

Oracle ORDS Technical Guide

Aptean Ltd
Copyright © 2011-2026.

Contents

- 1 Oracle ORDS Technical Guide.....1
 - 1.1 Pre-requirements.....1
 - 1.2 Overview of Affected Tables.....1
 - 1.3 Setting Up.....1
 - 1.4 Examples.....2
 - 1.5 Basic Queries.....4

1 Oracle ORDS Technical Guide

This is a technical guide for the setup of Oracle ORDS, used for OAUTH2 webservice through Oracle Fusion Middleware.

 **Note:** The majority of the ORDS schema creation is through a WAR file that is installed when the process is initialized.

The initial build consists of a java unzip, and then using the WAR file to create schemas / deploy objects to the database (logged in as sys) - this WAR file is also the ?ORDS java? side and a minor amount of config.

This guide is intended to be a small introduction to the technical tables and requirements, for support purposes.

1.1 Pre-requirements

- PL/SQL Developer.
- TNS Name for the Oracle database.

1.1.1 Users

- ords
- ordsimport
- mts_owner

1.2 Overview of Affected Tables

1.2.1 Settings for users when editing ORDS

```
select * from user_ords_roles;
select * from user_ords_privileges;
select * from user_ords_privilege_roles;
select * from user_ords_privilege_mappings;
```

1.2.2 Creation

The actual web services

```
select * from user_ords_modules;
select * from user_ords_services;
select * from user_ords_handlers;
select * from user_ords_templates;
select * from user_ords_schemas;

select * from user_ords_parameters;
```

1.2.3 Execution

```
select * from v$session where type = 'USER'
select * from user_ords_clients;
select * from user_ords_client_privileges;
select * from user_ords_client_roles;
```

1.3 Setting Up

As mts_owner:

```
create user ords identified by {password} default tablespace data temporary tablespace temp profile default;
grant ords_administrator_role to ords;
grant ords_runtime_role to ords;
grant create session to ords;
grant select any table to ords;

create user ordsimport identified by {password} default tablespace data temporary tablespace temp;

grant create session to ordsimport;
grant ords_runtime_role to ordsimport;
grant select any table to ordsimport;
```



```
grant execute on mts_owner.dp_ctms_import to ordsimport;
```

 **Note:** The grant of execute on the package must be the package that the ORDS handler is going to execute within the MTS_OWNER space. If this is different (or there are several), then this must be changed and added to here.

as ords:

```
begin
  ords_metadata.ords.enable_schema(
    p_schema      => 'ordsimport',
    p_url_mapping_type => 'BASE_PATH',
    p_url_mapping_pattern => 'import'
  );

  commit;
end;
/

declare
  l_roles_arr owa.vc_arr;
  l_patterns_arr owa.vc_arr;
begin
  l_roles_arr(1) := 'ordsimport_role';
  l_patterns_arr(1) := 'ordsimport_pattern/';

  ords_metadata.ords.define_privilege (
    p_privilege_name => 'ordsimport_priv',
    p_roles          => l_roles_arr,
    p_patterns       => l_patterns_arr,
    p_label          => 'test priviledge',
    p_description    => null
  );

  commit;
end;
/

select * from user_ords_privileges;
select * from user_ords_privilege_roles;
select * from user_ords_privilege_mappings

begin
  ords_metadata.oauth.create_client(
    p_name      => 'ordsimport_client',
    p_grant_type => 'client_credentials',
    p_support_email => 'noreply@email.com',
    p_privilege_names => 'ordsimport_priv'
  );

  commit;
end;
/

select * from user_ords_clients;
select * from user_ords_client_privileges;

begin
  ords_metadata.oauth.grant_client_role(
    p_client_name => 'ordsimport_client',
    p_role_name   => 'ordsimport_role'
  );
  commit;
end;
/

select * from user_ords_client_roles;
```

1.4 Examples

Example Order Create - example uses XML - REST, XML-based order create method/service.

```
BEGIN
  ORDS.define_module(
    p_module_name => 'rest-xml-import-order-create',
```



```

    p_base_path      => 'order',
    p_items_per_page => 0);

ORDS.define_template(
  p_module_name => 'rest-xml-import-order-create',
  p_pattern     => 'create/');

ORDS.define_handler(
  p_module_name => 'rest-xml-import-order-create',
  p_pattern     => 'create/',
  p_method      => 'POST',
  p_source_type => ORDS.source_type_plsql,
  p_source      =>
q'[
  declare
    l_response xmltype;

    CHUNK_SIZE constant pls_integer := 8192;
    vOffset    pls_integer := 1;
    DOC1       clob;
    vChunk      varchar2(CHUNK_SIZE CHAR);

  begin

    l_response := mts_owner.dp_ctms_import.import_order(ctms_ord => XMLTYPE.createXML(:body_text));
    owa_util.mime_header ('application/xml', true);

    DOC1 := XMLTYPE.getClobVal(l_response);
    loop
      vChunk := substr (DOC1, vOffset, CHUNK_SIZE);
      exit when vChunk is null;
      http.prn(vChunk);
      vOffset := vOffset + length(vChunk);
    end loop;

    end;
  ]',
  p_items_per_page => 0);

  COMMIT;
END;
/

```

Example handler for LogiNext interface - REST, JSON-based order update method/service.

```

BEGIN
  ORDS.define_module(
    p_module_name => 'rest-json-import-order-update',
    p_base_path   => 'order',
    p_items_per_page => 0);

  ORDS.define_template(
    p_module_name => 'rest-json-import-order-update',
    p_pattern     => 'update/');

  ORDS.define_handler(
    p_module_name => 'rest-json-import-order-update',
    p_pattern     => 'update/',
    p_method      => 'POST',
    p_source_type => ORDS.source_type_plsql,
    p_source      =>
q'[
  declare
    l_response json_object_t := json_object_t();

    CHUNK_SIZE constant pls_integer := 8192;
    vOffset    pls_integer := 1;
    vData       clob := EMPTY_CLOB();
    vChunk      varchar2(CHUNK_SIZE CHAR);

  begin

    l_response := mts_owner.dp_loginext.process_inbound(i_message => json_object_t.parse(:body));
    owa_util.mime_header ('application/json', true);

    vData := l_response.to_clob;

```



```

    loop
        vChunk := substr (vData, vOffset, CHUNK_SIZE);
        exit when vChunk is null;
        http.prn(vChunk);
        vOffset := vOffset + length(vChunk);
    end loop;

exception when others then

l_response := JSON_OBJECT_T.parse('{"status":"exception","sqlcode":"' || sqlcode || '", "sqlerrm":"' || sqlerrm || '"}');
http.p(l_response.stringify);

end;
j',
p_items_per_page => 0);

COMMIT;
END;
/

```

1.5 Basic Queries

```

select * from user_ords_handlers
where UPPER(source) like '%DP_LOGINEXT%'

```

