Guía instalación entorno local APX BBVA

1. Prerrequisitos

* Modificar standalone para JBOSS con la configuración para las bases de datos locales. Para ello nos vamos a D:/ENTORNO\_LOCAL\_APX/JBOSS/standalone/configuration y abrimos el archivo standalone.xml. Buscamos el apartado <xa-datasource> que estarán las conexiones a bd. Si no hacen falta bases de datos no es necesaria esta parte.

Texto

Descripción generada automáticamente

En el apartado de <drivers> se indica el tipo de bd

Texto

Descripción generada automáticamente

* En D:/ENTORNO\_LOCAL\_APX/dirBaseRepositorio se encuentran las UUAAs junto a sus jar. Estos jar serán los que se ejecuten en el servidor y que luego podremos usar para debuggear. Dentro de cada UUAA se dividen dos carpetas: online y batch. En ellas tenemos una carpeta jar con una carpeta modules que dentro se dividirán las librerías, dtos y transacciones.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Tener la base de datos configurada en local, para Oracle es mas complicado ya que se necesita una máquina virtual (Docker por ejemplo) pero si las sentencias SQL son sencillas se podrían adaptar a postgres. Para ello instalar postgres y crear la base de datos y añadir su conexión al standalone de JBOSS. <https://www.postgresql.org/download/windows/>

1. Entorno local

Iniciar APX Online ejecutando “startOnline.bat” de la carpeta ENTORNO\_LOCAL\_APX. Esto creará una partición de disco D: donde se ejecutará el servidor local.

Interfaz de usuario gráfica

Descripción generada automáticamenteSe abrirán dos ventanas en la que una indica el estado del servidor hasta que esté completamente arrancado y la otra muestra los logs de JBOSS del servidor.

Cuando en el log de la izquierda, donde pone “ARCHITECTURE NOT AVAILABLE” ya haya finalizado y salga así

Texto

Descripción generada automáticamente

nos iremos a las siguientes URLS:

·       APX ONLINE -<http://localhost:8101/APX_Operation/online/components?iv-user=ADMIN>

·         Félix -<http://localhost:8101/APX_Runtime_Online/system/console/bundles>

* 1. APX ONLINE

En la ruta del ONLINE hay que importar unos JSON con variables para los proyectos (UUAA), para ello hay que ir a Configuration - Online y poner en el buscador la UUAA correspondiente para importar las variables.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Una vez que se ha buscado, por ejemplo, KBGE encima de los botones Import y Export tiene que poner *Configuration (Online) KBGE*

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Se hace clic en el botón Import y se selecciona el JSON de las variables de KBGE

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si todo es correcto, después de hacer clic en el botón OK de la imagen anterior, saldrá una imagen como la siguiente y solo habría que hacer clic en el botón *Import*.

Tabla

Descripción generada automáticamente

* 1. APACHE FÉLIX

En félix la primera vez (y de vez en cuando) te pedirá usuario y clave hay que poner **admin**/**admin** (user/pass).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamenteEn esta dirección podremos ver los jar de los distintos componentes que hayamos subido. Tendremos transacciones, librerías y DTOs.

Texto

Descripción generada automáticamenteEn esta parte lo más importante es comprobar que los componentes estén exponiendo servicio. Para ello nos vamos a la implementación de un componente (KTRAR022IMPL) y comprobamos que las dependencias y los import sean acordes a la versión de ese componente. Para saber si está exponiendo un componente debe aparecer service ID junto a un número.

Cuando se importan las variables de entorno a través del APX ONLINE debemos reiniciar los componentes que usen esas variables. Simplemente paramos el servicio y lo volvemos a iniciar y ya tendrá las variables actualizadas.

1. Ejecución

Una vez tengamos los componentes exponiendo servicios ya podremos ejecutar las transacciones.

Para ello, desde thunder client o postman creamos una conexión con la siguiente URL: <http://localhost:8101/APX_Runtime_Online/PGRestService>

En el body usaremos el json de entrada de la transacción y en las cabeceras deberemos poner las correspondientes para llamar a la transacción.

Estas cabeceras serán:

logical-transaction-code: En esta cabecera pondremos el nombre de la UUAA. Ejemplo: KBGET001, KBGET002. Hasta ahí solo

type-code: Normalmente es 01.

Subtype-code: Normalmente es 01.

version-code: La versión de la transacción. Ejemplo: 03 -> Sería de KBGET001-**03-**ZZ

country-code: El código de país de la transacción: ZZ, ES, MX …

user-code: el usuario que ejecute la transacción debe ser válido para esa transacción.

Dejo aquí las cabeceras que suelo usar, solo habría que modificar las que necesitéis.

Accept: \*/\*

User-Agent: Thunder Client (https://www.thunderclient.com)

Content-Type: application/json

header-name: QP06

logical-transaction-code: KBGET001

type-code: 01

subtype-code: 01

version-code: 03

country-code: ZZ

entity-code: 0182

branch-code: 1128

workstation-code: 0

operative-entity-code: 0182

operative-branch-code: 6252

ip-address: 0.0.0.0

channel-code: 01

environ-code: 01

product-code: 0001

language-code: es

user-code: IMMA120

operation-date: 20160316

operation-time: 164948

currency-code: EUR

secondary-currency-code: EUR

authorization-version: 0

authorization-code: 0

client-identification-type: 6

client-document: IMMB120LD844389LDDXE70414

contact-identifier: empty

request-id: ddd192fd-adc9-4e84-b99f-f225b6654cd8

service-id: 1

aap: 00000016

app:

Origin: aa

remaining-time: 4000

agent-user:

manager-user:

mac:

accounting-terminal:

logical-terminal:

sequence-id:

1. Debug en eclipse

Una vez compilado y arrancado el servidor se puede poner el IDE en modo debug, para ello hay que:

1. Hacer clic en *Debug Configurations…*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Hay que crear, en *Remote Java Application* una nueva tarea. En Project hay que indicar la transacción que quieres probar.

En connection Type hay que poner la configuración de la siguiente imagen.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Para poder debugguear hay que añadir, en la pestaña source, los proyectos que se quieren debuggear.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Finalmente, pinchamos en debug y se arrancarán los proyectos apuntando al servidor local. **Importante** que lo que haya en el workspace de eclipse se corresponda al jar instalado en el servidor local.

Si se quiere hacer algún cambio, se debe actualizar el jar instalado en la ruta D:ENTORNO\_LOCAL\_APX/dirBaseRepositorio del entorno local. Automáticamente, tras cierto tiempo el propio servidor actualiza el jar y ya tendrías los cambios en el modo debug.

1. Problemas
2. Puede ser que levantemos el servidor y no arranque el online, una de las posibles razones es porque los war no se han deployado

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para solucionar esto hay que borrar los que estén *undeployed* y se vuelve a desplegar (no hace falta para el servidor)

1. Si se produce un error al recuperar una constante

Imagen que contiene Calendario

Descripción generada automáticamente

En este caso se produce un numberException porque no recupera el valor de la constante.

Texto, Carta

Descripción generada automáticamente

Hay que mirar si está declarada esa variable en el ONLINE

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si está declarada lo que hay que hacer es, en Félix, parar y arrancar el componente que ha dado el error y ya debería de funcionar. Si no está declarada se añade a la UUAA como nueva variable junto a su valor y volver a parar y arrancar el componente.