# Progress Application Server

- **PAS** is a Tomcat Java web application server with Progress customizations and extensions
- Tomcat is a component of the J2EE Platform and provides:
  - Catalina to run Servlets
  - Coyote connector to provide an http-Server
  - Jasper to interpret Java Server Pages
  - Spring Security
- Provides optional web applications for remote administration using text, html, or JMX proxy APIs

PAS for OpenEdge

- **PASOE** is a PAS with a pre-installed extension to run ABL Code in ABL Agents in the context of Tomcat Web Applications.
- The ABL Agents handle multiple sessions contexts using multiple threads to provide a multi-Sessioned Agent (MS-Agent)
- You design, package, deploy, configure, debug, and control access to your ABL application in the context of a Tomcat web application
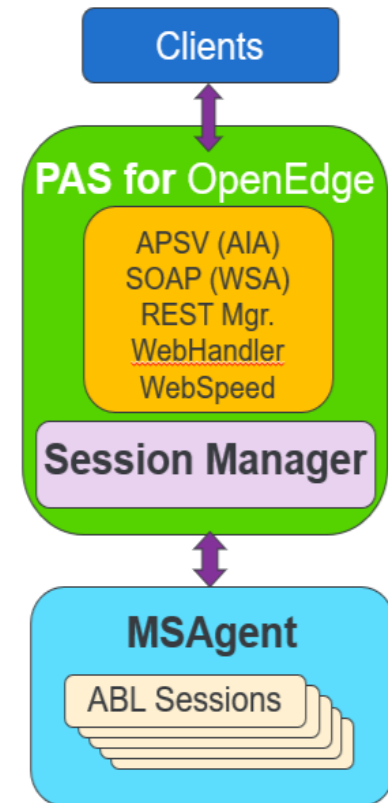
Why creating a next generation AppServer?

Simplicity

- Administration, scalability, application migration, deployment
- AppServer connection and operating STATEs

Security

- Spring Security
- Production license

Better monitoring and management

- Built-in metrics gathering, current state queries

Performance

- Runs same ABL application and client load with less memory and CPU

WebSpeed

- Runs classic code but also OOABL WebHandlers

Why a Web Application Server?

👆 Todays Application Architecture needs to be ready

- to work with various UI Technologies, including GUI, Web Browser and Mobile
- Provide API-based access to business logic for „Client-Applications" including Custom Extensions, Micro Apps and B2B-Integration
- Need for a standards-based, open and secure Interface in contrast to the proprietary binary OE GUI –AppServer protocol.

👆 Software as a Service is the main growth path

- Developing „Web- and Cloud-native" applications
- Secure and Scalable
- Easy deployment and management in virtualized environments (VMs, Containers)

Classic Appserver versus PASOE

- Runs ABL code
- Supports GUI Clients directly
- Supports other Clients by Adapters
  - GUI Clients via http/s tunnel
  - REST Adapter
  - SOAP/WSA Adapter
  - WebSpeed w/ Messenger
- Adapters based on a number of technologies
- Needs external Servers

- Runs ABL code
- Supports http-Clients
  - GUI Clients via http/s tunnel (APSV)
  - REST, SOAP
  - WEB (ABL)
    - WebSpeed Handler
    - REST Handler
    - PDS generated
    - PDS mapped
- Less Resources, Higher Performance

Comparison Classic AppServer vs. PASOE

Feature	Classic AppServer	PASOE
Is a Web Server	No	Yes
Security	No built-in	Built in Spring-Security
Built in Protocol Support	GUI	ASPV, REST, SOAP, Web Handler
External Protocol Adapters	AIA, WSA, REST Manager, WebSpeed Messenger	All built-in
Monitoring	OpenEdge Explorer/Mgt.	JMX, OEManager with OEE/OEM, HealthScanner
Requires AdminServer	Yes	No
Scalability	By number of Agents	By number of Sessions and Agents
Load Balancing	By Name Server Load Balancer	Standard Apache http Server AJP
Production Best-Practices	AppServer and DB on one Server, Shared Memory	Deploy PASOE on-prem or in the Cloud, physical, VM or Container
Works well with	OE 11 GUI Client	OE 12 RDBMS

Architecture: Components

IAP

Classic AppServer Components

- Client
- AdminServer
- AIA WSA WebSpeed REST/Mobile
- NameServer
- AppServer Broker
- AppServer Agent
  - ABL Session
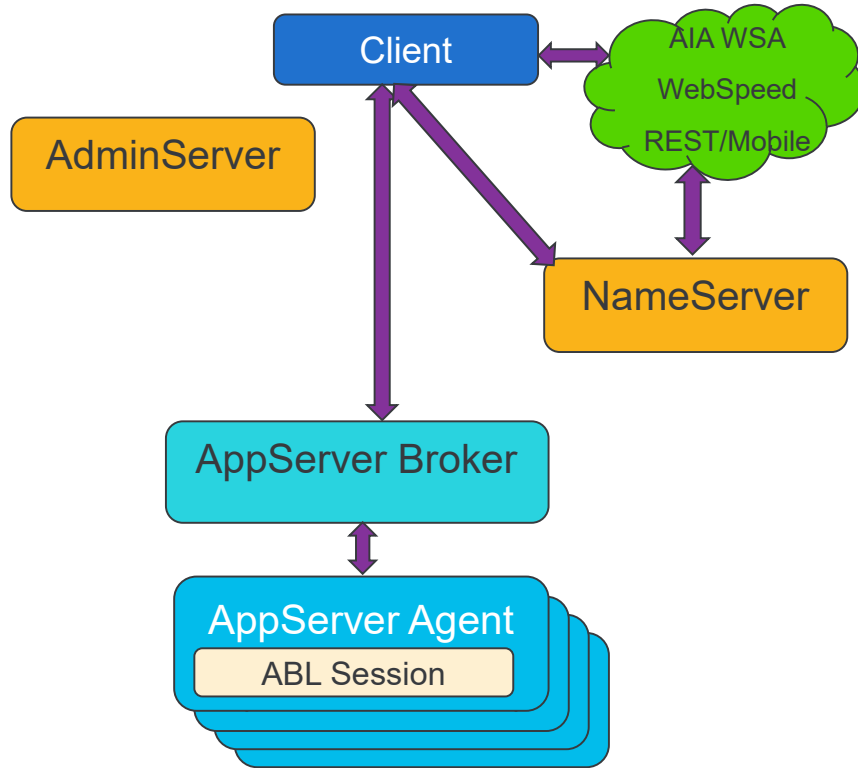
PAS for OpenEdge Components

- Client
- **PAS for** OpenEdge
  - APSV (AIA) SOAP (WSA) REST/Mobile WebSpeed
  - **Session Manager**
- **MSAgent**
  - ABL Sessions
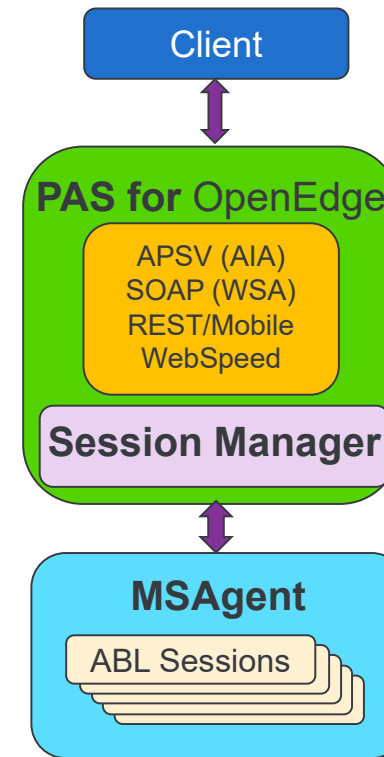
PASOE – find the AppServer components

IAP

AppServer Web Stack

PAS for **OE Web Stack**

Tomcat
- AIA
- SOAP (WSA)
- REST

uBroker

Agent
- ABL Session

1 named AppServer Instance

PAS
- PAS extensions
- OEPAS extensions

APSV (AIA) SOAP (WSA) REST

Session Manager

1 **oeabl** web application

MSAgent
- ABL Sessions

1 named PAS for OE Instance
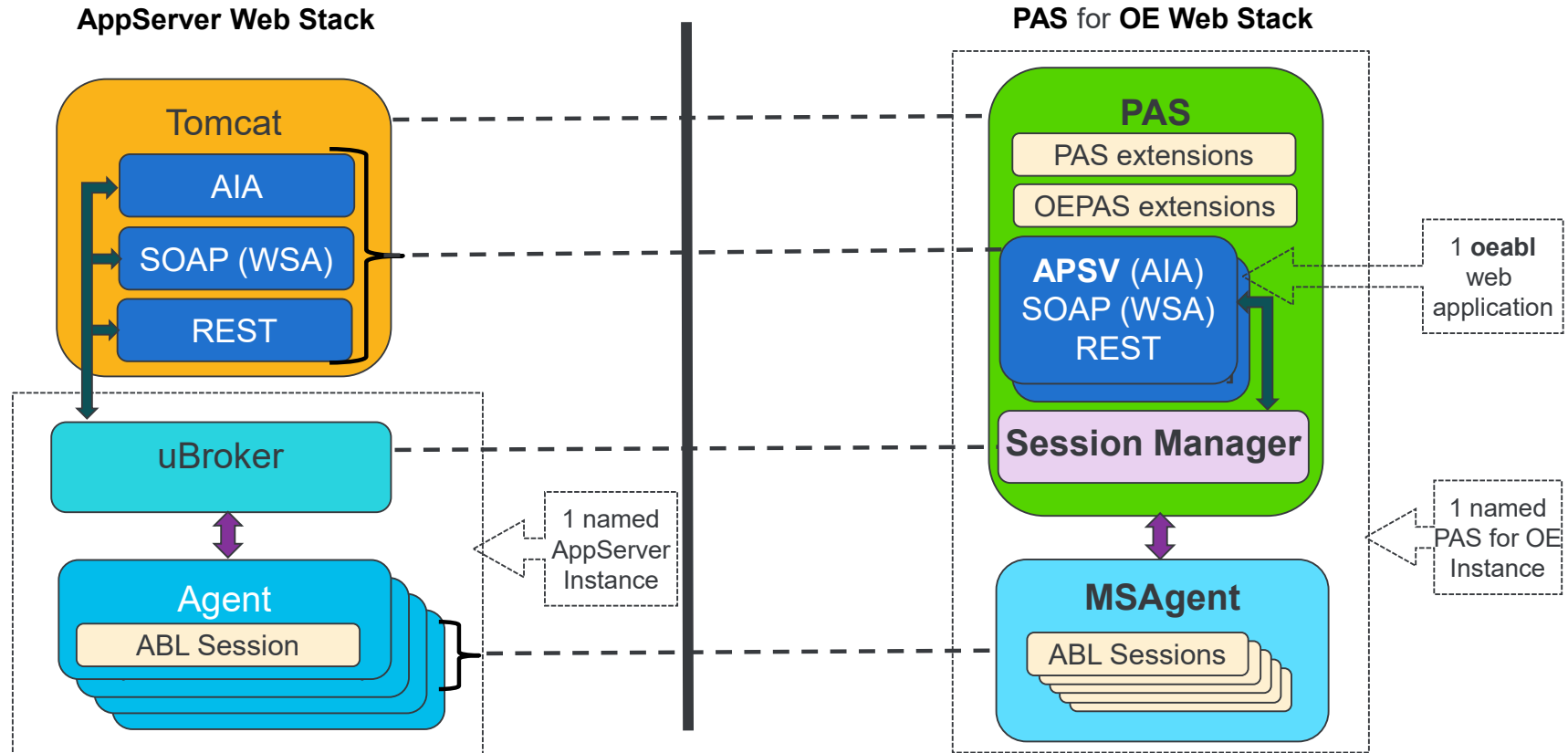
PASOE Development versus Production

PASOE for OE Development	PASOE for OE Production
Can compile ABL source code	Cannot compile ABL source code
Restricted: 5 concurrent requests and 1 agent	Unrestricted: concurrent requests and agents
Non-secure configuration	Secure configuration
Test server instance in $WRKDIR	No test server instances

Remote administration included
    Tomcat/Openedge Manager Apps
	Remote administration optional
	    Tomcat/OpenEdge Manager Apps

Built-in oeabl web application (ROOT)
    All transport deployed and enabled
	When instance created oeabl web application (ROOT):
	    All transports deployed but disabled

defaultHandler= OpenEdge.Web.CompatiblityWebHandler	defaultHandler= OpenEdge.Web.DefaultWebHandler

PASOE Transport Layer Handler

IAP

- REST
  - the old REST Adapter of the classic Appserver
- SOAP
  - the old Web Service Adapter of classic AppServer
- APSV
  - the classic adapter OpenEdge client and OpenClient
- WEB
  - the new adapter for REST and SOAP web services
  - very flexible
  - easy to handle and configurate

Transport Layer

PASOE at Instance Create and Run-time
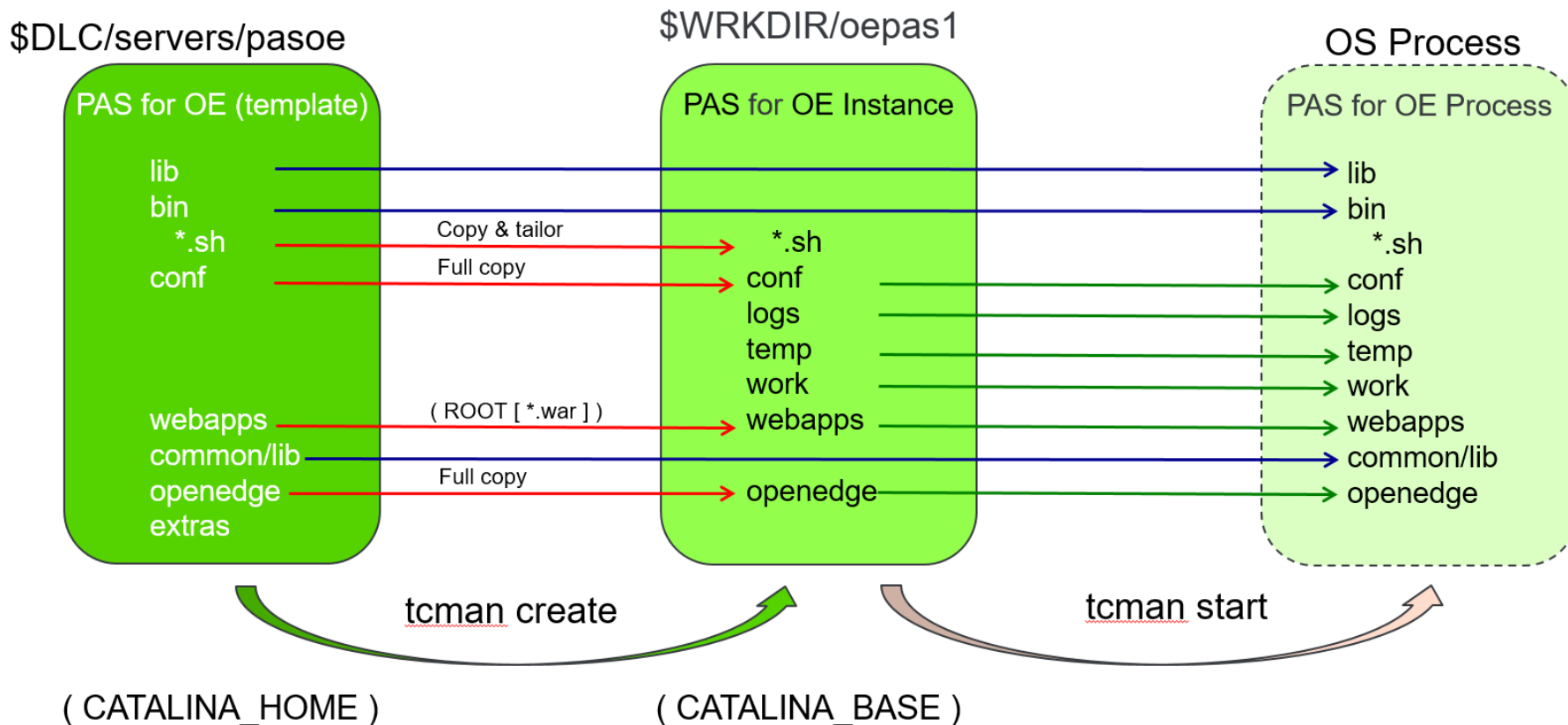
Important files and directories

- CATALINA_HOME

  Template root

  Contains also all Tomcat executables

  Location $DLC/severs/pasoe

- CATALINA_BASE

  PASOE instance

  Contains the web applications

  Contains the instance related scripts and settings

  Location i.e. oepas1 $DLC_WRK/oepas1

- $CATALINA_BASE/$OEPAS_instance/logs/$OEPAS_instance.agent.$date.log

  i.e. d:\dlc\122-wrk.64bit\oepas1\logs\oepas1.agent.2023-04-07.log

  OpenEdge agent log file

- $CATALINA_BASE/$OEPAS_instance/logs/startupErrors

  contains the error if an instance can't be started
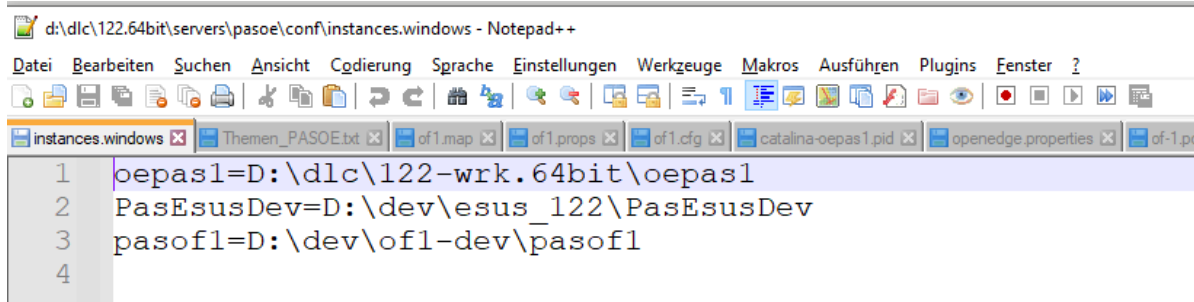
Important files and directories

- ## $CATALINA_BASE/conf/openedge.properties

  Appserver configuration file

  the former ubroker.properties file in the classic appserver

- ## $CATALINA_BASE/conf/catalina.properties

  Contains all settings for the PASOE Tomcat instance like ports etc.

- ## $CATALINA_BASE/conf/server.xml

  The Tomcat configuration file

  Copy of $CATALINA_BASE/conf/server.xml

  The settings are here variables set in catalina.properties
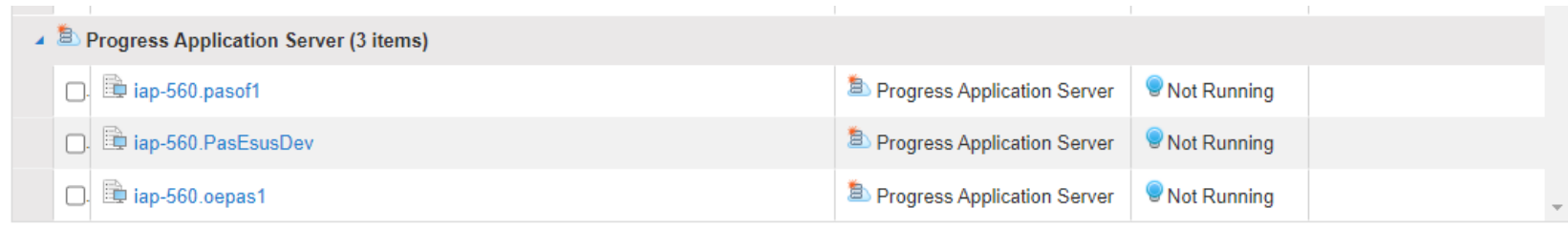
Important files and directories

- {$CATALINA_HOME}\conf\instances.windows

    Contains all known PASOE instances

TCMAN

- TCMAN is a command-line utility for managing and administering a server instance.

- Syntax: {$CATALINA_HOME|$CATALINA_BASE}
  /bin/tcman action [general_options] [action_options]

- For Windows use tcman.bat, for UNIX use tcman.sh

- Frequent commands for developers

create	Create a new server instance
delete	Delete server instance
start	Start an instance
stop	Stop a running instance

Create a Rest service with PDS

```
  7      Author(s)    : klaus.devries
  8      Created      : Fri Apr 07 15:35:13 CEST 2023
  9      Notes        :
 10    ----------------------------------------------------------------*/
 11
 12⊕USING Progress.Lang.*.▢
 16
 17  BLOCK-LEVEL ON ERROR UNDO, THROW.
 18
 19  CLASS playground.FinPUGRest INHERITS WebHandler:
 20
 21⊕   METHOD OVERRIDE PROTECTED INTEGER HandleNotAllowedMethod( INPUT poRequest AS OpenEdge.Web.IWebRequest ):▢
 26
 27⊕   METHOD OVERRIDE PROTECTED INTEGER HandleNotImplemented( INPUT poRequest AS OpenEdge.Web.IWebRequest ):▢
 32
 33
 34
 35    /*--------------------------------------------------------------
 36            Purpose: Default handler for the HTTP GET method. The request being
 37                     serviced and an optional status code is returned. A zero or
 38                     null value means this method will deal with all errors.
 39            Notes:
 40    --------------------------------------------------------------*/
 41⊕   METHOD OVERRIDE PROTECTED INTEGER HandleGet( INPUT poRequest AS OpenEdge.Web.IWebRequest ):▢
 98
 99
100    /*--------------------------------------------------------------
101            Purpose:
102            Notes:
103    --------------------------------------------------------------*/
104⊕   METHOD OVERRIDE PROTECTED INTEGER HandlePut( INPUT poRequest AS OpenEdge.Web.IWebRequest ):▢
108
109
110    /*--------------------------------------------------------------
111            Purpose:
112            Notes:
113    --------------------------------------------------------------*/
114⊕   METHOD OVERRIDE PROTECTED INTEGER HandlePost( INPUT poRequest AS OpenEdge.Web.IWebRequest ):▢
118
119    /*--------------------------------------------------------------
120            Purpose:
121            Notes:
122    --------------------------------------------------------------*/
123⊖    METHOD OVERRIDE PROTECTED INTEGER HandleDelete( INPUT poRequest AS OpenEdge.Web.IWebRequest ):
124
125      UNDO, THROW NEW Progress.Lang.AppError("METHOD NOT IMPLEMENTED").
126      END METHOD.
127
128
129  END CLASS.
```
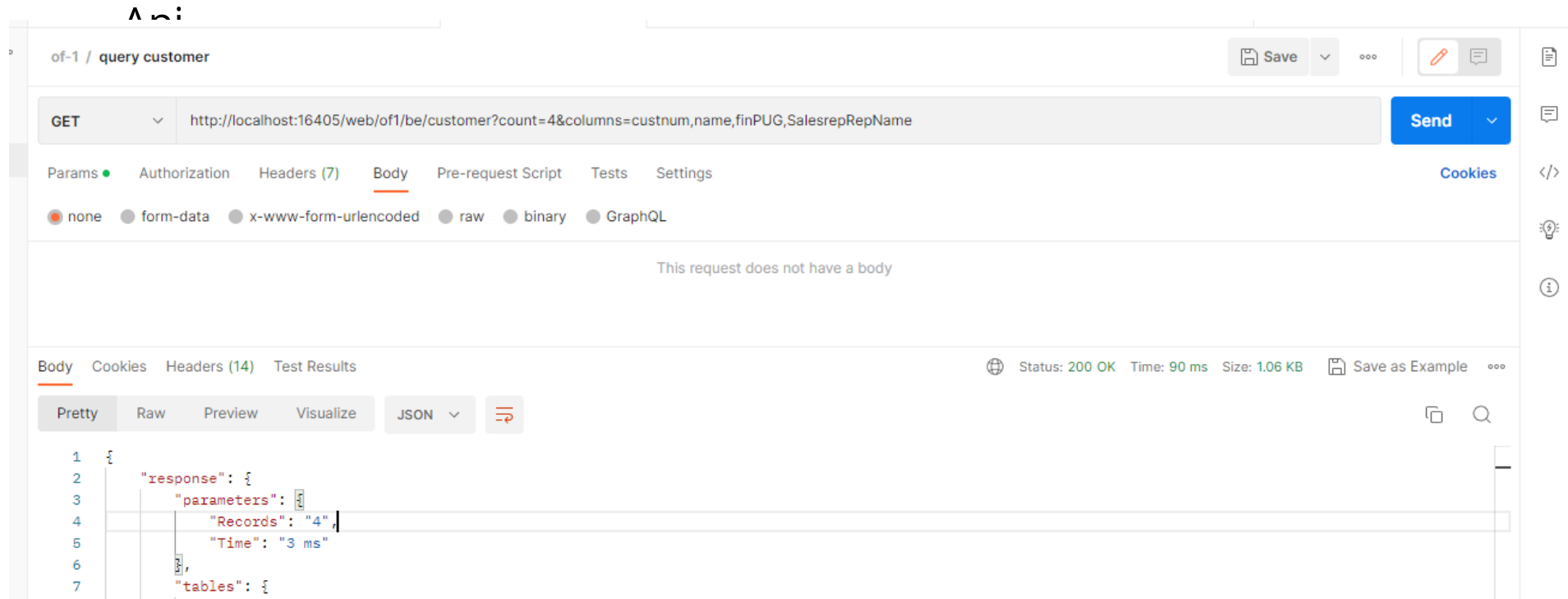
Create a Rest service Demo

- Create an entry in openedge.properties

- Create a handler for GET

- Parse parameter

- Access to the data base

- Send back a JSON file

  –

Test with Postman

- Postman is very popolar tool for testing REST Api

Swagger

- Swagger is a collection of open source tools

- For designing, creating, documenting and using HTTP web services

- Swagger uses the OpenAPI description standard.

Swagger OF-1 Rest API Documentation

Swagger OF-1 Rest API Documentation

Additional infos

- https://pugchallenge.org/downloads2017/284_web_handlers_deep_dive.pdf

-

- Ask us
  Klaus de Vries   kdv@iap.de
  Bernd Hellmann   bh@iap.de

www.iap.de/of-1

info@iap.de