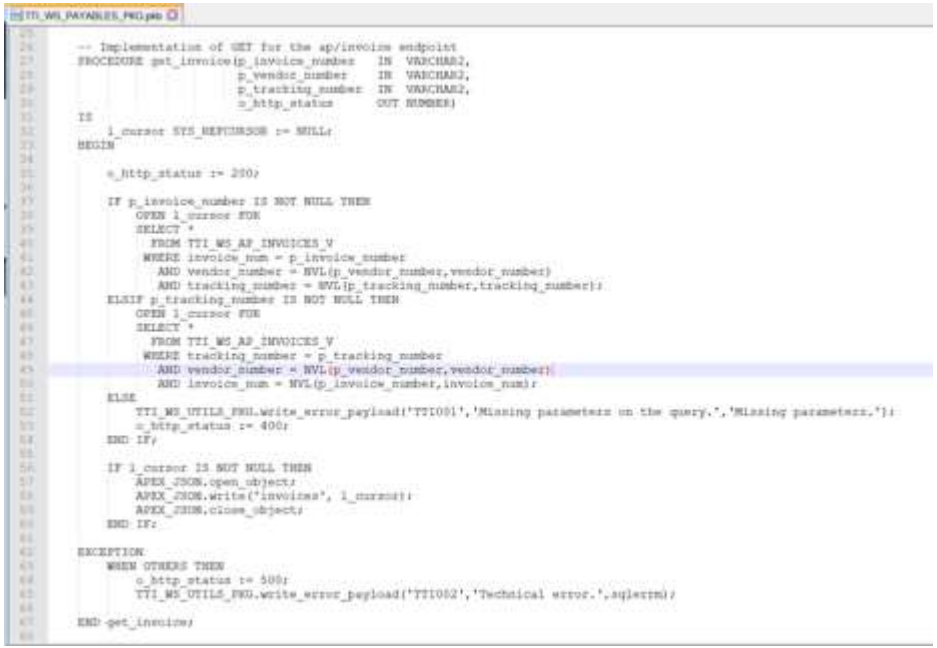
Oracle APEX – Resource Handler



GLOC
Great Lakes Oracle Conference

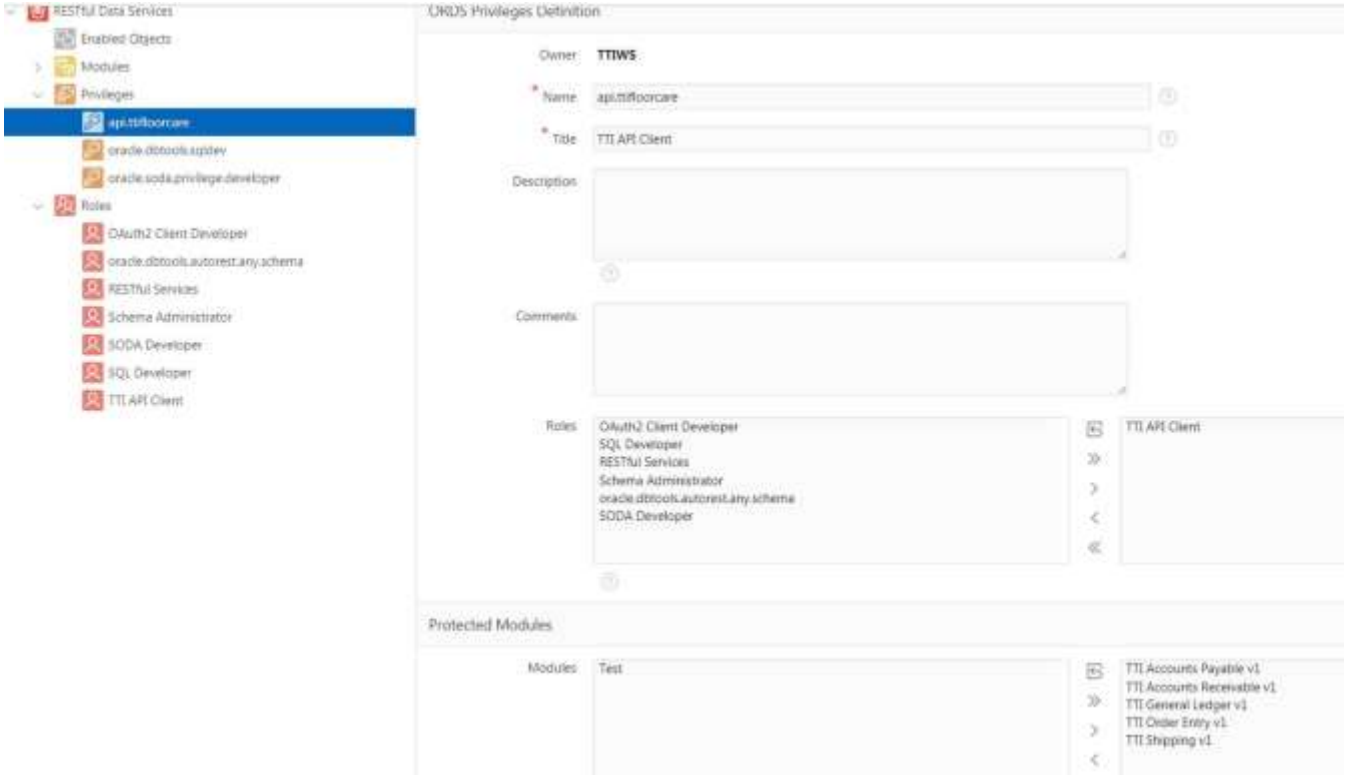
Leverage Existing APEX Packages – APEX_JSON

- Custom PL/SQL package for business logic can access built-in JSON packages that are part of APEX



```
-- Implementation of GET for the ap/invoice endpoint
PROCEDURE get_invoice(p_invoice_number   IN  VARCHAR2,
                      p_vendor_number    IN  VARCHAR2,
                      p_tracking_number  IN  VARCHAR2,
                      o_http_status      OUT NUMBER)
IS
    l_cursor SYS_REFCURSOR := NULL;
BEGIN

    o_http_status := 200;

    IF p_invoice_number IS NOT NULL THEN
        OPEN l_cursor FOR
        SELECT *
          FROM TTI_WS_AP_INVOICES_V
         WHERE invoice_num = p_invoice_number
           AND vendor_number = NVL(p_vendor_number,vendor_number)
           AND tracking_number = NVL(p_tracking_number,tracking_number);
    ELSIF p_tracking_number IS NOT NULL THEN
        OPEN l_cursor FOR
        SELECT *
          FROM TTI_WS_AP_INVOICES_V
         WHERE tracking_number = p_tracking_number
           AND vendor_number = NVL(p_vendor_number,vendor_number)
           AND invoice_num = NVL(p_invoice_number,invoice_num);
    ELSE
        TTI_WS_UTILS_PKG.write_error_payload('TTI001','Missing parameters on the query.','Missing parameters.');
        o_http_status := 400;
    END IF;

    IF l_cursor IS NOT NULL THEN
        APEX_JSON.open_object;
        APEX_JSON.write('invoices', l_cursor);
        APEX_JSON.close_object;
    END IF;

EXCEPTION
    WHEN OTHERS THEN
        o_http_status := 500;
        TTI_WS_UTILS_PKG.write_error_payload('TTI002','Technical error.',sqlerrm);

END get_invoice;
```

Oracle APEX - Privileges



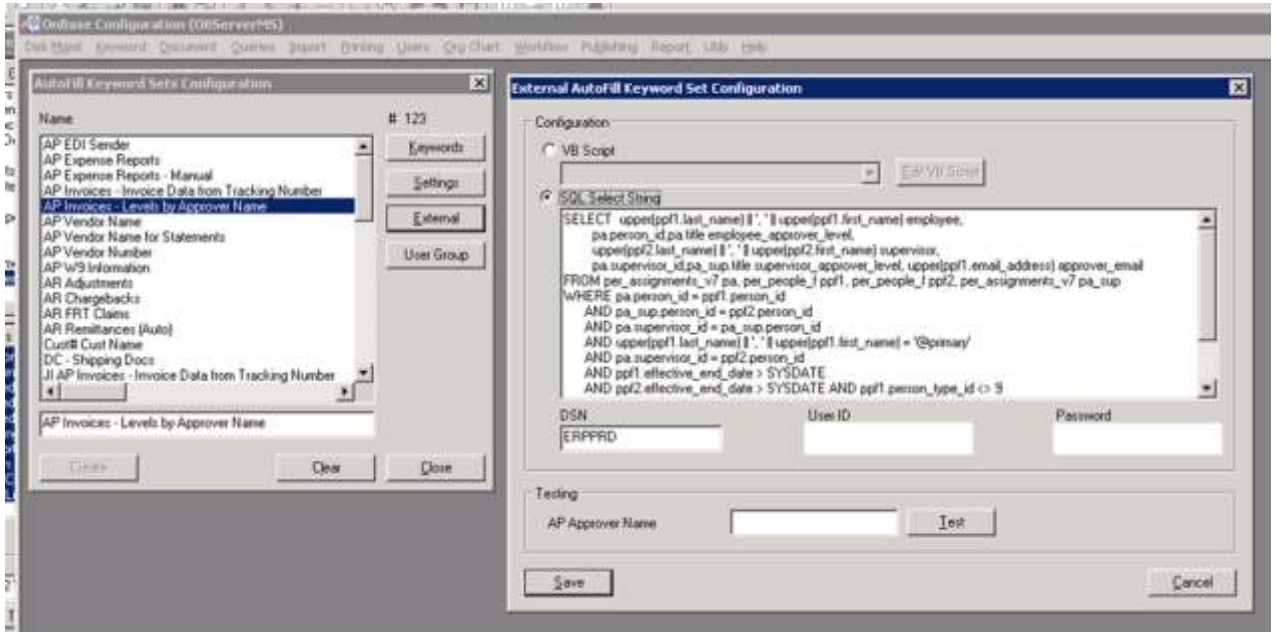
GLOC
Great Lakes Oracle Conference

Oracle eBusiness Suite

- Oracle APEX offers you a quick and highly productive way to extend your Oracle E-Business Suite environment with almost no impact to your existing implementation
- Our Oracle APEX server has limited access to the Oracle EBS via explicit grants to a dedicated custom user/schema (XXAPEX_EBS_EXT)
  - GRANT SELECT ON APPS.AP_INVOICES_ALL TO XXAPEX_EBS_EXT;
  - GRANT SELECT ON APPS.AP_INVOICE_LINES_ALL TO XXAPEX_EBS_EXT;
  - GRANT SELECT ON APPS.TTI_WS_AP_INVOICES_V TO XXAPEX_EBS_EXT;
  - GRANT SELECT ON APPS.TTI_WS_AP_APPROVERS_V TO XXAPEX_EBS_EXT;
  - GRANT SELECT ON APPS.TTI_WS_AP_VENDORS_V TO XXAPEX_EBS_EXT;

Documentation & Development – Existing Definitions

- We were able to leverage the existing queries and database calls to provide a specification for the web services that we needed to develop



Existing SQL Query to Web Service Definition

```sql
-- AP Invoices - Levels by Approver Name
SELECT  upper(ppf1.last_name) || ', ' || upper(ppf1.first_name) employee,
        pa.person_id,pa.title employee_approver_level,
        upper(ppf2.last_name) || ', ' || upper(ppf2.first_name) supervisor,
        pa.supervisor_id,pa_sup.title supervisor_approver_level, upper(ppf1.email_address) approver_email
FROM per_assignments_v7 pa, per_people_f ppf1, per_people_f ppf2, per_assignments_v7 pa_sup
WHERE pa.person_id = ppf1.person_id
      AND pa_sup.person_id = ppf2.person_id
      AND pa.supervisor_id = pa_sup.person_id
      AND upper(ppf1.last_name) || ', ' || upper(ppf1.first_name) = :p_full_name
      AND pa.supervisor_id = ppf2.person_id
      AND ppf1.effective_end_date > SYSDATE
      AND ppf2.effective_end_date > SYSDATE AND ppf1.person_type_id <> 9
      AND nvl(pa.title,'Z') <> 'Z';
```

We need the JSON!

Name:	**AP Approver**		
Description:	Get details for an employee regarding AP approval limits and supervisor's approval limits.		
URL:	/api/v1/ap/approver		
HTTP Method:	GET		
HTTP Header Parameters:	**Parameter Name**	**Type**	**Comments**
	employeeName	String	Required unless employeeNameStartsWith is provided. Format: "Last Name, First Name".
	employeeNameStartsWith	String	Required unless employeeName is provided. Format: "Last Name, First Name". Performs a LIKE-type of search on employees.
Example Request:			

Example Response:	<pre>{
	  "approvers": [
	  {
	  "EMPLOYEE": "ROJAS, JUAN",
	  "PERSON_ID": 18881,
	  "EMPLOYEE_APPROVER_LEVEL": "L0",
	  "SUPERVISOR": "HUDDY, CHRIS",
	  "SUPERVISOR_ID": 18169,
	  "SUPERVISOR_APPROVER_LEVEL": "L10",
	  "APPROVER_EMAIL": "JUAN.ROJAS@TTIFLOORCARE.COM"
	  }
	  ]
	}</pre>		

Testing your web services - CURL

- CURL from the command line
  - C:\obdebug\curl\bin>curl -u devonbase:XXXXXXXXXXXX http://PRODAPACHE.XXX.ttifc.net/api/v1/wsh/pickslip?deliveryId=20412442


```
{
  "pickslips":[
    {
      "DELIVERY_ID":20412442,
      "ACCOUNT_NUMBER":"01384",
      "PARTY_NAME":"AMAZON SUPPLY",
      "SOURCE_HEADER_NUMBER":"10579254",
      "TRIP_ID":11123582,
      "ORGANIZATION_CODE":"ODC",
      "ATTRIBUTE3_DATE":"2018-09-11T00:00:00Z",
      "CUST_PO_NUMBER":"3ETTNNCY",
      "ATTRIBUTE1":"00416830084820082",
      "ATTRIBUTE4":"00416830084820075"
    }
  ]
}
```

Testing your web services - Powershell

- Using Invoke-WebRequest cmdlet, script many connection tests and capture JSON and timing

endpoints.csv

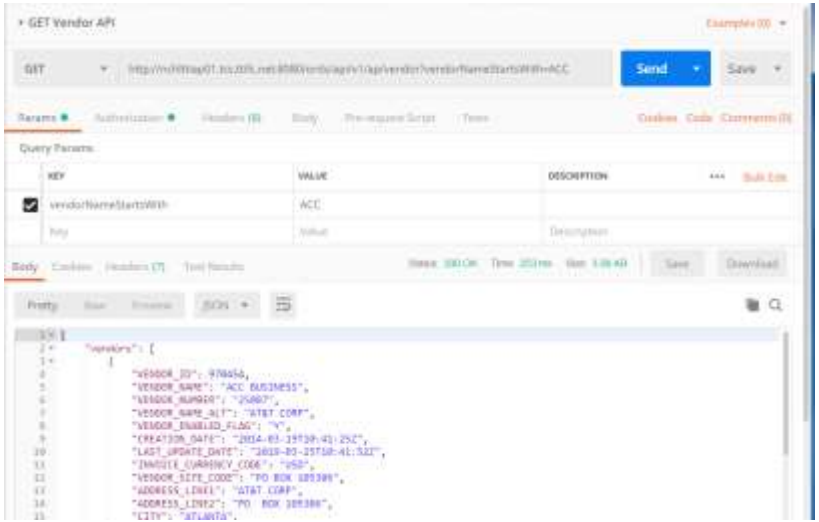
```
URL
https://11.222.333.44/api/v1/wsh/picklip?deliveryId=20412442
https://11.222.333.44/api/v1/wsh/picklip?deliveryId=20412442
https://PRODAPACHE.xxx.ttifc.net/api/v1/wsh/picklip?deliveryId=20412442
https://PRODAPACHE.xxx.ttifc.net/api/v1/wsh/picklip?deliveryId=20412442
https://api.ttiflooreare.com/api/v1/wsh/picklip?deliveryId=20412442
https://api.ttiflooreare.com/api/v1/wsh/picklip?deliveryId=20412442
https://PRODAPACHE.xxx.ttifc.net/api/v1/gl/account?fullAccount=20.0000.223503.10.00000.00000
https://PRODAPACHE.xxx.ttifc.net/api/v1/gl/account?fullAccount=20.0000.223503.10.00000.00000
https://PRODTOMCAT.xxx.ttifc.net:8080/orda/api/v1/wsh/picklip?deliveryId=20412442
```

invoke.ps1

```
Write-Output '-------------------------------------------------------------'
Write-Output '-- prod environment web service testing ---------------------'
Write-Output '-------------------------------------------------------------'
$start=Get-Date
Write-Output $start
Write-Output '-------------------------------------------------------------'

$stopwatch =  [system.diagnostics.stopwatch]::StartNew()

$user = "devonhase"
$pass= "XXXXXXXXXXXXXX"
$pair = "$($user):$($pass)"

$encodedCreds = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($pair))

$basicAuthValue = "Basic $encodedCreds"

$Headers = @{
    Authorization = $basicAuthValue
}

$csv = Import-Csv c:\obdebug\endpoints-pro.csv
foreach ($line in $csv) {
     Write-Output '&&-------------------------------------------------------&&'
     $root = $line.URL
     Write-Output $root
     Invoke-WebRequest -Uri $root -Headers $Headers | Select StatusCode, Content
}

Write-Output "Total Seconds: "
$stopwatch.Elapsed.TotalSeconds
$stopwatch.Stop()

Write-Output '-------------------------------------------------------------'
Write-Output '-------------------------------------------------------------'
$end=Get-Date
Write-Output $end
Write-Output '-------------------------------------------------------------'
```

Testing your web services - Postman

- Use **Postman** to facilitate testing of your web services
  - Support for authentication, header variables, scripts, etc.
  - Essential tool for testing POST/PUT/DELETE



POSTMAN

GLOC
Great Lakes Oracle Conference

Demo of Working Final Solution

Additional Key References

- **Oracle Database XE, APEX, ORDS, Tomcat and httpd on CentOS 7: all-in-one guide – introduction by Denis Savenko**
  - https://dsavenko.me/oracledb-apex-ords-tomcat-httpd-centos7-all-in-one-guide-introduction/
- **Extending Oracle E-Business Suite Release 12 using Oracle APEX by Oracle**
  - https://www.oracle.com/technetwork/developer-tools/apex/learnmore/apex-ebs-extension-white-paper-345780.pdf

Questions?

GLOC
Great Lakes Oracle Conference