



# Web Service Implementation of Oracle EBS Using Apache, Tomcat, ORDS, & APEX

MAY 15 & 16, 2019

CLEVELAND PUBLIC AUDITORIUM, CLEVELAND, OHIO

[WWW.NEOOUG.ORG/GLOC](http://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!

# 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



# 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

The screenshot displays the OnBase web client interface. On the left, the 'OnBase' logo is at the top, followed by navigation options like 'LIFE CYCLE VIEW' and 'WORK FOLDER'. Below this is a 'Combined Inbox' section with a tree view containing 'AP Invoice with PO', 'Outstanding Discrepancies', and 'AP Invoice without PO' (with a sub-item 'Awaiting Approval'). An 'Inbox' section below shows a list of invoices with columns for 'AP INV ID', 'AP VENDOR NAME', and 'AP INVOICE'. The selected item is '43 UPS-FREIGHT 74894'. At the bottom left are buttons for 'Approve', 'Send to AP', 'View Invoice', 'Approve (Special Routing)', and 'Deny Inv'.

The main area is titled 'TTI INVOICE DETAILS'. It has tabs for 'Basic Changes' and 'View Invoice'. The 'Basic Changes' tab is active, showing vendor information: Vendor Name: UPS-FREIGHT, Vendor Number: 3771, Invoice Number: 748944935, Invoice Date: 3/15/2018, Invoice Amount: 788.19 USD, and Tracking Number: 6327960. Below this is the 'Status' section: Workflow Status: AWAITING #57 APPROVAL, Validation Status: VALID FORM. A section for 'GL Code' has tabs for 'Tax', 'Shipping', and 'Comments (Notes)'. It shows 'Amount remaining to be assigned: 0.00' and a table with columns 'Amount', 'GL Code', and 'Validated'. The table contains one row: Amount: 788.19, GL Code: 20 1101 631001 10 0000 0000, Validated: VALID. An arrow points to the 'Validated' cell. At the bottom is a 'Add GL Code' button.

# OnBase Workflow AP Invoice Approval – Unity Client

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

The screenshot shows the OnBase Workflow AP Invoice Approval interface. The main window displays a table of invoices with columns for AP Org ID, AP Vendor Name, and AP Invoice #. The table contains the following data:

AP Org ID	AP Vendor Name	AP Invoice #
42	ACCOUNTEMPS	52090545
42	ANGRETT EXPRESS INC.	2862947657
42	INFLAIR SOFTWARE INC.	112223
42	LINE AMERICA INC.	34002988847657

Below the table, the 'Awaiting Approval' section shows details for the selected invoice: 42 - ACCOUNTEMPS - 52090545 - 1607.04 - NET135.1805 - AP Invoice... The 'Invoice Details' section shows the following information:

Vendor Name: ACCOUNTEMPS Vendor Number: 64164  
Invoice Number: 52090545 Invoice Date: 16/02/2018  
Invoice Amount: 1607.04 USD Tracking Number: 10771392800

The 'Workflow Status' section shows: Workflow Status: APPROVAL (527 APPROVAL)  
Validation Status: NOT VALIDATED



The screenshot shows the 'Keywords' list in the OnBase interface. The list contains the following keywords:

- ACC BUSINESS
- ACCELERANT RESEARCH LLC
- ACCESS DBA
- ACCOUNTEMPS
- ACCURPARTNERS INC
- ACCURATE FELT & GASKET MFG CO
- \*END OF RESULTS\*
- AP Payscale Code
- AP Invoice Amount

# 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

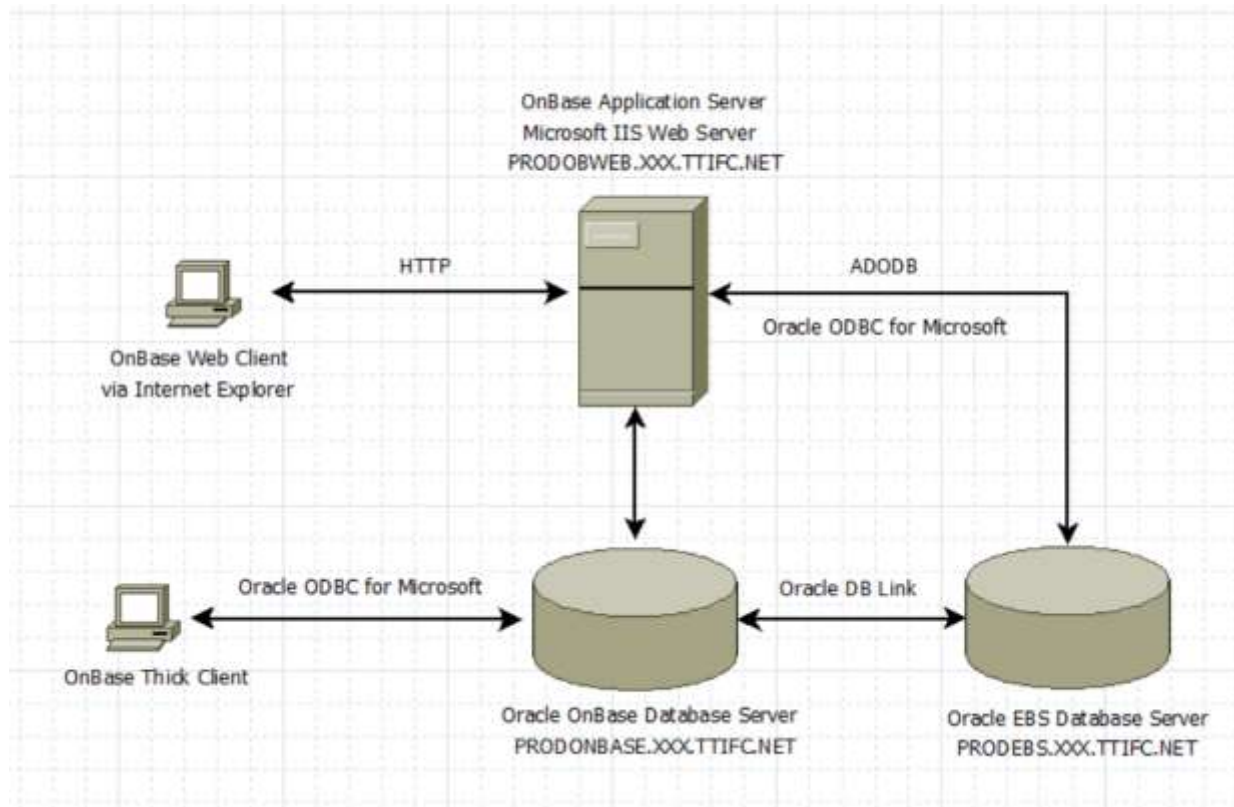
One company  
Unlimited possibilities



## Integrations and Custom Code

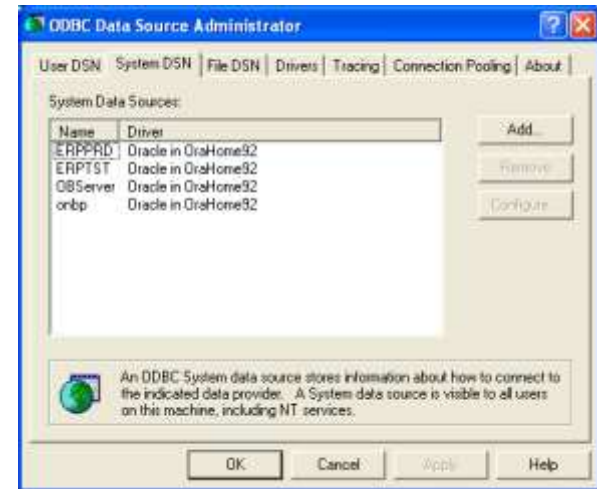
Integrations with other applications are key components of many OnBase systems. Integrations can be challenging when migrating to the cloud. Generally, the biggest challenge is how to communicate between an on-premises application and OnBase in the Hyland Cloud. This will take some planning and testing. 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**)
  - ODBC 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

The screenshot displays the Oracle OBDC Configuration dialog box, titled "Oracle Configuration (OBServerM5)". The main window shows a list of keyword sets on the left, with "AP Invoices - Levels by Approver Name" selected. The right pane is titled "External AutoFill Keyword Set Configuration" and contains the following configuration details:

- Configuration:**
  - VB Script
  - SQL Select String
- SQL Select String:**

```
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) = '@primary/  
      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
```
- DSN:** ERPPRD
- User ID:** [Empty field]
- Password:** [Empty field]
- Testing:** AP Approver Name [Empty field] [Test]
- Buttons:** [Save] [Cancel]

# ADODB via VB Scripts- External Files

```
1 Option Explicit
2
3 Sub Main35 ()
4     Dim objApp
5     Set objApp = CreateObject("DnBase.Application")
6
7     Dim TheForm
8     Set TheForm = objApp.CurrentDocument
9
10    Dim TheFormKeys
11    Set TheFormKeys = TheForm.Keywords
12
13    Dim FormKeyCount
14    FormKeyCount = TheFormKeys.Count()
15
16    Dim conn, rs
17    Set conn = CreateObject("ADODB.Connection")
18    Set rs = CreateObject("ADODB.Recordset")
19
20    conn.Open "DRIVER={Microsoft ODBC for Oracle};SERVER=PTIII;DESC ID=XXXXXXXXXX;PASSWORD=XXXXXXXXXX"
21
22    Dim SQLText
23
24    Dim VendorNum, InvoiceNum
25    Dim i
26
27    For i=0 To FormKeyCount-1
28        Select Case TheFormKeys.Item(i).Name
29            Case "AF Vendor #"
30                VendorNum = TheFormKeys.Item(i).Value
31            Case "AF Invoice #"
32                InvoiceNum = TheFormKeys.Item(i).Value
33        End Select
34    Next
35
36    ' Get the sitaid code
37
38    SQLText = "SELECT aia.invoice_num invoice_numbez, pv.segment1 vendor_number FROM ap.ap_invoices_all aia, ap.ap_suppliers pv WHERE aia.vendor_id =
39    pv.vendor_id And pv.segment1 = (VendorNum) And aia.invoice_num = (InvoiceNum)"
40
41    SQLText = Replace(SQLText, "(VendorNum)", VendorNum)
42    SQLText = Replace(SQLText, "(InvoiceNum)", InvoiceNum)
43
44    rs.Open SQLText, conn
45    Dim RowsetCount
```

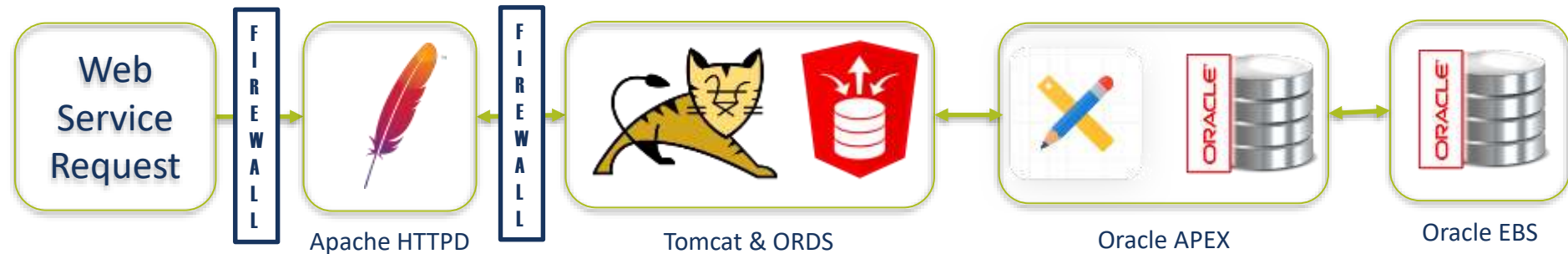


# ASP Web Pages – External Web Server

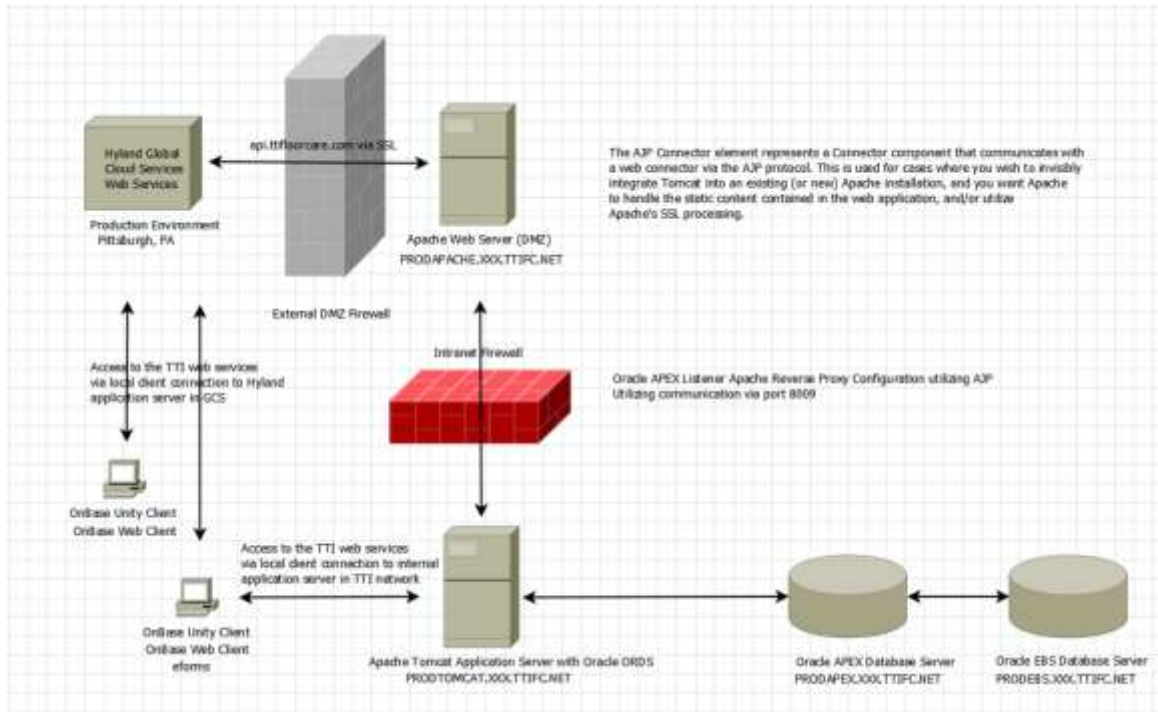
```
1 <!--
2 response.expires=0
3
4 set conn=Server.CreateObject("ADODB.Connection")
5 conn.Open "DSN=XXXXX", "XXXXXXXXXX", "XXXXXXXXXX"
6
7 set rs=Server.CreateObject("ADODB.Recordset")
8
9 Dim CodeArray, CodeParts, i, OutText
10
11 CodeArray = Split(request.querystring("codes"),",")
12
13 For i=0 to UBound(CodeArray) - 1
14
15     Dim RecordCount
16     RecordCount = 0
17
18     CodeParts = Split(CodeArray[i+1], ".")
19
20     if UBound(CodeParts) = 8 Then
21         Dim SQLText
22         SQLText = "select * from gl_gl_code_combinations where enabled_flag = 'Y' and end_date_active is NULL and segment1='" & CodeParts(0) & "'
23         SQLText = SQLText & " AND segment2=''" & CodeParts(1) & "'
24         SQLText = SQLText & " AND segment3=''" & CodeParts(2) & "'
25         SQLText = SQLText & " AND segment4=''" & CodeParts(3) & "'
26         SQLText = SQLText & " AND segment5=''" & CodeParts(4) & "'
27         SQLText = SQLText & " AND segment6=''" & CodeParts(5) & "'
28         SQLText = SQLText & " AND segment7=''" & CodeParts(6) & "'
29         SQLText = SQLText & " AND segment8=''" & CodeParts(7) & "'
30         SQLText = SQLText & " AND segment9=''" & CodeParts(8) & "'
31
32
33         rs.Open SQLText, conn
34
35         do until rs.EOF
36             RecordCount = RecordCount + 1
37             rs.MoveNext
38         loop
39     end if
40 end For
41 -->
```

# 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)
- 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

```
10-apache-aps443.conf
1 # trying to forward SSL to tomcat as well
2 <VirtualHost *:443>
3     ServerName api.ttifloorcare.com
4     DocumentRoot /home/tti_apache/www
5
6
7     SSLEngine on
8     SSLCertificateFile "/home/tti_apache/cert/api_mastercert.cer"
9     SSLCertificateKeyFile "/home/tti_apache/cert/api.ttifloorcare.com.key"
10
11     # alias for apex image files
12     Alias "/i" "/var/www/apex/images/"
13
14     # proxy ORDS requests to tomcat
15     ProxyRequests off
16     <Location "/api">
17         ProxyPass "ajp://PRODTOMCAT.XXX.ttifc.net:8009/ords/api" timeout=1200
18         ProxyPassReverse "ajp://PRODTOMCAT.XXX.ttifc.net:8009/ords/api"
19     </Location>
20 </VirtualHost>
21
```

# 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

```
112     </SSLHostConfig>
113 </Connector>
114 -->
115
116 <!-- Define an AJP 1.3 Connector on port 8009 -->
117 <Connector port="8009" protocol="AJP/1.3" redirectPort="8443"
118           connectionTimeout="10000" keepAliveTimeout="10000" />
119
120
121 <!-- An Engine represents the entry point (within Catalina) that processes
122     every request. The Engine implementation for Tomcat stand alone
```



# 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
```

# Apache Tomcat – ORDS Installation – Part Three

- Database Setup Log Output Example

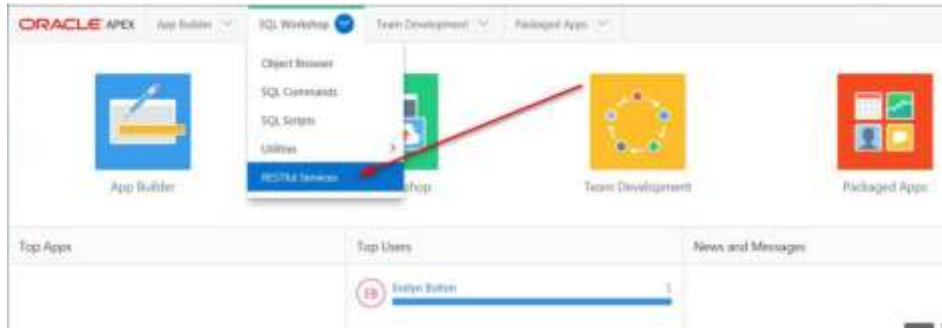
```
[tcs@PRODTOMCAT ords]$ java -jar ords.war setup
Enter the name of the database server [psmlttiapx01.tcs.ttific.net]:
Enter the database listen port [1521]:
Enter 1 to specify the database service name, or 2 to specify the database SID [1]:2
Enter the database SID [PTTIAPX]:
Enter 1 if you want to 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



# Oracle APEX – Resource Handler

The screenshot shows the Oracle APEX interface for configuring a Resource Handler. The left-hand navigation pane shows a tree structure under 'RESTful Data Services' > 'Modules' > 'TTI Accounts Payable v1' > 'GET'. The right-hand pane is titled 'ORDS Handler Definition' and contains the following configuration:

- RESTful Service Module:** TTI Accounts Payable v1
- Module Base Path:** /v1/ap/
- URI Template:** vendor
- Full URL:** http://nd01ap01.tcs.tti.com/ords/ap/v1/ap/vendor
- Method:** GET
- Source Type:** PL/SQL
- Format:** JSON
- Pageation Size:** (empty field)
- Comments:** (empty text area)

At the bottom, the 'Source' code editor shows the following PL/SQL code:

```
1 BEGIN
2   TTI_US_PAYABLES_PKG.get_vendor(:vendorNumber, :vendorName, :vendorNameStartsWith, :status);
3 END;
```

# Leverage Existing APEX Packages – APEX\_JSON

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

```
1  -- Implementation of GET for the ap/invoices endpoint
2  PROCEDURE get_invoice(p_invoice_number IN VARCHAR2,
3                       p_vendor_number IN VARCHAR2,
4                       p_tracking_number IN VARCHAR2,
5                       o_http_status OUT NUMBER)
6  IS
7     i_cursor SYS_REFCURSOR := NULL;
8  BEGIN
9     o_http_status := 200;
10
11     IF p_invoice_number IS NOT NULL THEN
12         OPEN i_cursor FOR
13             SELECT *
14             FROM TTI_WS_AP_INVOICES_V
15             WHERE invoice_num = p_invoice_number
16                AND vendor_number = NVL(p_vendor_number, vendor_number)
17                AND tracking_number = NVL(p_tracking_number, tracking_number);
18     ELSEIF p_tracking_number IS NOT NULL THEN
19         OPEN i_cursor FOR
20             SELECT *
21             FROM TTI_WS_AP_INVOICES_V
22             WHERE tracking_number = p_tracking_number
23                AND vendor_number = NVL(p_vendor_number, vendor_number)
24                AND invoice_num = NVL(p_invoice_number, invoice_num);
25     ELSE
26         TTI_WS_UTILS_PUB.write_error_payload('TTI001', 'Missing parameters on the query.', 'Missing parameters.');
```

# Oracle APEX - Privileges

The screenshot shows the Oracle APEX interface for defining RESTful Data Services (ORDS) privileges. The left-hand navigation pane is expanded to show the 'Privileges' section, with the specific privilege 'api.ttlfloorcare' selected. The main area is titled 'ORDS Privileges Definition' and contains the following fields:

- Owner:** TTIWS
- Name:** api.ttlfloorcare
- Title:** TTI API Client
- Description:** (Empty text area)
- Comments:** (Empty text area)
- Roles:** A list of roles is shown on the left, including Oauth2 Client Developer, SQL Developer, RESTful Services, Schema Administrator, SODA Developer, and SODA Developer. The role 'TTI API Client' is selected and highlighted on the right.

Below the main definition area, there is a section for 'Protected Modules'. It shows a list of modules on the left and a list of selected modules on the right:

- Modules:** Test
- Selected Modules:** TTI Accounts Payable v1, TTI Accounts Receivable v1, TTI General Ledger v1, TTI Order Entry v1, TTI Shipping v1

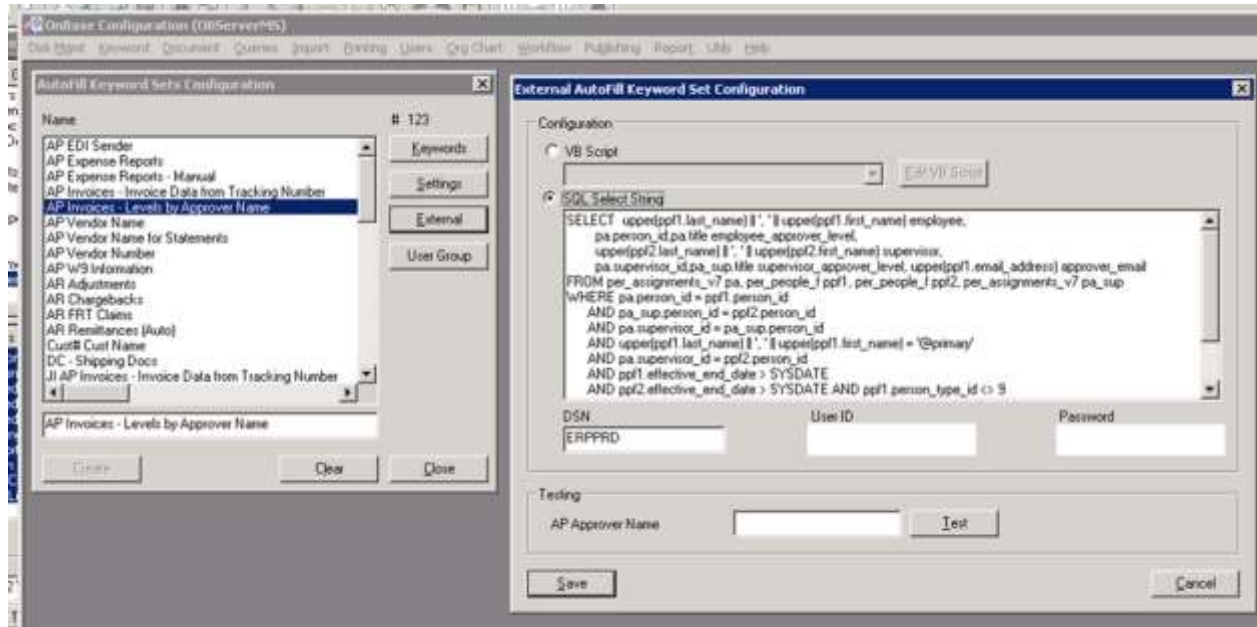


# Oracle eBusiness Suite

- 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

```
1 -- AP Invoices - Levels by Approver Name
2 SELECT upper(ppf1.last_name) || ', ' || upper(ppf1.first_name) employee,
3        pa.person_id,pa.title employee_approver_level,
4        upper(ppf2.last_name) || ', ' || upper(ppf2.first_name) supervisor,
5        pa.supervisor_id,pa_sup.title supervisor_approver_level, upper(ppf1.email_address) approver_email
6 FROM per_assignments_v7 pa, per_people_f ppf1, per_people_f ppf2, per_assignments_v7 pa_sup
7 WHERE pa.person_id = ppf1.person_id
8        AND pa_sup.person_id = ppf2.person_id
9        AND pa.supervisor_id = pa_sup.person_id
10        AND upper(ppf1.last_name) || ', ' || upper(ppf1.first_name) = :p_full_name
11        AND pa.supervisor_id = ppf2.person_id
12        AND ppf1.effective_end_date > SYSDATE
13        AND ppf2.effective_end_date > SYSDATE AND ppf1.person_type_id <> 9
14        AND nvl(pa.title,'X') <> 'X';
```



We need the JSON!

<b>Name:</b>	<b>AP Approver</b>											
<b>Description:</b>	Get details for an employee regarding AP approval limits and supervisor's approval limits.											
<b>URL:</b>	/api/v1/ap/approver											
<b>HTTP Method:</b>	GET											
<b>HTTP Header Parameters:</b>	<table border="1"><thead><tr><th>Parameter Name</th><th>Type</th><th>Comments</th></tr></thead><tbody><tr><td><u>employeeName</u></td><td>String</td><td>Required unless <u>employeeNameStartsWith</u> is provided. Format: "Last Name, First Name".</td></tr><tr><td><u>employeeNameStartsWith</u></td><td>String</td><td>Required unless <u>employeeName</u> is provided. Format: "Last Name, First Name". Performs a LIKE-type of search on employees.</td></tr></tbody></table>	Parameter Name	Type	Comments	<u>employeeName</u>	String	Required unless <u>employeeNameStartsWith</u> is provided. Format: "Last Name, First Name".	<u>employeeNameStartsWith</u>	String	Required unless <u>employeeName</u> is provided. Format: "Last Name, First Name". Performs a LIKE-type of search on employees.		
Parameter Name	Type	Comments										
<u>employeeName</u>	String	Required unless <u>employeeNameStartsWith</u> is provided. Format: "Last Name, First Name".										
<u>employeeNameStartsWith</u>	String	Required unless <u>employeeName</u> is provided. Format: "Last Name, First Name". Performs a LIKE-type of search on employees.										
<b>Example Request:</b>												
<b>Example Response:</b>	<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.ttfc.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

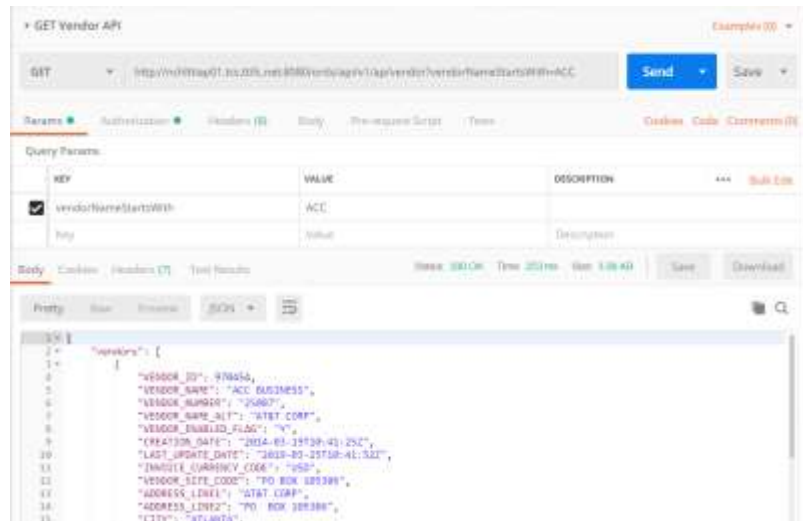
```
Invoke-WebRequest
1 URL
2 https://111.222.333.44/api/v1/web/pickalip7deliveryId=20412442
3 https://111.222.333.44/api/v1/web/pickalip7deliveryId=20412442
4 https://FRODOBAGGINS.xxx.tti.fc.net/api/v1/web/pickalip7deliveryId=20412442
5 https://FRODOBAGGINS.xxx.tti.fc.net/api/v1/web/pickalip7deliveryId=20412442
6 https://api.tti.frodobags.com/api/v1/web/pickalip7deliveryId=20412442
7 https://api.tti.frodobags.com/api/v1/web/pickalip7deliveryId=20412442
8 https://FRODOBAGGINS.xxx.tti.fc.net/api/v1/cpl/account?fullAccount=20,0000,223503,10,00000,00000
9 https://FRODOBAGGINS.xxx.tti.fc.net/api/v1/cpl/account?fullAccount=20,0000,223503,10,00000,00000
10 https://FRODOBAGGINS.xxx.tti.fc.net:8080/origin/api/v1/web/pickalip7deliveryId=20412442
11
```

```
Invoke-WebRequest
1 Write-Output "-----"
2 Write-Output " -- prod environment web service testing -----"
3 Write-Output "-----"
4 $start=Get-Date
5 Write-Output $start
6 Write-Output "-----"
7
8 $stopwatch = [system.diagnostics.stopwatch]::StartNew()
9
10 $user = "devendash"
11 $pass= "XXXXXXXXXXXXXXXX"
12 $pair = "$($user)|$($pass)"
13
14 $encodedCreds = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($pair))
15
16 $basicAuthValue = "Basic $encodedCreds"
17
18 $headers = @{
19     Authorization = $basicAuthValue
20 }
21
22 $csv = Import-Csv c:\obdebag\endpoints-pro.csv
23 foreach ($line in $csv) {
24     Write-Output "-----"
25     $root = $line.URL
26     Write-Output $root
27     Invoke-WebRequest -Uri $root -Headers $headers | Select StatusCode, Content
28 }
29
30 Write-Output "Total Seconds:"
31 $stopwatch.Elapsed.TotalSeconds
32 $stopwatch.Stop()
33
34 Write-Output "-----"
35 Write-Output "-----"
36 $end=Get-Date
37 Write-Output $end
38 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



# 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?

