**Configuración de Wrapper Provider en el cliente:**

|  |  |
| --- | --- |
| Título: | Procedimiento de configuración wrapper |
| Versión | 1.3 |
| Versión Fwk | 12.0 y 10.4 |
| Fecha de confección: | 10/11/2016 |

El wrapper de conexión de la arquitectura fwk es quien provee de toda la funcionalidad de conexión, manejo de errores, y flexibilidad para poder comunicar un cliente o server cualquiera con el/los despachadores de servicio que sean necesario

**Proveedor de configuración de wrapper**

El wrapper de conexión está basado en Providers. El provider define las siguiente opciones de configuración

* **type** Tipo de dispatcher servidor (local, remoto web service , WCF http, web api etc)
* **appId** Identificador de compañía o empresa o aplicación
* **serviceMetadataProviderInfo** Proveedor de metadato de servicios. (A que configuración de metadata de servicios y seguridad del lado del servidor apunta el cliente)
* **sourceinfo =** Puede ser la url donde está el dispatcher su configuración depende del atributo type
* **name:** Nombre identificador y unico del wrapper

**Procedimiento de configuración de wrapper**

Pasos:

1. Ubicar el archivo de configuración de la aplicación (.config)
2. Agregar la siguiente sección

<configSections>

<section name="FwkWrapper" type="Fwk.ConfigSection.WrapperProviderSection, Fwk.Bases" />

</configSections>

1. Agregar la configuración de las sección:

<FwkWrapper> </FwkWrapper>

1. Dentro de la sección anterior agregar todos proveedores de wrapper que sean necesarios.
2. Serán necesarios tantos proveedores como solicitudes a servicios diferentes existan para la aplicación. Como mínimo debe existir una, que será el proveedor por defecto.
3. Además será necesario establecer en la sección el atributo **defaultProviderName** que indicara cual es el proveedor de wrapper por defecto.

<FwkWrapper defaultProviderName="RecursosHumanos">

<Providers>

<add name="XML\_test" type="Fwk.Bases.Connector.LocalWrapper,Fwk.Bases.Connector" sourceinfo="" appId="" serviceMetadataProviderInfo="test"/>

</Providers>

</FwkWrapper>

**Especificaciones técnicas:**

En la siguiente tabla se detallan el significado de cada atributo del proveedor.

|  |  |
| --- | --- |
| name | Representa el nombre del proveedor del wrapper. |
| appId | Es el nombre o identificador de aplicación. Este atributo es útil para identificar otras cosas en la capa de negocio, el identificador  Este atributo es la semilla inicial para una implementación multiempresa.  Ver documento: Arquitectura Tecnológica - Modelo para identificar empresa V 2.0.doc |
| serviceMetadataProviderName | Es el nombre del proveedor de servicio que se utilizara a fin de definir configuración de metadatos de servicio y de seguridad. El nombre deberá corresponder a un FwkServiceMetadataProvider en el dispatcher.  Ver documento: Arquitectura Tecnológica Configuración Metadata de Servicios |
| sourceinfo | Este valor representa la conexión al servidor y depende del tipo de wrapper a utilizar. Depende por lo tanto del atributo **type:**  **LocalWrapper** : “”  **WebServiceWrapper** : La URL al Web Service  **WCFWrapper** : La URL al Web Service o Ruta + nombre de archivo de configuración remoting  **RemotingWrapper**: Ruta + nombre de archivo de configuración remoting |
| type | Información sobre el tipo de wrapper que se está utilizando:  Fwk.Bases.Connector.**LocalWrapper**,Fwk.Bases.Connector  Fwk.Bases.Connector.**WebServiceWrapper**,Fwk.Bases.Connector  Fwk.Bases.Connector.**RemotingWrapper**,Fwk.Bases.Connector  Fwk.Bases.Connector.**WCFWrapper\_NetTcpBinding**,Fwk.Bases.Connector  Ejemplo sourceinfo="net.tcp://localhost:8001/FwkService"  Fwk.Bases.Connector.**WCFWrapper\_BasicHttpBinding**,Fwk.Bases.Connector  Ejemplo sourceinfo=<http://pelsoftsite2.itempurl.com/service.svc>  Fwk.Bases.Connector.REST\_APIWrapper, Fwk.Bases.Connector  Ejemplo sourceinfo="http://localhost:63251/ |

Configuración según el tipo de wrapper *type*

|  |  |
| --- | --- |
| local | sourceInfo = “”  type="Fwk.Bases.Connector.**LocalWrapper**,Fwk.Bases.Connector" |
| Remoting service | sourceInfo = “Ruta + nombre de archivo de configuración remoting”  type="Fwk.Bases.Connector.**RemoteWrapper**,Fwk.Bases.Connector" |
| Web service | sourceInfo = “URL del web service”  type="Fwk.Bases.Connector.**WebServiceWrapper**,Fwk.Bases.Connector" |
| WCFWrapper\_NetTcpBinding | WCF Wrapper que utiliza **NetTcpBinding** <add name="wrap\_wcf" type="**Fwk.Bases.Connector.WCFWrapper\_NetTcpBinding,Fwk.Bases.Connector**" appId="clientes" serviceMetadataProviderName="metadata\_on\_server\_clientes" sourceinfo="**net.tcp://localhost:8001/FwkService**"/> |
| WCFWrapper\_BasicHttpBinding | WCF Wrapper que utiliza WsHttpBinding  <add name="wrap\_wcf" type="Fwk.Bases.Connector.WCFWrapper\_ **BasicHttpBinding**,Fwk.Bases.Connector" appId="clientes" serviceMetadataProviderName="metadata\_on\_server\_clientes" sourceinfo=" [http://localhost:8732/WcfDispatcher/FwkService"/](http://localhost:8732/WcfDispatcher/FwkService%22/)> |
| REST\_APIWrapper | <add name="REST\_APIWrapper" type="Fwk.Bases.Connector.REST\_APIWrapper,Health.Front.Base" appId="pelsoft"  serviceMetadataProviderName="health" sourceinfo="http://localhost:63251/" / |

Dispatchers para cada wrapper

Excepto para wrapper de tipo local, todos los demás tipos van a requerir un host remoto que los escuche. Actualmente existen desarrollados dispatchers para cada uno de los tipos en la sección Dispatcher del FWK.

Incluso existe una versión Consola que es muy utilizada en momentos de pruebas.

Las versiones consola desarrollados son para **RemoteWrapper** (Remoting) y WCFWrapper\_NetTcpBinding

Ejemplo:

<FwkWrapper defaultProviderName="health\_local">

<Providers>

<add name="health\_local" type="Fwk.Bases.Connector.LocalWrapper,Fwk.Bases.Connector" appId="pelsoft" serviceMetadataProviderName="health" sourceinfo="" />

<add name="health\_ws" type="Fwk.Bases.Connector.WebServiceWrapper,Fwk.Bases.Connector" appId="pelsoft" serviceMetadataProviderName="health" sourceinfo="http://192.168.1.189/health.WS/SingleService.asmx" />

</Providers>

</FwkWrapper>