



Advanced API Development with Oracle REST Data Services (ORDS)

Thursday, June 5th 2025 (20:00 - 20:50)



Robert Marz
DATABEE
Die IT-Architekten



Robert Marz – Independent Consultant

Primary Role

Senior Technical Architect
with database centric view of the world

ora2know

The German Oracle Database Community.
Database first. Community first. ora2know.de
Member of the Board



DATABEE
Die IT-Architekten



Databees.



ora2know
The German Oracle Database Community

SYM^{L2}



@robbie.databee.org



<https://robbie.databee.org>



robert.marz@databee.org



**Oracle ACE
Pro**



Eine starke Community für Oracle Datenbank Benutzer

Ziele: Wissensvermittlung für Entwickler und DBAs

Angebote: MeetUps, Podcasts, Live Events

Projekte: Veranstaltung zum Kennenlernen

Kontakt: info@ora2know.de



ora2know

The German Oracle Database Community



The Oracle ACE Program

400+ technical experts helping peers globally



- The Oracle ACE Program recognizes and rewards community members for their technical and community contributions to the Oracle community
- 3 membership levels: Director, Pro, and Associate
- Nominate yourself or a colleague at ace.oracle.com/nominate
- Learn more at ace.oracle.com

✉ aceprogram_ww@oracle.com

Facebook.com/OracleACEs

@oracleace

Oracle ACE Program Group



SYMPOSIUM⁴²

Created by the community, to support the community

Sharing of reliable knowledge

Supporting the various user groups and individuals



@sym_42



<https://sym42.org/>

Designing your REST API

API



API Design Approaches

**Code first,
think never?**

- **Design carefully**
- **Switch roles - You are not the user**

“Programs must be written for people to read,
and only incidentally for machines to execute.”

Harold Abelson, Structure and Interpretation of Computer Programs, 1984

This applies to APIs, even more.





API Grammar

Nouns / What?

Your API Objects

e.g. contracts, cars, VirtualMachines, ...

GET /products

GET /VirtualMachines/4711

Verbs / How?

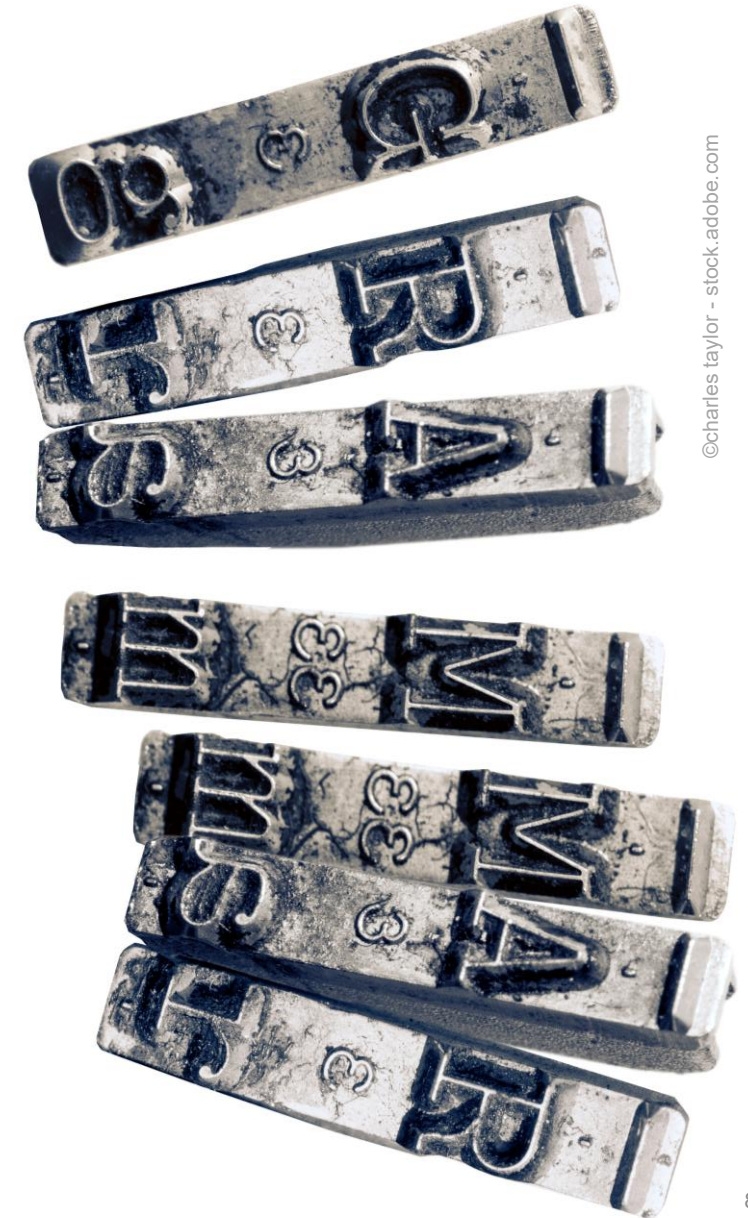
http Methods

e.g GET, POST, PUT, DELETE

Relations

Sub-resources

e.g DELETE /VirtualMachines/4711/VMDiskMappings/5



©charles taylor - stock.adobe.com



API Design Best Practices

- Try it, Test it, Document it
- Be redundant
- Use nouns, but no verbs, Nouns are plural
- GET method should never alter states
- Use HTTP headers
- Provide Filtering, Sorting, Field Selection & Paging
- Use ETAGs and HATEOAS

Document your REST API





Swagger vs. open API

Swagger

Cloud Platform

Swagger Editor

Swagger UI

Swagger Codegen

Swagger Hub

Swagger Inspector

API description standard

donated to Linux Foundation

renamed to OpenAPI Specification

JSON or YAML files



Swagger™

Supported by SMARTBEAR

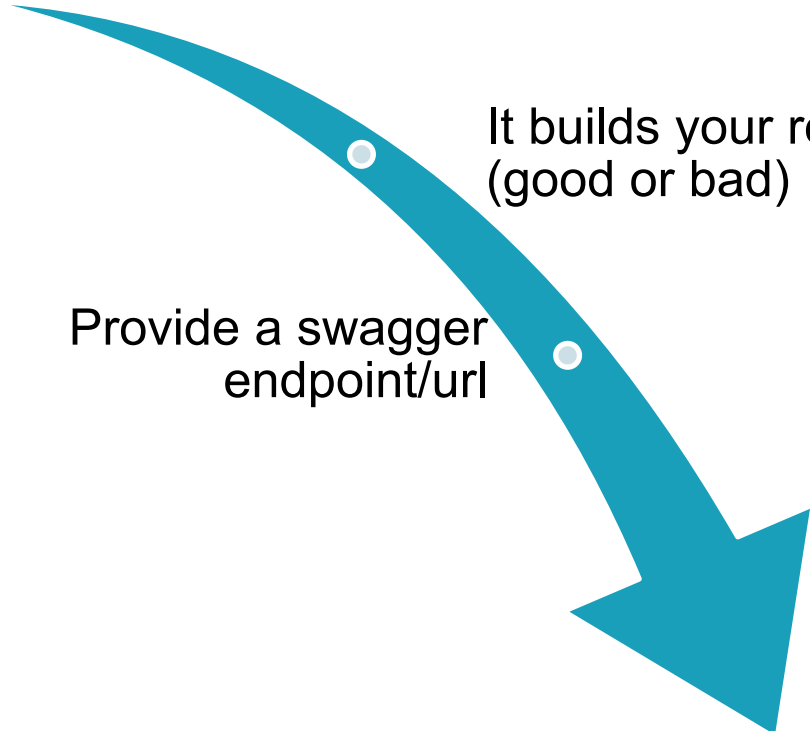


OPENAPI
INITIATIVE



Open API Documentation

This is the only Documentation, your users will see



It builds your reputation (good or bad)

Provide a swagger endpoint/url

Keep your API consistent with the Docs

The screenshot shows the Swagger UI for the Petstore API. At the top, there's a header with the Swagger logo, the URL `https://petstore.swagger.io/v2/swagger.json`, and an **Explore** button. Below the header, the title **Swagger Petstore** is displayed with version tags **1.0.7** and **OAS 2.0**. The base URL is `petstore.swagger.io/v2` and the Swagger JSON URL is `https://petstore.swagger.io/v2/swagger.json`. A paragraph of text explains that this is a sample server and provides links to Swagger documentation and a special API key `special-key`. There are links for [Terms of service](#), [Contact the developer](#), [Apache 2.0](#), and [Find out more about Swagger](#). Below this, there's a **Schemes** dropdown menu set to **HTTPS** and an **Authorize** button with a lock icon. The main content area shows a section for **pet** (Everything about your Pets) with a **Find out more** link. Two API endpoints are listed: **POST** `/pet/{petId}/uploadImage` (uploads an image) and **POST** `/pet` (Add a new pet to the store). Both endpoints have a lock icon and a dropdown arrow.



Convert

openapi.yml distributed over
multiple files
into a single json file

```
java -jar swagger-codegen-cli-3.0.64.jar generate -l openapi -i ${api} -o ./out
```

<https://swagger.io/docs/open-source-tools/swagger-ui/usage/installation/>



SwaggerUI

URL to your
openapi.json

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="description" content="SwaggerUI" />
    <title>SwaggerUI</title>
    <link rel="stylesheet" href="https://unpkg.com/swagger-ui-dist@5.11.0/swagger-ui.css" />
  </head>
  <body>
    <div id="swagger-ui"></div>
    <script src="https://unpkg.com/swagger-ui-dist@5.11.0/swagger-ui-bundle.js" crossorigin></script>
    <script src="https://unpkg.com/swagger-ui-dist@5.11.0/swagger-ui-standalone-preset.js" crossorigin></script>
    <script>
      window.onload = () => {
        window.ui = SwaggerUIBundle({
          url: 'https://petstore3.swagger.io/api/v3/openapi.json',
          dom_id: '#swagger-ui',
          presets: [
            SwaggerUIBundle.presets.apis,
            SwaggerUIStandalonePreset
          ],
          layout: "StandaloneLayout",
        });
      };
    </script>
  </body>
</html>
```

<https://swagger.io/docs/open-source-tools/swagger-ui/usage/installation/>



Open API – What to document

Endpoints (Modules)

- Description

Parameters

- URL / Header
- Datatypes
- Examples

Body & Response

- JSON Schemas
- Examples



©Shaiith Nowak Jacek - stock.adobe.com

Organize your Code





Custom PL/SQL oder ORDS Functionality

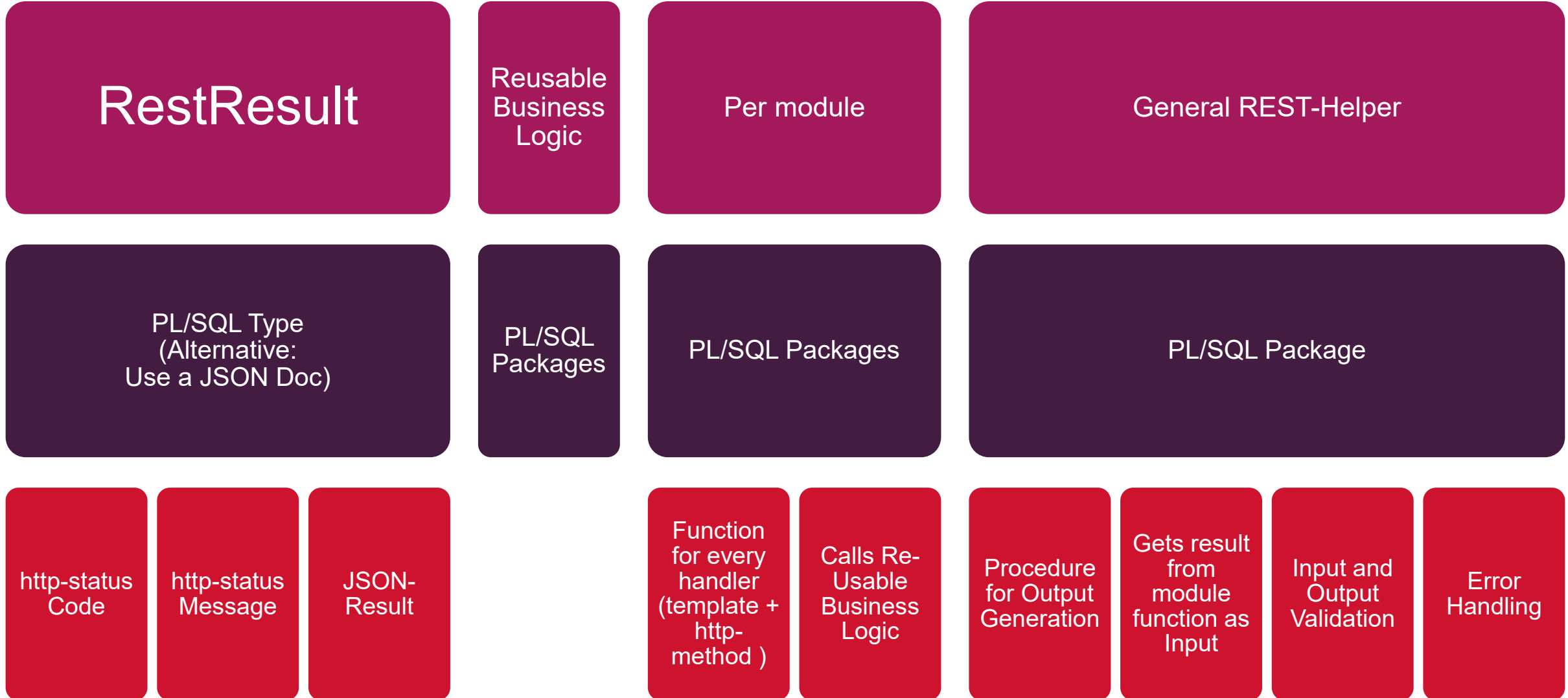
use ords source type
whenever you can



use custom pl/sql
whenever you must

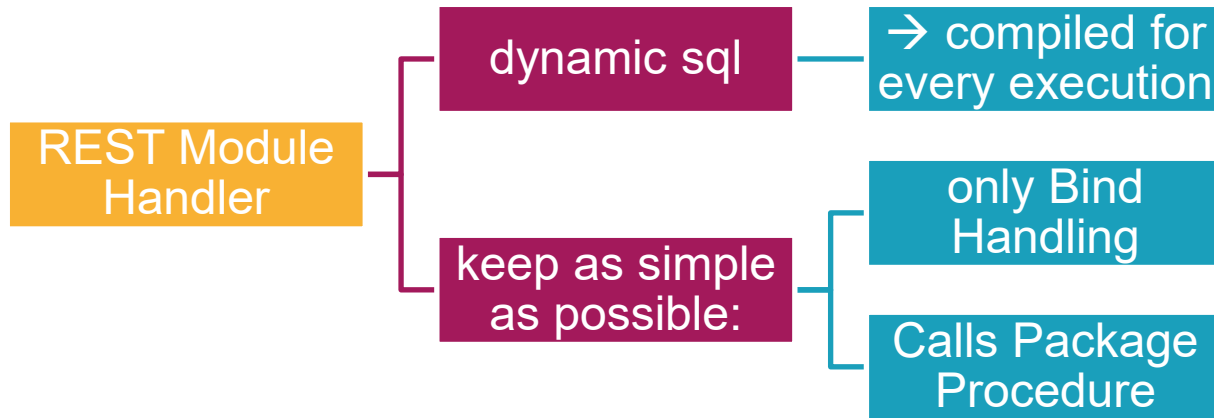


Organize PL/SQL Code





REST Module Handler Definition



REST > Module > ffacmd_v1 > execwf > POST

execwf

Zuletzt aktualisiert: vor 9 Minute

Keine Kommentare verfügbar

Quellentyp: plsql/block Seitengröße: 0

http://localhost:8080/ords/fsm_core/ffacmd/v1/execwf

Quelle

```
1 begin
2   :status_code := common.process_rest_result(rest_execwf.post(:body));
3 end;
```



Paging

Paging is crucial for large datasets

REST call are stateless and independent

Provide metadata in the responses:

- are there more rows
- current offset
- links to following and preceding pages

The ORDS source types provides those for free

Hypermedia as the Engine of Application State (HATEOAS)



ETags for Caching and DML Locking

ETags (Entity Tags)

- http headers
- identify specific versions of resources
- part of http/1.1 spec (RFC 7232)
- great for result caching

Optimistic Concurrency Control

- Prevention of Mid-air Collisions
- ensure client has read the latest changes before updating

ORDS source types

- bring ETags for free





JSON Schemas

Validate all JSON against schemas

- Incoming
- Outgoing
- Resting in the Database

Make sure schemas

are consistent with docs
→ re-use your openAPI spec



ORDS Auth with JWTs



JWT (JSON Web Token)

JWT (JSON Web Token)

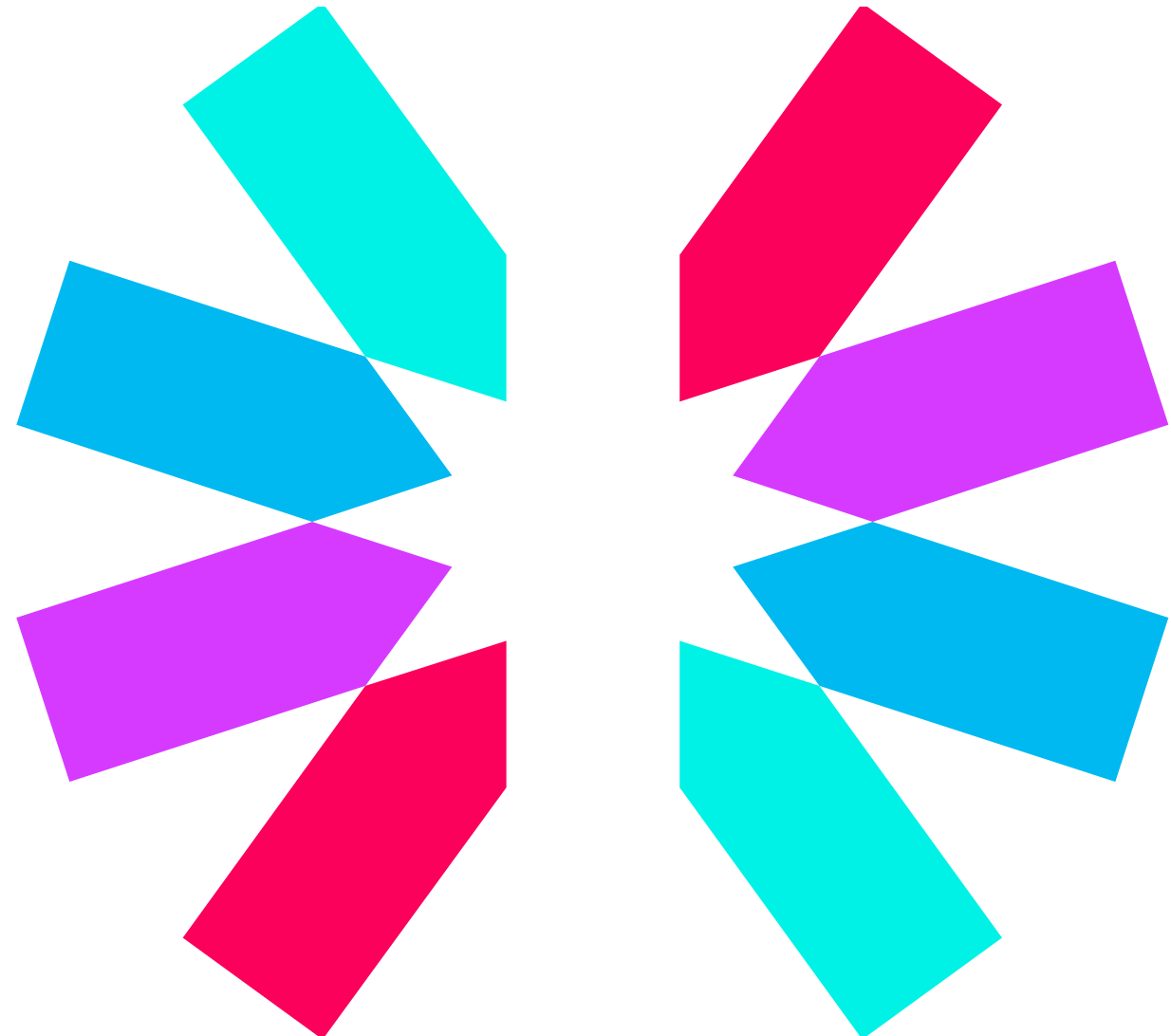
- industry standard RFC 7519
- represent claims securely between two parties

Transport

- http header field
- Authorization: Bearer {{token}}

Parts

- 3 Sections:
 - Header (JSON)
 - Payload (JSON)
 - Signature
- Base64 encoded





Java Web Token – How it works

```
{
  "exp": 1732104719,
  "iat": 1732102919,
  "jti": "874be9ae-b186-4684-af50-f1cfa726e581",
  "iss": "https://host.example.com/realms/tenant1",
  "sub": "5db8e25d-21c7-4cd4-ac1e-a26832b46850",
  "typ": "Bearer",
  "azp": "Dispo",
  "session_state": "8af6a246-7277-4216-b660-4c44b95a5c20",
  "acr": "1",
  "allowed-origins": [
    "https://host.example.com/Acceptance/dispoManager/",
    "http://localhost:5000",
    "https://host.example.com/dispoManager/"
  ],
  "scope": "openid email profile",
  "sid": "8af6a246-7277-4216-b660-4c44b95a5c20",
  "name": "Robert Marz",
  "groups": ["/Dispatcher", "/Technican"],
  "preferred_username": "postman-test-user-03",
  "given_name": "Robert",
  "family_name": "Marz",
  "email": "robert.marz@databee.org"
}
```

Client

establishes Session
with ID Provider

Receives Bearer
Token with Claims

Sends Token as http-
header to Server

Server

Decodes Token

Validate Token (online
or offline)

Extracts Information

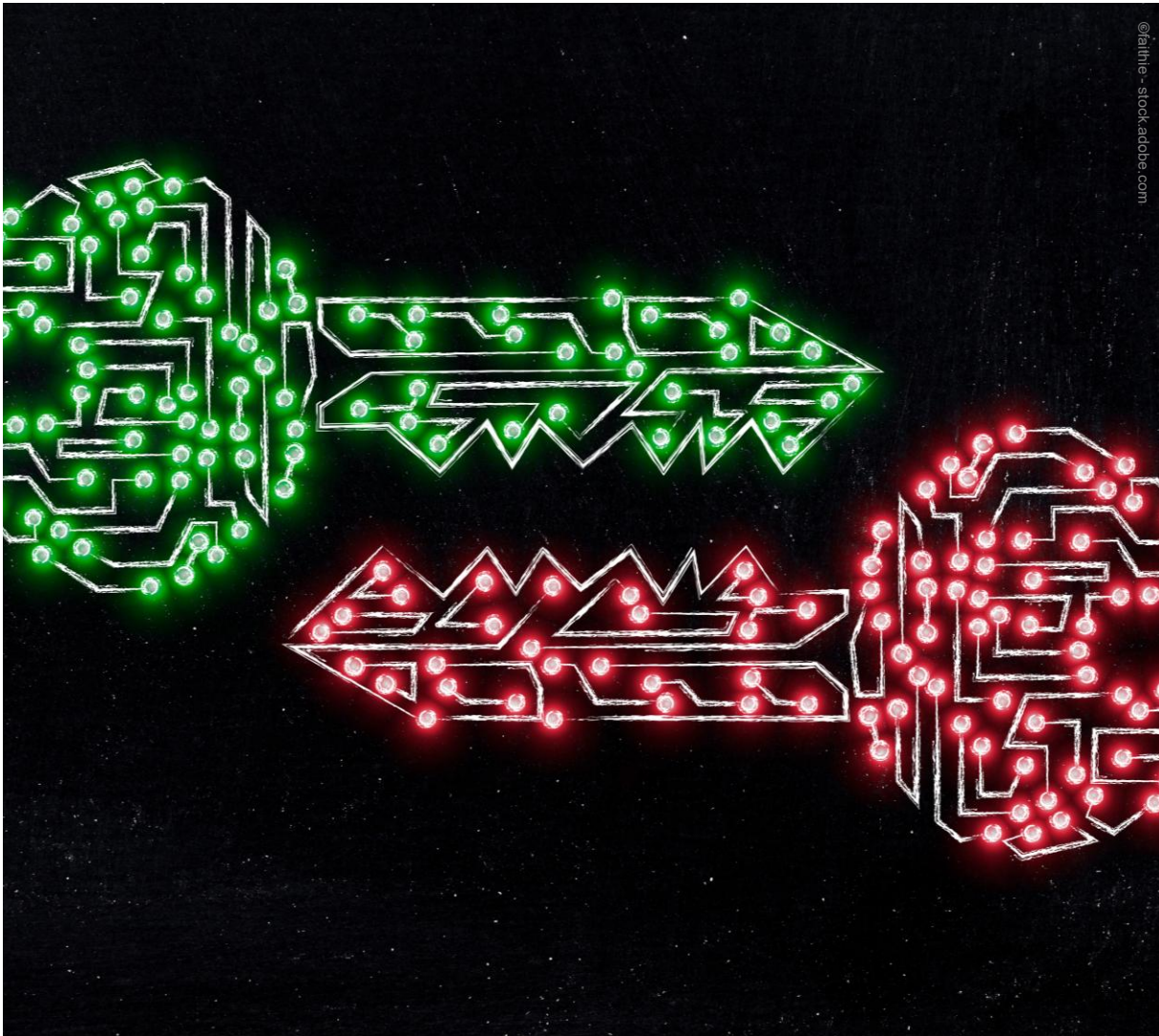
Username

Full-Name

Groups



ORDS Auth with JWTs



ORDS

- latest release has JWT profiles
- enabled by default
- if not configured, every JWT will result in an error and Java Stacktrace

ORDS JWT profile disable

```
ords config set  
security.jwt.profile.enabled false
```

Parse JWT

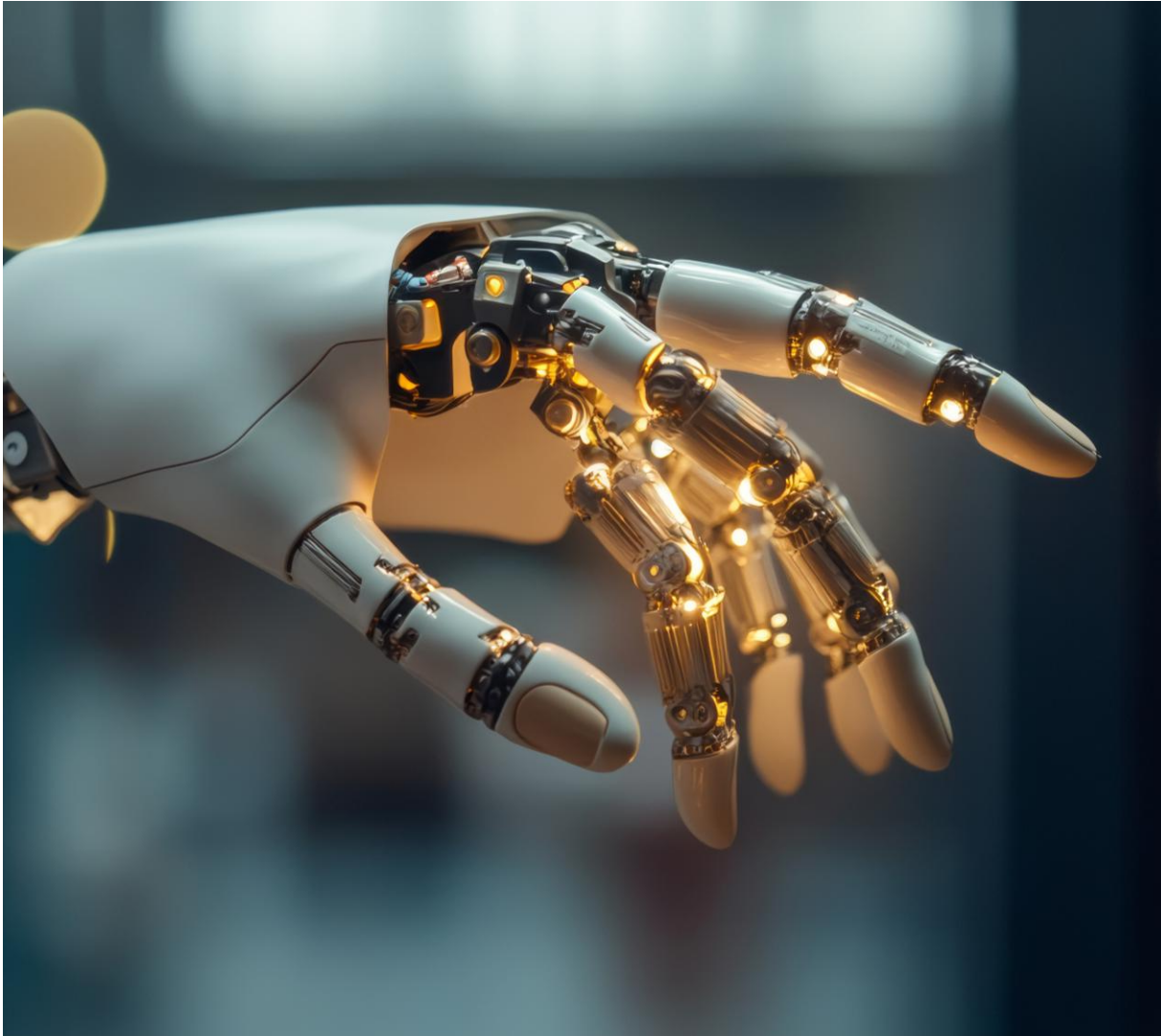
- You have to parse the payload for claims like roles, groups, etc anyway

A photograph of a dormitory room. The room features rows of wooden bunk beds on both sides, with white bedding and pillows. A central aisle leads to a window with dark curtains. The ceiling has a white fan and recessed lights. The walls are wood-paneled.

Sharing Auth between APEX and REST



JWTs and Apex



Apex

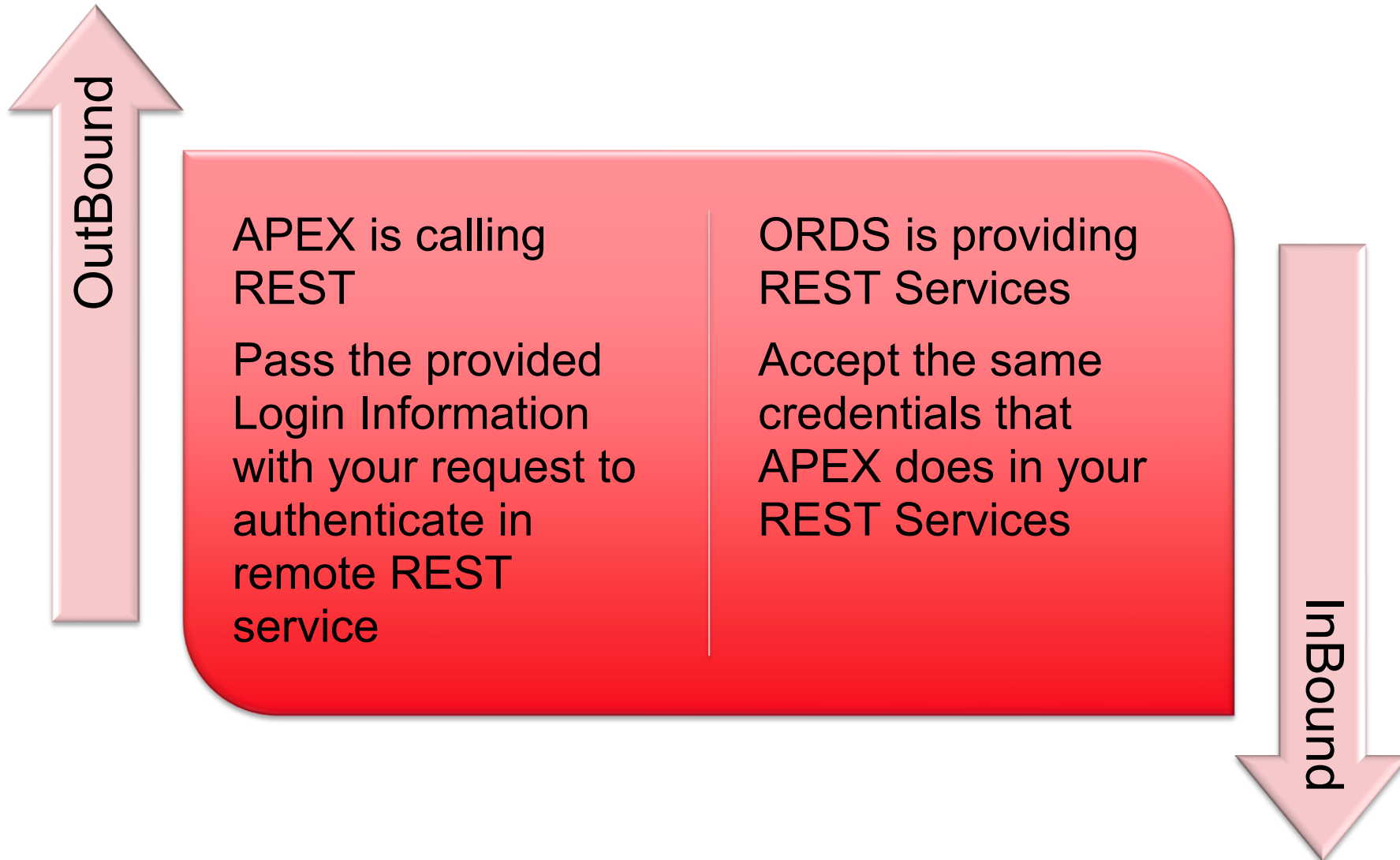
- Built-In Support

Social-Sign-In Auth Scheme

- social networks and enterprise identity providers that support OpenID Connect or OAuth2 standards
 - - MS Entra ID (ex Azure Active Directroy)
 - - KeyCloak
 - ...



REST & APEX Scenarios





JWTs and REST outbound

```
/* We want to mimic this curl command in PL/SQL.
curl --request POST 'https://host.example.com/cool/service' \
--header 'Authorization: Bearer eyJhbGciOiJI ... '
*/

DECLARE
    l_response      CLOB;
    jwt_token       VARCHAR2(32000) := 'eyJhbGciOiJI ... ';
BEGIN
    -- Set the Request HTTP Headers.
    apex_web_service.set_request_headers
    (p_name_01  => 'Authorization',
     p_value_01 => 'Bearer ' || jwt_token,
     p_reset   => TRUE);

    -- Call API using Username and Password.
    l_response := apex_web_service.make_rest_request
    (p_url           => 'https://dummyjson.com/auth/login',
     p_http_method  => 'POST');
END;
```



JWTs and REST inbound



Parse incoming JWT

Package APEX_JWT

- encode
- decode
- validate – no session verification!



Check if Session is valid

online: REST call to ID Provider

offline: check signature against public key



Make user details available

Set package Variables

set session context

ORDS Pre-Hook





ORDS Pre Hook



ORDS Pre Hook

PL/SQL function called before every ORDS Request
(that includes APEX calls)



returns boolean

- true: all good
- false: HTTP status 403 forbidden
- unavailable: HTTP status 500 internal error



Configure

per connecton pool
ords setting `procedure.rest.preHook`



→ Perfect place for JWT validation



ORDS PreHook best Practices

small pre_hook function

place in schema on its own

standalone function (no package)

call pre_hooks function in REST-enabled-Schema

return true if not needed



pre_hooks (in packages)

for every REST enabled schema

or even for every module

do the actual job



Be careful

defect pre_hook stops

REST Services and Apex for your ords-pool



```
create or replace function ords_pre_hook
  return boolean
  authid current_user
as
  l_apex_base varchar2(32767 char) := OWA_UTIL.get_cgi_env('X-APEX-BASE');
  l_path varchar2(32767 char) := OWA_UTIL.get_cgi_env('X-APEX-PATH');
  l_result boolean := true;
begin
  if l_apex_base like '%/ords/fsm_%'
  then
    -- ignore swagger services
    if l_path like 'swagger%' then return true; end if;
    -- ignore health services
    if l_path like 'health%' then return true; end if;
    -- ignore sdw (sql dev web) services
    if l_path like 'sign-in%' then return true; end if;
    -- ignore services that start with an underscore
    if regexp_like (l_path, '^_|^/_') then return true; end if;
    -- ignore metadata-catalog
    if l_path like 'metadata-catalog%' then return true; end if;
    execute immediate 'begin :1 := ' || user || '.common.ords_pre_hook; END;' using out l_result;
  end if;
  return l_result;
end ords_pre_hook;
/

grant execute on fsm_core.ords_pre_hook to ORDS_PUBLIC_USER
/
```



Timing and Logging



Timing and Logging

Integrate it in your Framework

- prehook
- output processor

Telemetry is important

Think about retention times

- establish a cleanup process

log at least per handler

entry and exit

parameter and body

response





API Management Systems

- Azure API Management
- Amazon API Gateway
- Google Cloud API Gateway
- Apigee (now part of Google Cloud)
- MuleSoft Anypoint Platform
- IBM API Connect
- Kong Enterprise
- Tyk Cloud
- Postman API Platform
- RapidAPI

Cloud-based



- Kong Gateway (open source version)
- Tyk Open Source API Gateway
- Apache APISIX
- WSO2 API Manager
- Gravitee.io
- KrakenD
- Gloo Edge
- 3scale API Management (Red Hat)

On-premises
and Open
Source



- Axway API Management
- Software AG webMethods API Management
- CA API Management (Broadcom)
- Dell Boomi API Management
- Oracle API Platform Cloud Service

Hybrid and
Multi-Cloud

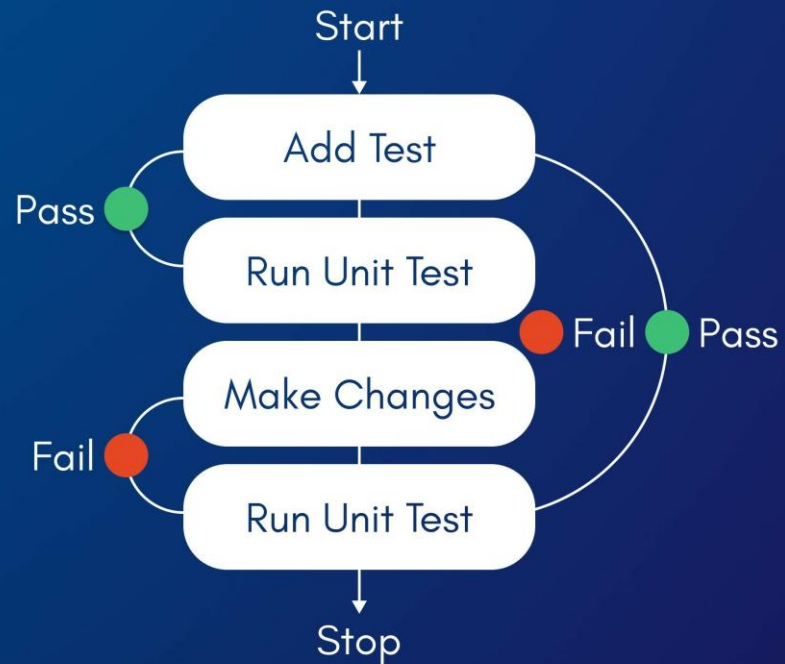




API Testing



UNIT TESTING *In Software Engineering*



Test your

- Business Functions
- Modules
- Endpoints

Use Tools

- [utPLSQL](#)



API Tests with REST Client

Rest Clients

- Postman
- Insomnia
- ...

Build Collections

- contain every endpoint
- use case
- variable

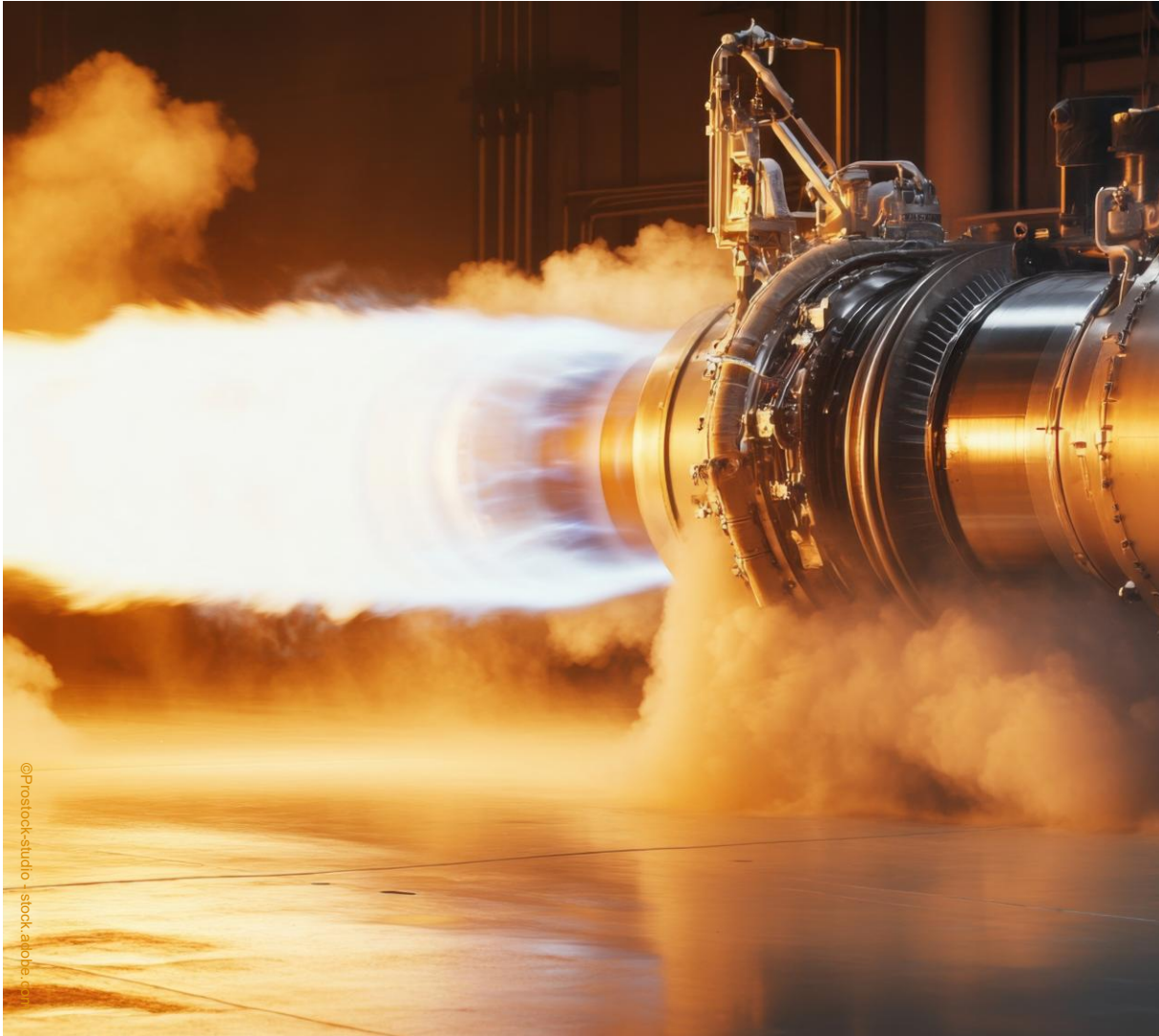
Run through workflow

- generate and collect values
- Script (JS)





Stress and Load Tests



@Pirostock-studio + stock.adobe.com

Performance

- normal usage
- peak usage
- DoS Attacks
- Mistakes on Client side

Open Source Tools

- Apache Jmeter
- k6 (now part of Grafana Labs)
- Gatling
- Locust
- ...

Conclusion





Advanced API Development with ORDS

Design your API carefully

Document your API

Keep code in ORDS Module Handler short

Authenticate with JWTs / oAuth

ORDS pre Hook is powerful

Test your API

PLEASE

**DO
TRY THIS
AT HOME**