

De lo abstracto a lo concreto en PaaSOS ESB

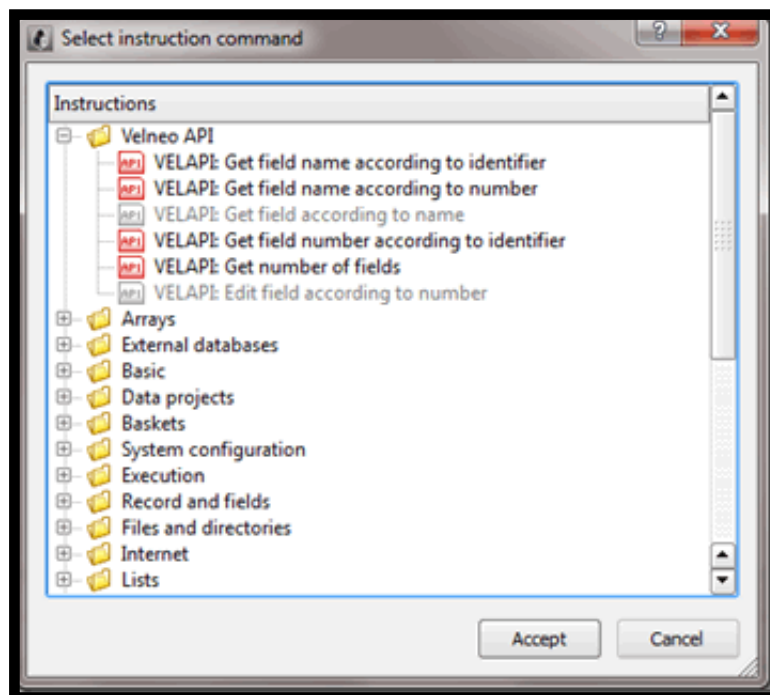
Esta claro que la arquitectura de v7 es muy abstracta, solo tenéis que revisaros el siguiente vídeo sobre arquitectura v7.

Todo son objetos procesables por su procesador. Una tabla, un formulario, una rejilla, un proceso, una búsqueda, todos son procesables por su procesador. Un procesador tiene una entrada, variables locales y una salida. Además los objetos disponen de propiedades, métodos, interface... Vamos, muy abstracto, todo es lo mismo.

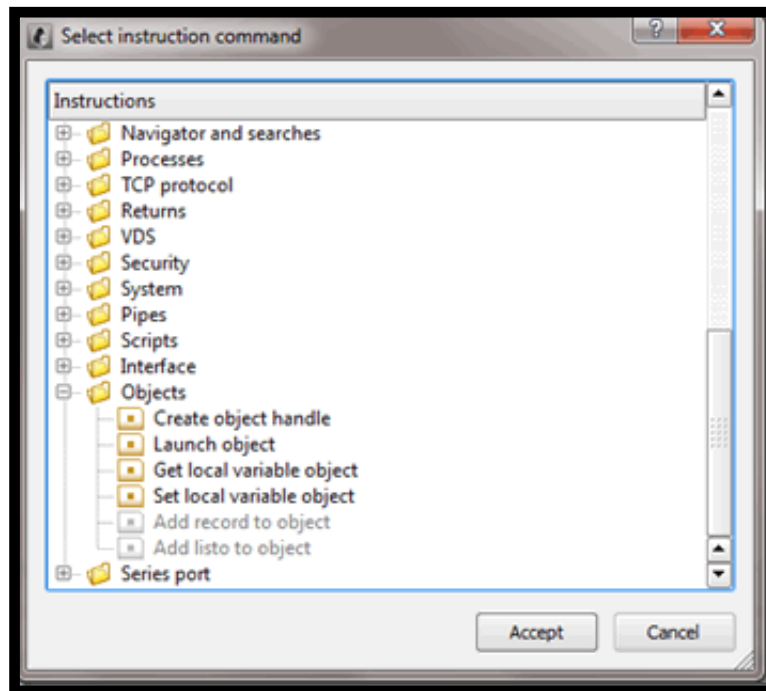
Ahora bien, si desarrollamos mediante v7 veremos que sucede lo contrario, casi todo lo que hacemos lo hacemos de forma concreta. Cuando creamos un proceso lo asociamos a un cierto tipo de tabla, lo mismo con una rejilla, ídem con los slots... en resumen lo que en su núcleo es muy abstracto en la practica es totalmente concreto. Esta forma de entender v7 hace que sea sencillo de usar aún siendo tan abstracto en su núcleo.

Como contrapartida nos encontramos con bastantes dificultades cuando queremos acceder de forma abstracta a los datos u objetos de Velneo v7.

Para intentar solventar esta situación Velneo v7 dispone de ciertas funciones de API:



y otras de manejo de objetos:



Pero ambos conjuntos son concretos (necesitan el objeto concreto), por lo que no resuelven ciertas necesidades avanzadas.

La actuales funciones API de Velneo y de manejo de objetos son insuficientes para resolver ciertas situaciones por ser demasiado concretas.

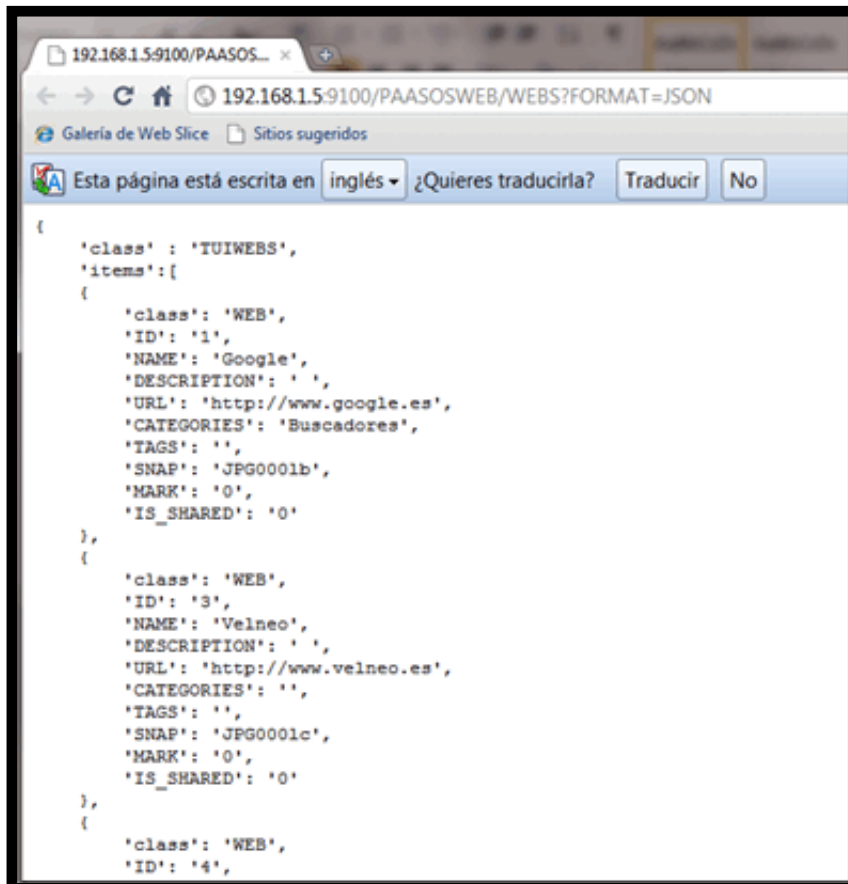
Vamos a intentar poner un ejemplo claro de las limitaciones de la actual API: Estamos planificando como será la versión Web+Ajax de **PaaSOS** y hemos empezado con las primeras pruebas de como construir un **ESB** (bus de servicios empresariales) sobre **vModApache**. Este bus será el interface de acceso estándar a **PaaSOS**. El futuro **ESB** de **PaaSOS** permitirá la construcción de cualquier tipo de aplicación empresarial basada en **PaaSOS**.

Hemos desarrollado un primer proceso que permita obtener una lista de elementos de una cierta tabla en cualquiera de los formatos deseados (HTML, CSV, XML si tipo, json...).



```

<?xml version="1.0" encoding="UTF-8" ?>
- <WEBS>
- <WEB>
  <ID>1</ID>
  <NAME>Google</NAME>
  <DESCRIPTION />
  <URL>http://www.google.es</URL>
  <CATEGORIES>Buscadores</CATEGORIES>
  <TAGS />
  <SNAP>JPG0001b.png</SNAP>
  <MARK>0</MARK>
  <IS_SHARED>0</IS_SHARED>
</WEB>
- <WEB>
  <ID>3</ID>
  <NAME>Velneo</NAME>
  <DESCRIPTION />
  <URL>http://www.velneo.es</URL>
  <CATEGORIES />
  <TAGS />
  <SNAP>JPG0001c.png</SNAP>
  <MARK>0</MARK>
  <IS_SHARED>0</IS_SHARED>
</WEB>
  
```



```

{
  'class' : 'TUIWEBS',
  'items':{
    {
      'class': 'WEB',
      'ID': '1',
      'NAME': 'Google',
      'DESCRIPTION': ' ',
      'URL': 'http://www.google.es',
      'CATEGORIES': 'Buscadores',
      'TAGS': '',
      'SNAP': 'JPG0001b',
      'MARK': '0',
      'IS_SHARED': '0'
    },
    {
      'class': 'WEB',
      'ID': '3',
      'NAME': 'Velneo',
      'DESCRIPTION': ' ',
      'URL': 'http://www.velneo.es',
      'CATEGORIES': '',
      'TAGS': '',
      'SNAP': 'JPG0001c',
      'MARK': '0',
      'IS_SHARED': '0'
    },
    {
      'class': 'WEB',
      'ID': '4',
    }
  }
}
  
```

Este proceso es concreto (solo es de una tabla).

Con las actuales funciones de API podemos recorrer una tabla y retornar todos los

campos de la misma, pero tendríamos que crear un proceso único por cada tabla.

Si pensamos que en un ESB tradicional tenemos al menos cuatro funciones ([CRUD](#)) por objeto y **PaaSOS** tiene no menos de treinta módulos, tenemos mucho trabajo por delante (copiar y pegar).

El ideal sería que en las actuales funciones de API pudiese pasarse dinámicamente el identificador de una tabla u objeto. Si fuera de esta forma tendríamos un único conjunto de procesos comunes para todo nuestro ESB y el resultado sería mucho más gratificante.