Debug Mode Component: Testing Service App

Esta sección explica el proceso para realizar debug en los componentes que contienen los servicios que se migran y exponen.

Testing Service App

Son conng		JSON Data	
("company_token": "3421cff2f6b	2117dfaf86f8398(^	{"user": "SICOF", "password": "sicof@	0123456789", "ip":"172.16.2
<	>	<	>
Service Class		Launch Type	
n_cst_service_login		Event	
vent ue_login		Validation Type Company Token V	alidation
Execution time: 547	milliseconds.	copy to dipboard	Run Test
<pre>` "return_code": "1", "return_user_message": "ses "return_technical_message": "return_object": { "token_session": "43443a3 }, "return_response_date": "28/ }</pre>	ión vigente", "sesión vigente", baf7292e48865f52156 05/2019 16:23:35"	66ba30e46a96b80fb7ea837b1f1ea57f0	86697a"

Es una utilidad integrada en el framework el cual permite ejecutar los servicios sin necesidad de generar un despliegue.

Ventajas

- No se requiere un despliegue para realizar el debug
- Modo escritorio para debuggear las aplicaciones
- Velocidad en el proceso
- No se requiere aplicaciones de terceros
- Compatibilidad 100% con la versión desplegada

Desventajas

• Se deben adicionar todas las librerias del proyecto

Last update: 2019/05/29 ada:tips:sicoferp:general:pbtows:procesos:debug http://wiki.adacsc.co/doku.php?id=ada:tips:sicoferp:general:pbtows:procesos:debug 12:16

- Se debe mantener actualizado todo el repositorio de componentes
- Se debe tener configurado el acceso local a servicios y bases de datos

Estructura de la Información

La utilidad se divide en 8 zonas las cuales serán explicadas a continuación:

ISON Config 1		JSON Data 2
{"company_token":	"3421cff2f6b2117dfaf86f839	98(^ {"user": "SICOF", "password": "sicof@0123456789", "ip": "172.16.2
< _		> <
Service Class 3)	Launch Type 🖊
n_cst_service_login		Event
Event ue_login	5	Validation Type Company Token Validation 6
Execution time:	547 milliseco	onds. copy to dipboard 7 Run Test
Console		
<pre>{ "return_code": " "return_user_me "return_technica "return_object": "token_sessio }, "return_response }</pre>	1", Issage": "sesión vigente", I_message": "sesión vigente" { n": "43443a3baf7292e48865i e_date": "28/05/2019 16:23:	, ", 5f521566ba30e46a96b80fb7ea837b1f1ea57f086697a" :35"
,		

Zona 1: JSON Config

Esta zona representa la captura de la cadena de configuración del consumo que se va a testear.

Zona 2: JSON Data

Esta zona representa la captura de la cadena de la data del consumo que se va a testear.

Zona 3: Service Class

Esta zona representa la clase que contiene el servicio del consumo que se va a testear.

Zona 4: Launch Type

Esta zona representa el tipo de lanzamiento (ejecución) del servicio del consumo que se va a testear.

Zona 5: Event

Esta zona representa el evento de lanzamiento del servicio del consumo que se va a testear.

Zona 6: Validation Type

Esta zona representa el tipo de validación de acceso (autenticación de sesión) del servicio del consumo que se va a testear.

Zona 7: Process

Esta zona representa el area de procesos la cual dispone de los siguientes elementos:

- Execution Time: Tiempo de ejecución (en milisegundos) del último servicio ejecutado.
- copy to clipboard: Copiar la respuesta del servicio al portapapeles.
- Run Test: Ejecutar el consumo del servicio configurado.

Zona 8: Console

Esta zona representa la salida por consola del servicio del consumo que se va a testear.

Como registrar un componente

Para registrar un componente siga los siguientes pasos:

Paso 1: Proyecto

Abra el proyecto principal. Ubique y seleccione el proyecto que se ve en la imagen (La ruta puede variar según su copia local).



Last update: 2019/05/29 ada:tips:sicoferp:general:pbtows:procesos:debug http://wiki.adacsc.co/doku.php?id=ada:tips:sicoferp:general:pbtows:procesos:debug 12:16

Paso 2: Check Out Proyecto

Asegurese que puede adicionar al contenido del proyecto el componente que va a testear. Para esto ejecute un Check Out en la raíz del proyecto como se ve en la imagen.

E SICOF PBtoWS (D: \ADA \SICO	OF\SICOF PBtoWS\Local\Branch	es\branches	; 12.5.2.5.0)
🖶 🖕 🎯 ws00_company (d: \ada	\sicof\sicof pbtows\local\branch	es\branches	12.5.2.5.0\00_main\company)
🗄 🔩 😪 ws00_login (d: \ada \sico	f\sicof pbtows\local\branches\b	ranches 12.5	5.2.5.0\00_main\ogin)
🗄 🔩 😪 ws00_storage_bag (d: 🖓	ada\sicof\sicof pbtows\local\bra	nches\brand	ches 12.5.2.5.0\00_main\storagebag)
🗄 🔩 😪 ws00_utilities (d: \ada \si	cof\sicof pbtows\local\branches	branches 12	.2.5.2.5.0\00_main\util)
🗄 🔩 😪 ws03_tercero (d: \ada \s	icof\sicof pbtows\ocal\branches	s\branches 1	12.5.2.5.0\03_contabilidad\tercero)
👾 😱 🛞 ws00_proxies (d: \ada \si	icof\sicof pbtows\ocal\branches	s\branches 1	12.5.2.5.0\proxy)
🗄 🖕 🛞 test_sicofpbtows (d:	:\ada\sicof\sicof pbtows\lo	cal\branch	hes\branches 12.5.2.5.0\test)
🗄 🗣 🛞 wsxx_🕻 SCC Check Ou	Jt		hes 12.5.2.5.0\test\capacitacion)
Select mu	ultiple files contained in this targ operation on this one target file OK Cancel	jet only	

Paso 3: Adicionar las Librerias

Una vez bloqueada la raíz del proyecto presione el click derecho del mouse y vaya a las propiedades, dirijase a la sección de librerias como se aprecia en la imagen.



Adicione las librerias del proyecto. (No adicione las librerias del framework ya que previamente han sido agregadas).

Paso 4: Ubicar los puntos de ruptura

Una vez adicionadas las librerias del componente ubique las clases del componente que desea testear y adicione los puntos de ruptura según su criterio de debug como se aprecia en la imagen.



Paso 5: Ejecutar

Una vez configurados los puntos de ruptura ejecute la aplicación e ingrese la siguiente información:

 Cadena de Configuración (Json Config): Requerida para la autenticación en el ecosistema de servicios.

- Cadena de la Data (Json Data): Si el servicio lo requiere debe ingresar en este campo la data de consumo
- Clase de Servicio (Service Class): Ingrese la clase del servicio que desea testear.
- Tipo de Lanzamiento (Lauch Type): Todo el ecosistema es invocado por medio de eventos iniciales. Este campo ya esta predefinido.
- Evento (Evento): Ingrese el nombre del evento lanzador del servicio.
- Tipo de Validación de Autenticación (Validation Type):Campo requerido que identifica el tipo de autenticación que realizará el servicio debe tener presente que este campo está asociado al Json Config.
- Lanzar Servicio (Run Test): Una vez ingresada la información requerida ejecute el servicio por medio de este boton.

Según el modo de ejecución, verá el paso a paso o la salida por consola del consumo como cualquier debug en powerbuilder classic.

ISON Config		JSON Data		
{"company_token": "3421cff2f	6b2117dfaf86f8398c ^	{"user": "SICOF", "password": "sicof@012	3456789", "ip":"172.16.2 ^	
<	>	<	>	
Service Class		Launch Type		
n_cst_service_login		Event	~	
Event ue_login		Validation Type Company Token Valid	ation	
Execution time: 547	milliseconds.	copy to dipboard	Run Test	
<pre>{ "return_code": "1", "return_user_message": "s "return_technical_message" "return_object": { "token_session": "43443 }, "return_response_date": "2 }</pre>	esión vigente", ": "sesión vigente", a3baf7292e48865f5215 8/05/2019 16:23:35"	566ba30e46a96b80fb7ea837b1f1ea57f0866	97a"	

