Llamadas HTTP: **DAPI** v1.0 (DILVE - API)

v2 r2 09/2024

Septiembre 2024

Contenido

Introducción **1**

Estructura de las llamadas **1**
Parámetros 2

Respuestas **2**
Compresión 4

Errores **4**
Formato de la respuesta de error 4
Códigos de error 4

Utilización de DAPI **4**

getRecordListX **4**
Finalidad 4
Nombre de la llamada 5
Parámetros 5
Errores 6
Gestión del parámetro `program` 6
Gestión del parámetro `publisher` 6
Gestión del parámetro `country` 6
Resultados 7

getRecordStatusX **8**
Finalidad 8
Nombre de la llamada 8
Parámetros 8
Errores 10
Gestión del parámetro `program` 10
Gestión del parámetro `publisher` 10
Gestión del parámetro `country` 10
Timestamps 11
Resultados 11

getRecordsX **13**
Finalidad 13
Nombre de la llamada 14
Parámetros 14
Errores 15
Resultados 15

getResourceX **17**
Finalidad 17
Nombre de la llamada 17
Parámetros 18
Errores 18
Resultados 18

getResourceStatusX **18**
Finalidad 18
Nombre de la llamada 18
Parámetros 19
Errores 20
Timestamps 20
Resultados 20

Apéndice I: Esquemas XML de las respuestas **22**

getRecordListX 22
getRecordStatusX 24
getRecordsX 25
fichaLibros 26
getResourceX 27
getResourceStatusX 28

Apéndice II: Códigos de error de DAPI 30
Introducción 30
Familias de errores 30
Mapeos de errores 32

Introducción

La API (Application Programming Interface – Interfaz de programación de aplicaciones) de DILVE (DILVE-API o **DAPI**) es un conjunto de llamadas vía HTTP que permiten extraer información del repositorio desde un entorno de programación.

La API complementa a toda una serie de métodos y herramientas disponibles en DILVE para extraer la información en distintos formatos. Las extracciones de datos programadas a través de la interfaz y que pueden programarse para que se ejecuten de forma automática, permiten extraer los metadatos del catálogo completo de DILVE en todas las versiones del estándar ONIX, en CSV y en SINLI (en varias versiones). El detalle del formato de los archivos CSV (archivo de texto con los campos delimitados por comas u otro carácter) se configura en la gestión de los Mapas CSV. Estos mismos mapas se pueden utilizar en la descarga de metadatos mediante la API.

En relación a la selección del subconjunto de libros con se quieran manejar en cada caso, es posible hace búsquedas tanto por criterios de los propios metadatos (libros de texto, libros de un conjunto determinado de editoriales, libros digitales, etc.), como por criterios de gestión de los propios libros (fecha de alta en DILVE, libros modificados en un intervalo de tiempo, libros que han cambiado de precio, etc.). Estos criterios pueden configurarse o en la configuración de las propias extracciones, o en búsquedas guardadas que a su vez se utilizan en las propias extracciones. La API permite marcar algunos criterios de selección (editoriales, fechas de modificación, etc.), pero para los casos en los que dichos criterios no sean suficientes, es posible utilizar los criterios de una extracción configurada en la interfaz para la selección de libros de las consultas mediante la API.

Este conjunto de llamadas se divide en dos grupos. El primero (Gestión de registros) permite conocer la identificación (ISBN) de los registros incluidos en DILVE, así como las altas, modificaciones y borrados en un período. El segundo (Extracción de información) permite obtener el contenido (metadatos) de registros de DILVE, así como los ficheros de recursos asociados: imágenes, etc.

En su versión 2.0 DAPI está formada por las siguientes llamadas:

Gestión de registros

`getRecordListX`	permite obtener un listado (en XML) de los ISBN de los registros existentes en DILVE en el momento en que se hace la llamada
`getRecordStatusX`	permite obtener un listado (en XML) de los ISBN de los registros añadidos, modificados o borrados en DILVE entre dos momentos
`getResourceStatusX`	permite obtener un listado (en XML) de los ISBN de los registros en los que se haya modificado algún tipo de recurso refenciado.

Extracción de información

`getRecordsX`	permite obtener fichas de libros existentes en DILVE
`getResourceX`	permite descargar un archivo de recursos (imágenes, etc.), asociado a un ISBN, cargado en DILVE y del cual se conoce el nombre. Los nombres correspondientes se pueden obtener en el registro ONIX correspondiente obtenido mediante **`getRecordsX`**.

Utilización de DAPI

La API HTTP de DILVE está diseñada para incorporar las llamadas en una aplicación local, que gestione una copia local de una Base de Datos con la información de DILVE. En ningún caso deben utilizarse desde entornos web o JavaScript o transmitir directamente a DILVE solicitudes recibidas de un entorno público.

Para generar una copia local del contenido de DILVE (o de un subconjunto de los registros de DILVE), debe hacerse, inicialmente, una llamada a **`getRecordListX`** para obtener la lista de ISBN existentes en DILVE en la fecha en la que se inicie la copia.

Posteriormente se deben obtener, mediante **`getRecordsX`**, los registros correspondientes a la lista de ISBN devuelta por la llamada inicial a **`getRecordListX`**.

Es importante recordar que cada llamada a **`getRecordsX`** permite pedir hasta 128 registros en la misma llamada. Aunque la gestión de este tipo de llamadas con identificadores múltiples pueda complicar algo la programación, el rendimiento del sistema aumenta de forma notable por lo que se recomienda utilizar las llamadas con múltiples identificadores siempre que sea posible.

Si se va a hacer una réplica local de un número significativo de registros de DILVE será mucho más operativo hacer una *extracción* inicial para obtener directamente (vía FTP por ejemplo) un archivo con todos los registros ONIX deseados. Si su número es muy elevado, se puede dividir esta extracción inicial en varios archivos de menor tamaño.

Una vez obtenida una réplica local inicial de partida, se deben utilizar llamadas **`getRecordStatusX`** de forma periódica para obtener la información de las nuevas altas o de los registros modificados, así como para borrar en local los que se devuelvan en **`deletedRecords`**.

Para asegurar la continuidad en las llamadas **`getRecordStatusX`**, se recomienda dejar el parámetro **`toDate`** vacío, y utilizar como valor del parámetro **`fromDate`** de cada llamada el valor devuelto en el campo **`toDate`** de la respuesta de la llamada anterior.

En los casos en los que se desee gestionar de forma separada los recursos referenciados (imágenes de cubierta, por ejemplo) se deben utilizar llamadas **`getResourceStatusX`** con los mismos criterios indicados para las llamadas **`getRecordStatusX`**.

Estructura de las llamadas

Todas las llamadas tienen la siguiente estructura:

`https://www.dilve.es/dilve/dilve/accion.do?param1=valor1&param2=valor2...`

Donde:

- `accion` es getRecordListX, getRecordStatusX, getRecordsX, getResourceX o getResourceStatusX.
- `param1, param2`… son nombres de parámetros. En el resto de este documento se indican los parámetros permitidos para cada acción
- `valor1, valor2` ... son los valores de los parámetros correspondientes

Parámetros

Tanto para los nombres de acciones como para los nombres de parámetros, las mayúsculas y minúsculas son significativas. Deberán escribirse tal como figuran en este documento.

Existen tres tipos de parámetros:

Obligatorios	Siempre deben figurar, con un valor no vacío.
Dependientes	Su obligatoriedad depende de que figure o no otro parámetro
Opcionales	Pueden figurar o no

Respuestas

Las respuestas a todas las llamadas, excepto **`getResourceX`**, son archivos en XML. Los esquemas correspondientes se presentan en el Apéndice I de este documento.

La codificación de los archivos xml es UTF-8, por defecto, aunque se puede elegir ISO-8859-1 si se envía el parámetro correspondiente.

La tabla siguiente muestra, para cada una de las llamadas, el elemento raíz de la respuesta XML, que coincide con el nombre del esquema XML correspondiente:

Llamada	Raíz de la respuesta / Esquema
getRecordListX	getRecordListXResponse
getRecordStatusX	getRecordStatusXResponse
getRecordsX	getRecordsXResponse
getResourceX	n/a
getResourceStatusX	getResourceStatusXResponse

A continuación se muestra un ejemplo de respuesta correcta a una llamada **`getRecordStatusX`**:

```
<?xml version="1.0" encoding="UTF-8" ?>
<getRecordStatusXResponse
xmlns="http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse" version="1.0">
  <header>
    <fromDate>2007-01-08T00:00:00Z</fromDate>
    <toDate>2009-08-20T00:00:00Z</toDate>
  </header>
  <existingRecords count="189"/>
  <newRecords count="4">
    <record>
      <id type="isbn">9788496224490</id>
      <touchDate>2009-07-16T13:36:59Z</touchDate>
    </record>
    <record>
      <id type="isbn">9788496224322</id>
      <touchDate>2009-07-16T13:36:59Z</touchDate>
    </record>
    <record>
      <id type="isbn">9788496804135</id>
      <touchDate>2009-07-16T13:37:05Z</touchDate>
    </record>
    <record>
      <id type="isbn">9788432312502</id>
      <touchDate>2009-07-09T11:15:11Z</touchDate>
    </record>
  </newRecords>
  <changedRecords count="1">
    <record>
      <id type="isbn">9788482631059</id>
      <touchDate>2009-07-14T14:29:41Z</touchDate>
    </record>
  </changedRecords>
  <deletedRecords count="2959">
    <record>
      <id type="isbn">9788436818239</id>
      <touchDate>2007-10-25T08:17:37Z</touchDate>
    </record>
    <record>
      <id type="isbn">9788436813654</id>
      <touchDate>2007-10-25T08:17:37Z</touchDate>
    </record>
    <record>
      <id type="isbn">9788420725925</id>
      <touchDate>2007-10-25T08:17:05Z</touchDate>
    </record>
    <record>
      <id type="isbn">9788429121049</id>
      <touchDate>2007-10-25T19:47:45Z</touchDate>
    </record>
    ...
  </deletedRecords>
</getRecordStatusXResponse>
```

Compresión

Las respuestas se comprimen con **`gzip`** o con **`deflate`**, si en las cabeceras de la petición http el cliente indica que admite respuestas comprimidas.

Dado que los archivos generados pueden ser de gran tamaño, se recomienda usar la opción de compresión siempre que sea posible.

Errores

Una petición http puede generar dos tipos de error:

Error DAPI Son errores que no interrumpen el proceso normal de la petición pero que impiden la devolución de una respuesta correcta al cliente. Se da, por ejemplo, si se solicita un ISBN que no existe en DILVE. En estos casos se devuelve un xml indicando un código de error y un texto explicativo (salvo en el caso de **`getResourceX`**). Adicionalmente, dependiendo del tipo de error se puede devolver un STATUS HTTP en la cabecera de la respuesta, indicando que se ha producido un error.

Error http Se genera cuando la petición no puede ser procesada correctamente por algún error sintáctico grave, o por problemas en el servidor o en el servicio DILVE. En estos casos se genera un STATUS http en la cabecera de la respuesta que indica la naturaleza del error. También, dependiendo del tipo de error, se puede generar una respuesta XML con un código de error DILVE, aunque toda la información necesaria, en su caso, para procesar el error irá en las cabeceras http.

En cualquier entorno de programa que utilice las llamadas de DAPI es fundamental procesar las cabeceras HTTP de las respuestas, antes de procesar (en su caso) el cuerpo de la respuesta.

Formato de la respuesta de error

En el caso en que se devuelva un mensaje de error, su estructura será semejante a la siguiente:

```
<?xml version="1.0" encoding="UTF-8" ?>
<getRecordStatusXResponse
xmlns="http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse" version="1.0">
 <error>
  <code>1420</code>
  <text> Incorrect datetime format </text>
 </error>
</getRecordStatusXResponse>
```

Códigos de error

Véase el Apéndice II para un listado detallado de los códigos de error.

getRecordListX

Finalidad

Permite obtener un listado (en XML) de los ISBN de los registros existentes en DILVE en un momento dado. También devuelve el total de registros existentes en DILVE. Permite obtener los ISBN con o sin guiones. No se considera la devolución de ISBN-10.

No se especifica ninguna fecha en la llamada. Se supone que se refiere siempre “al momento actual”. En la devolución de resultados, se incluye un timestamp detallado que especifica el momento efectivo en que se ha realizado la consulta.

Los registros no-públicos se ignoran.

Nombre de la llamada

`getRecordListX`

Parámetros

Nombre	Tipo/Valores	Comentario
`user`	**Obligatorio**	Nombre de usuario DILVE
`password`	**Obligatorio**	Password del usuario
`type`	**Opcional**. Valores permitidos: **T** = *Total* (Total) **L** = *List* (Lista) Valor por defecto	Indica si se quiere conocer sólo el número de registros existentes (valor **T)** o se quiere las lista de los ISBN (valor **L**)
`hyphens`	**Opcional** Valores permitidos **N** = *No*. Valor por defecto **Y** = *Yes*	Indica si los ISBN se tienen que devolver con guiones (*hyphens*=*Y*) o sin guiones (*hyphens*=*N*). Esta última es la opción por defecto si no figura el parámetro.
`program`	**Opcional** / Texto Incompatible con el parámetro **`publisher`** y con el parámetro **`country`**.	Indica el nombre de una *programación DILVE* que debe existir para el usuario que realiza la petición. De esta programación, sólo se utilizan los datos de la primera pantalla (selección de registros).
`publisher`	**Opcional** Incompatible con el parámetro **`program`**.	Lista de los identificadores de las editoriales o grupos editoriales de los que se quieren obtener los catálogos completos. Los valores deben separarse mediante la barra vertical, "|".
`publishertype`	**Opcional** Incompatible con el parámetro **`program`**. Valores permitidos: **A** = *Autor/Editor* **E** = *Editorial* / Valor por defecto **ED** = *Editoriales DILVE* **EP** = *Editoriales Plataforma ISBN*[1]	Indica el tipo de editorial de la que se desea consultar los datos.
`country`	**Opcional** Incompatible con el parámetro **`program`**.	Lista de los identificadores de países. Códigos de la lista 91 de ONIX. Los valores deben

[1] Las editoriales Plataforma ISBN únicamente pueden cargar datos básicos. No pueden, por ejemplo, incluir textos ricos ni imágenes de la cubierta.

Nombre	Tipo/Valores	Comentario
		separarse mediante la barra vertical, "|".

Errores

Las tablas del Apéndice 2 detallan los errores que pueden producirse en las llamadas a **`getRecordListX`**.

Gestión del parámetro `program`

Si figura el parámetro **`program`**, este hará mención al nombre de una programación DILVE que debe tener creada el usuario que hace la petición.

En el caso de que figure la referencia a una programación, se utilizarán sus criterios para hacer una primera selección de registros.

De esa programación, sólo se utilizarán los elementos de la primera pantalla: *Selección de criterios y búsqueda*. Es decir, se ignorarán los detalles sobre el formato, forma de envío y periodicidad de la extracción.

Este parámetro no puede coincidir, en la misma llamada, con el parámetro **`publisher`** ni con el parámetro **`country`**.

Gestión del parámetro `publisher`

El parámetro **`publisher`** puede contener uno o varios códigos de editorial, o grupo editorial, participante en DILVE separados mediante barra vertical "|".

Los códigos pueden obtenerse:

1. De las fichas de las editoriales en DILVE bajo el epígrafe "Código interno DILVE".
2. Del listado de participantes (*http://www.dilve.es/dilve/getArchivoDocumentacion.do?iddocumento=521*) en la columna CODIGO.
3. De las fichas ONIX extraídas de DILVE del campo NameCodeValue (con NameCodeType=02 y NameCodeTypeName=DILVE_PUBLID) del bloque Publisher.

Los códigos que se envíen en este parámetro pueden tener el formato DLV0000XXXX, que se utiliza en DILVE, o únicamente el código numérico final.

Gestión del parámetro `country`

El parámetro **`country`** puede contener uno o varios códigos de país (lista 91 de códigos ONIX) separados mediante barra vertical "|".

En el caso de que no figure ni el parámetro **`country`** ni el parámetro **`program`** se devolverán los libros en venta en el mercado español.

Únicamente los usuarios con permisos de selección de productos en venta en otros mercados podrán hacer uso de este parámetro.

Resultados

Introducción

El resultado de una llamada a **`getRecordListX`** es un archivo en XML, que sigue el esquema incluido en el Anexo I.

Básicamente, la estructura global de la respuesta depende de que se haya procesado correctamente o se haya producido un error.

Estas estructuras son:

- Proceso correcto

```
<getRecordListXResponse>
   <header>
   <records>
```

- Proceso con error

```
<getRecordListXResponse>
   <error>
```

Detalle

Etiqueta *getRecordListXResponse*

Incluye los atributos **`xmlns`** y **`version`**. Inicialmente la versión es la 1.0.

- Ejemplo:

```
<getRecordListXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordListXResponse"
version="1.0">
   ...
</getRecordListXResponse>
```

Bloque *header*

Incluye el campo **`listDate`** con el timestamp efectivamente utilizado para hacer el SELECT.

El timestamp estará siempre en formato `YYYY-MM-DDTHH:MM:SSZ`.

- Ejemplo:

```
...
<header>
   <listDate>2007-11-07T00:00:00Z</listDate>
</header>
...
```

Campo *totalRecords*

Incluye el atributo **`count`** que especifica el número de registros existente en el momento *listDate*. Sólo contabiliza los registros públicos. Este campo no tiene contenido. Este campo siempre se incluye en la respuesta. Si *type=T*, es el único campo que se incluye.

- Ejemplo:

```
...
<totalRecords count="97834"/>
...
```

Si figura el parámetro **`program`**, se devuelve sólo el número de registros que cumplan las condiciones correspondientes a la programación especificada.

Bloque *records*

Sólo se devuelve este bloque si **`type`**=*L*, o si no figura ese parámetro. Contiene la lista ISBN de los registros existentes. Incluye un atributo **`count`** con el número de registros (aunque sea una repetición de **`totalRecords`**).

- Ejemplo

```
...
<records count="97834">
   <record>
      <id type="isbn"> 9788496479685</id>
   </record>
   <record>
      <id type="isbn">9788496479692</id>
   </record>
     ...
   <record>
      <id type="isbn">9788496479608</id>
   </record>
</records>
...
```

Si **hyphens**=*Y*, se incluirán guiones en los ISBN.

getRecordStatusX

Finalidad

Permite obtener un listado (en XML) de los ISBN de los registros añadidos, modificados o borrados en DILVE entre dos momentos. También devuelve el total de registros existentes en DILVE al final del intervalo especificado.

En cualquier caso, los registros *no-públicos* se ignoran.

Nombre de la llamada

getRecordStatusX

Parámetros

Nombre	Tipo/Valores	Comentario
user	**Obligatorio**	Nombre de usuario DILVE
password	**Obligatorio**	Password del usuario
fromDate	**Obligatorio** / Timestamp	Indica el momento desde el que se quiere el listado de cambios. Se interpreta siempre de acuerdo con la hora del servidor de BB.DD de DILVE.
toDate	**Opcional** / Timestamp Si no figura, DILVE establece el valor de este parámetro.	Indica el momento hasta el que se quiere el listado de cambios. Se interpreta siempre de acuerdo con la hora del servidor de BB.DD de DILVE.
type	**Opcional**. Valores permitidos: **N** = *New* (Nuevos) **C** = *Changed* (Cambiados)	Indica el tipo de cambios que se quieren monitorizar. El valor **T** indica que sólo interesa conocer el número total

Nombre	Tipo/Valores	Comentario
	D = *Deleted* (Borrados) **T** = *Total* (Total) **A** = *All* (Todos) Valor por defecto	de registros existentes en *toDate*.
`detail`	**Opcional** Valores permitidos **N** = *Normal*. Valor por defecto **D** = *Date* (Fecha) **S** = *Summary* (Resumen)	*Normal* devuelve un listado con los ISBN afectados, en función del valor del parámetro **`type`**. *Date* incluye, para cada ISBN el timestamp correspondiente al alta, modificación o borrado. *Summary* sólo devuelve el número de ISBN afectados.
`hyphens`	**Opcional** Valores permitidos **N** = *No*. Valor por defecto **Y** = *Yes*	Indica si los ISBN se tienen que devolver con guiones (*hyphens*=*Y*) o sin guiones (*hyphens*=*N*). Esta última es la opción por defecto si no figura el parámetro.
`program`	**Opcional** / Texto Incompatible con el parámetro **`publisher`** y con el parámetro **`country`**.	Indica el nombre de una *programación DILVE* que debe existir para el usuario que realiza la petición. De esta programación, sólo se utilizan los datos de la primera pantalla (selección de registros).
`publisher`	**Opcional** Incompatible con el parámetro **`program`**.	Lista de los identificadores de las editoriales o grupos editoriales de los que se quieren obtener los catálogos completos. Los valores deben separarse mediante la barra vertical, "|".
`publishertype`	**Opcional** Incompatible con el parámetro **`program`**. Valores permitidos: **A** = *Autor/Editor* **E** = *Editorial* / Valor por defecto **ED** = *Editoriales DILVE* **EP** = *Editoriales Plataforma ISBN*[2]	Tipo de editorial de la que se desea consultar los datos.
`country`	**Opcional** Incompatible con el parámetro **`program`**.	Lista de los identificadores de países. Códigos de la lista 91 de ONIX. Los valores deben separarse mediante la barra vertical, "|".

[2] Las editoriales Plataforma ISBN únicamente pueden cargar datos básicos. No pueden, por ejemplo, incluir textos ricos ni imágenes de la cubierta.

Errores

Las tablas del Apéndice 2 detallan los errores que pueden producirse en la llamada a **`getRecordStatusX`**.

Gestión del parámetro `program`

Si figura el parámetro **`program`**, este hará mención al nombre de una programación DILVE que debe tener creada el usuario que hace la petición.

En el caso de que figure la referencia a una programación, se utilizarán sus criterios para hacer una primera selección de registros.

De esa programación, sólo se utilizarán los elementos de la primera pantalla: *Selección de criterios y búsqueda*. Es decir, se ignorarán los detalles sobre el formato, forma de envío y periodicidad de la extracción.

Hay que tener en cuenta los siguientes detalles:

- El número de registros **`existingRecords`** se referirá al número de registros que cumplen los criterios establecidos en la programación, en el momento de hacer la consulta (**`toDate`**).
- Dado que no se pueden aplicar filtros (p.ej. editorial, materia, etc.) sobre los registros borrados, excepto el de la fecha del borrado, se devolverán los ISBN de todos los registros borrados en el intervalo de fechas determinado entre **`fromDate`** y **`toDate`**.

Este parámetro no puede coincidir, en la misma llamada, con el parámetro **`publisher`** ni con el parámetro **`country`**.

Gestión del parámetro `publisher`

El parámetro **`publisher`** puede contener uno o varios códigos de editorial, o grupo editorial, participante en DILVE separados mediante barra vertical “|”.

Los códigos pueden obtenerse:

1. De las fichas de las editoriales en DILVE bajo el epígrafe "Código interno DILVE".
2. Del listado de participantes (*http://www.dilve.es/dilve/getArchivoDocumentacion.do?iddocumento=521*) en la columna CODIGO.
3. De las fichas ONIX extraídas de DILVE del campo NameCodeValue (con NameCodeType=02 y NameCodeTypeName=DILVE_PUBLID) del bloque Publisher.

Los códigos que se envíen en este parámetro pueden tener el formato DLV0000XXXX, que se utiliza en DILVE, o únicamente el código numérico final.

Gestión del parámetro `country`

El parámetro **`country`** puede contener uno o varios códigos de país (lista 91 de códigos ONIX) separados mediante barra vertical “|”.

En el caso de que no figure ni el parámetro **`country`** ni el parámetro **`program`** se devolverán los libros en venta en el mercado español.

Únicamente los usuarios con permisos de selección de productos en venta en otros mercados podrán hacer uso de este parámetro.

Timestamps

Los timestamps se toman siempre referidos a la hora del sistema DILVE, es decir, con la misma referencia con la que se generan los timestamps del log de modificaciones de los registros.

En la respuesta de **`getRecordStatusX`** se incluye el timestamp de finalización del periodo de consulta, establecido por DILVE. El usuario debe utilizar este timestamp para el parámetro **`fromDate`** de la siguiente llamada.

Los timestamps deben tener la siguiente estructura:

```
YYYY-MM-DD
```

o

```
YYYY-MM-DDTHH:MM:SSZ
```

No se admiten otros formatos.

Si no figura HH:MM:SS, se asume 00:00:00 (medianoche, inicio del día correspondiente a la fecha indicada) para el parámetro fromDate y 23:59:59 para toDate.

No se admiten fechas anteriores a 2009. Para la recuperación de los libros dados de alta en DILVE debe utilizarse la llamada getRecordListX.

Resultados

Introducción

El resultado de una llamada a **`getRecordStatusX`** es un archivo en XML, que sigue el esquema incluido en el Apéndice I.

Básicamente, la estructura global de la respuesta depende de que se haya procesado correctamente o se haya producido un error.

Estas estructuras son:

- <u>Proceso correcto</u>

```
<getRecordStatusXResponse>
   <header>
   <existingRecords>
   <newRecords>
   <changedRecords>
   <deletedRecords>
```

- <u>Proceso con error</u>

```
<getRecordStatusXResponse>
   <error>
```

Detalle

Etiqueta *getRecordStatusXResponse*

Incluye los atributos **`xmlns`** y **`version`**. Inicialmente la versión es la 1.0.

- Ejemplo:

```
<getRecordStatusXResponse
xmlns="http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse" version="1.0">
   ...
</getRecordStatusXResponse>
```

Bloque *header*

Incluye los campos **fromDate** y **toDate** con los timestamps de inicio y final del periodo considerado. Ambos valores se incluyen siempre. Si el usuario no ha especificado el campo **toDate**, el sistema incluye el timestamp efectivamente utilizado para hacer el SELECT.

Los timestamp está siempre en formato `YYYY-MM-DDTHH:MM:SSZ`, independientemente de que en la llamada se haya detallado la hora o no.

- Ejemplo:

```
...
<header>
   <fromDate>2007-10-01T00:00:00Z</fromDate>
   <toDate>2007-10-02T01:17:54Z </toDate>
</header>
...
```

Etiqueta *existingRecords*

Incluye el atributo **count** que especifica el número de registros existente en el momento **toDate**. Sólo contabiliza los registros públicos. Este campo no tiene contenido. Este campo siempre se incluye en la respuesta. Si **type**=*T*, es el único campo que se incluye.

- Ejemplo:

```
...
<existingRecords count="97834"/>
...
```

Si figura el parámetro **program**, se devuelve solo el número de registros que cumplan las condiciones correspondientes a la programación especificada.

Bloques *newRecords*, *changedRecords*, *deletedRecords*

Los tres tienen la misma estructura. Esta dependerá del parámetro **detail**. Cada campo incluirá un atributo **count** con el número de registros en la situación correspondiente.

Adicionalmente, si **detail** = *N o D*, se incluye una lista con el detalle de la identificación de los registros afectados. Y, en el caso **detail** = *D*, se añade un campo con el timestamp de la última operación que haya afectado al registro.

- Ejemplos

1) *detail* = *N* (normal)

En este caso, cada bloque record, correspondiente a un registro, sólo incluye el elemento id. Este tiene un atributo type, que indica el tipo de identificador. Por ahora siempre se indicará **type=isbn"** (es el ISBN13 o EAN del libro)

```
...
<newRecords count="3">
   <record>
      <id type="isbn"> 9788496479685</id>
   </record>
   <record>
      <id type="isbn">9788496479692</id>
   </record>
   <record>
      <id type="isbn">9788496479608</id>
   </record>
</newRecords>
...
```

2) *detail* = *D* (date)

En este caso, para cada registro, se indica el timestamp de la última operación realizada sobre el mismo. Este dato se incluye como elemento aparte y no como atributo del elemento **record**, para

simplificar, en el futuro, la gestión de bloques record más complejos: con más fechas, más metadatos sobre el libro, etc.

```
...
<newRecords count="3">
   <record>
      <id type="isbn">9788496479685</id>
      <touchDate>2007-10-01T06:34:32Z</touchDate>
   </record>
   <record>
      <id type="isbn">9788496479692</id>
      <touchDate>2007-10-01T06:34:57Z</touchDate>
   </record>
   <record>
      <id type="isbn">9788496479608</id>
      <touchDate>2007-10-01T13:51:10Z</touchDate>
   </record>
</newRecords>
...
```

3) *detail* = *S* (summary)

En este caso solo se indica el número de registros afectados en cada categoría;.

```
...
<newRecords count="3" />
...
```

Bloque *error*

Incluye dos elementos: **`code`** y **`text`**. El primero indica el valor numérico del código de error y el segundo el texto correspondiente.

- Ejemplo

```
<getRecordStatusXResponse>
   <error>
      <code>1420</code>
      <text>Formato de fecha incorrecto: fromDate</text>
   </error>
</getRecordStatusXResponse>
```

getRecordsX

Finalidad

Permite obtener fichas de libros existentes en DILVE.

Las fichas se pueden obtener en distintos formatos. En septiembre de 2024 están implementados los siguientes:

- ONIX 3.1, con marcas cortas o largas
- ONIX 3.0, con marcas cortas o largas
- ONIX 2.1, con marcas cortas o largas
- CEGAL en Red
- CSV

En el caso CEGAL en Red se pueden obtener dos subformatos: ficha básica y ficha completa.

El formato CEGAL ficha completa permite obtener la imagen de cubierta. En caso de tratarse de libro electrónico sin soporte físico especificado (ProductForm EA, EB, EC o ED) no se extraerán los datos.

El formato CSV debe incluir la referencia a un mapa de campos creado previamente en la plataforma.

El detalle de estos formatos se expone más adelante.

Nombre de la llamada

`getRecordsX`

Parámetros

Nombre	Tipo/Valores	Comentario
`user`	**Obligatorio**	Nombre de usuario DILVE
`password`	**Obligatorio**	Password del usuario
`identifier`	**Obligatorio**	Lista de los identificadores de los registros que se quieren obtener. Se admiten ISBN13 e ISBN10 (con o sin guiones). No se comprueba la posición de los guiones. Se pueden introducir hasta 128 identificadores en una misma petición, separando los valores correspondientes mediante la barra vertical, "|".
`metadataformat`	**Opcional** *Valores permitidos*: **ONIX** = ONIX, marcas largas Valor por defecto **ONIX_ST** = ONIX, marcas cortas **CEGAL** = Formato CEGAL en red **CSV** = formato CSV (mapa de campos)	Formato de la respuesta. Ver el detalle de las distintas opciones más adelante.
`version`	**Opcional** Para **CEGAL:** **3:** Formato SINLI v3 Valor por defecto para CEGAL **4:** Formato SINLI v4 **5:** Formato SINLI v5 **6:** Formato SINLI v6 **7:** Formato SINLI v7 **8:** Formato SINLI v8 **9:** Formato SINLI v9 Para **ONIX:** **2.1:** Para ONIX v2.1 Valor por defecto para ONIX **3.0:** Para ONIX v3.0	Indica la versión concreta (P.ej. ONIX 2.1, 3.0 o 3.1) del formato de la respuesta

Nombre	Tipo/Valores	Comentario
	3.1: Para ONIX v3.1	
formatdetail	**Opcional** Si metadataformat=**CEGAL** se permiten los valores permitidos: **B** = Básica Valor por defecto **C** = Completa Si metadataformat=**CSV** se permite cualquier valor de mapa CSV creado en DILVE:	Se ignora en las respuestas en formato ONIX.
encoding	**Opcional** Valor por defecto: **UTF-8** si el formato es **ONIX**. Valor por defecto **ISO-8859-1** si el formato es **CEGAL** o **CSV**.	Indica el tipo de encoding para la respuesta XML en formato ONIX. Si no figura o si figura en blanco, se toma el valor por defecto.

Errores

Las tablas del Apéndice 2 detallan los errores que pueden producirse en las llamadas a **getRecordsX**.

Resultados

Introducción

El resultado de una llamada a **getRecordsX** es un archivo en XML, que sigue el esquema incluido en el Anexo I.

Básicamente, la estructura global de la respuesta depende de que se haya procesado correctamente o se haya producido un error.

Estas estructuras son:

- Proceso correcto

```
<getRecordsXResponse>
   <cuerpo de la respuesta>
```
(depende del formato, ver más adelante)

- Proceso con error

```
<getRecordsXResponse>
   <error>
```

Detalle

Etiqueta *getRecordsXResponse*

Incluye los atributos **xmlns** y **version**. Actualmente la versión es la 1.2.

- Ejemplo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   …
</getRecordsXResponse>
```

Bloque de contenido

La información solicitada vendrá dentro de un bloque cuyo nombre depende del formato especificado en el parámetro metadataformat. La etiqueta del bloque incluye un atributo xmlns que depende, también, del formato solicitado.

En el caso de formato ONIX si alguno de los ISBN solicitados no existe en DILVE, se omite el elemento <Product> (o <product>) correspondiente en el mensaje ONIX.

Dado que la especificación del formato del mensaje ONIX no prevé la inclusión de información binaria, no es posible incluir directamente las imágenes de cubierta u otros archivos de recursos directamente con la petición getRecordsX en ONIX. Existe una petición específica, getResourceX para este fin.

A continuación se detalla cada uno de estos bloques.

Formato ONIX 2.1 etiquetas largas

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <ONIXMessage xmls="http://www.editeur.org/onix/2.1/reference">
   ...
   </ONIXMessage>
</getRecordsXResponse>
```

El bloque <ONIXMessage> incluye un mensaje ONIX for Books válido completo (etiquetas largas), de acuerdo con el esquema especificado *en http://www.editeur.org/onix/2.1/ONIX_BookProduct_Release2.1_reference.*xsd

Formato ONIX 2.1 etiquetas cortas

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <ONIXmessage xmls="http://www.editeur.org/onix/2.1/short">
   ...
   </ONIXmessage>
</getRecordsXResponse>
```

El bloque <ONIXmessage> incluye un mensaje ONIX for Books válido completo (etiquetas cortas), de acuerdo con el esquema especificado *en http://www.editeur.org/onix/2.1/ONIX_BookProduct_Release2.1_short.*xsd

Formato ONIX 3.0 etiquetas largas

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <ONIXMessage xmls="http://ns.editeur.org/onix/3.0/reference">
   ...
   </ONIXMessage>
</getRecordsXResponse>
```

El bloque <ONIXMessage> incluye un mensaje ONIX for Books válido completo (etiquetas largas), de acuerdo con el esquema ONIX_BookProduct_3.0_reference.xsd

Formato ONIX 3.0 etiquetas cortas

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <ONIXmessage xmls="http://ns.editeur.org/onix/3.0/short">
   ...
   </ONIXmessage>
</getRecordsXResponse>
```

El bloque <ONIXmessage> incluye un mensaje ONIX for Books válido completo (etiquetas cortas), de acuerdo con el esquema ONIX_BookProduct_3.0_short.xsd

Formato ONIX 3.1 etiquetas largas

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <ONIXMessage xmls="http://ns.editeur.org/onix/3.1/reference">
   ...
   </ONIXMessage>
</getRecordsXResponse>
```

El bloque <ONIXMessage> incluye un mensaje ONIX for Books válido completo (etiquetas largas), de acuerdo con el esquema ONIX_BookProduct_3.1_reference.xsd

Formato ONIX 3.1 etiquetas cortas

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <ONIXmessage xmls="http://ns.editeur.org/onix/3.1/short">
   ...
   </ONIXmessage>
</getRecordsXResponse>
```

El bloque <ONIXmessage> incluye un mensaje ONIX for Books válido completo (etiquetas cortas), de acuerdo con el esquema ONIX_BookProduct_3.1_short.xsd

Formato CEGAL

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <FICHALIBROS xmlns="http://www.dilve.es/api/xsd/fichaLibros">
   ...
   </ FICHALIBROS >
</getRecordsXResponse>
```

DILVE implementa el formato CEGAL de esta llamada de acuerdo con las especificaciones del documento "ENVIOS_SINLI_HACIA_CEGAL_EN_RED.pdf " que puede descargarse del web de CEGAL *en http://www.cegal.es/lib/informes/listaInformes.php?codTipoInformeAso*=56.

El esquema exacto para la construcción del bloque FICHALIBROS es una adaptación del original (www.cegal.es/libros/esquemas/fichalibro.xsd) y que se detalla en el Apéndice I de este documento.

Formato CSV

La respuesta será del tipo:

```
<getRecordsXResponse xmlns="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
version="1.2">
   <CSV-DATA>
   ...
   </CSV-DATA>
</getRecordsXResponse>
```

Dentro del campo CSV-DATA se incluirá en primer lugar la cabecera con los nombre de los campos y en las líneas posteriores los datos de todos los registros solicitados.

getResourceX

Finalidad

Permite descargar de DILVE un archivo de recursos: imagen de cubierta, extracto..., asociado a un ISBN, y del cual se conoce el nombre.

Nombre de la llamada

`getResourceX`

Parámetros

Nombre	Tipo/Valores	Comentario
`user`	**Obligatorio**	Nombre de usuario DILVE
`password`	**Obligatorio**	Password del usuario
`identifier`	**Obligatorio**	Identificador del registro que tiene asociado el recurso que se quiere obtener. Se admiten ISBN13 e ISBN10 (con o sin guiones). No se comprueba la posición de los guiones.
`resource`	**Obligatorio**	Nombre del archivo que se quiere descargar. Debe incluir la extensión
`lastupdate`	**Opcional** / Timestamp	Fecha de última modificación del recurso. Si figura el recurso se extraerá sólo si su fecha de última modificación es igual o posterior a la indicada.

Errores

Las tablas del Apéndice II detallan los errores que pueden producirse en las llamadas a `getResourceX`.

Resultados

El resultado de `getResourceX` es el único de la DAPI (DILVE - API) que no está en formato XML. Será, directamente, el archivo solicitado, en el formato que este tenga: jpg, pdf, etc.

La respuesta de `getResourceX`, por lo tanto, será siempre o bien el archivo solicitado, o bien una respuesta de error a nivel de cabecera HTTP, según se detalla en la tabla "Mapeos de errores" del Apéndice II.

getResourceStatusX

Finalidad

Permite obtener un listado (en XML) de los ISBN de los registros en los que hayan modificado alguno de los recursos referenciados (habitualmente y por defecto, la imagen de cubierta). Como criterio (por defecto) se utilizará la fecha de modificación del recurso indicada por la editorial. Para los casos en los que la editorial no haya detallado esta información, existe la posibilidad de utilizar la fecha de modificación del producto en su conjunto.

Los registros *no-públicos* se ignoran.

Nombre de la llamada

`getResourceStatusX`

Parámetros

Nombre	Tipo/Valores	Comentario
`user`	**Obligatorio**	Nombre de usuario DILVE
`password`	**Obligatorio**	Password del usuario
`fromDate`	**Obligatorio** / Timestamp	Indica el momento desde el que se quiere el listado de cambios. Se interpreta siempre de acuerdo con la hora del servidor de BB.DD de DILVE.
`toDate`	**Opcional** / Timestamp Si no figura, DILVE establece el valor de este parámetro.	Indica el momento hasta el que se quiere el listado de cambios. Se interpreta siempre de acuerdo con la hora del servidor de BB.DD de DILVE.
`detail`	**Opcional** Valores permitidos **N** = *Normal*. Valor por defecto **D** = *Date* (Fecha)	*Normal* devuelve un listado con los ISBN afectados. *Date* incluye, para cada ISBN el timestamp correspondiente a la modificación. Esta fecha puede corresponder a la fecha de modificación de recurso (`resourceTouchDate`) o a la fecha de modificación del producto (`recordTouchDate`).
`hyphens`	**Opcional** Valores permitidos **N** = *No*. Valor por defecto **Y** = *Yes*	Indica si los ISBN se tienen que devolver con guiones (*hyphens*=*Y*) o sin guiones (*hyphens*=*N*). Esta última es la opción por defecto si no figura el parámetro.
`onixtype`	**Opcional** Valor de la lista 158 de códigos ONIX (01 valor por defecto)	Indica el tipo de recurso a consultar. Valor por defecto 01 (Cubierta). Se valida que corresponda a un valor de la lista 158 de códigos ONIX.
`fallback`	**Opcional** **N** = *No*. Valor por defecto **Y** = *Yes*	Indica si debe considerarse la fecha de modificación del registro completo en caso de que el recurso solicitado no la tenga informada. **N**, valor por defecto, indica que sólo se tendrán en cuenta los registros cuyos recursos, del tipo indicado, tengan fecha de modificación explícita. El resto de registros se ignora. **Y**, indica que, en el caso de que

Nombre	Tipo/Valores	Comentario
		el recurso no tenga fecha explícita, se considerará la fecha de modificación del registro completo.

Errores

Las tablas del Apéndice 2 detallan los errores que pueden producirse en la llamada a **`getResourceStatusX`**.

Timestamps

Los timestamps se toman siempre referidos a la hora del sistema DILVE, es decir, con la misma referencia con la que se generan los timestamps del log de modificaciones de los registros.

En la respuesta de **`getResourceStatusX`** se incluye el timestamp de finalización del periodo de consulta, establecido por DILVE. El usuario debe utilizar este timestamp para el parámetro **`fromDate`** de la siguiente llamada.

Los timestamps deben tener la siguiente estructura:

```
YYYY-MM-DD
```

o

```
YYYY-MM-DDTHH:MM:SSZ
```

No se admiten otros formatos.

Si no figura HH:MM:SS, se asume 00:00:00 (medianoche, inicio del día correspondiente a la fecha indicada) para el parámetro fromDate y 23:59:59 para toDate.

Resultados

Introducción

El resultado de una llamada a **`getResourceStatusX`** es un archivo en XML, que sigue el esquema incluido en el Apéndice I.

Básicamente, la estructura global de la respuesta depende de que se haya procesado correctamente o se haya producido un error.

Estas estructuras son:

- <u>Proceso correcto</u>

```
<getResourceStatusXResponse>
   <header>
   <existingRecords>
   <Records>
```

- <u>Proceso con error</u>

```
<getResourceStatusXResponse>
   <error>
```

Detalle

Etiqueta *getResourceStatusXResponse*

Incluye los atributos **xmlns** y **version**. Inicialmente la versión es la 1.0.

- Ejemplo:

```
<getResourceStatusXResponse
xmlns="http://www.dilve.es/dilve/api/xsd/getResourceStatusXResponse" version="1.0">
   ...
</getResourceStatusXResponse>
```

Bloque *header*

Incluye los campos **fromDate** y **toDate** con los timestamps de inicio y final del periodo considerado. Ambos valores se incluyen siempre. Si el usuario no ha especificado el campo **toDate**, el sistema incluye el timestamp efectivamente utilizado para hacer el SELECT.

Los timestamp está siempre en formato `YYYY-MM-DDTHH:MM:SSZ`, independientemente de que en la llamada se haya detallado la hora o no.

- Ejemplo:

```
...
<header>
   <fromDate>2007-10-01T00:00:00Z</fromDate>
   <toDate>2007-10-02T01:17:54Z </toDate>
</header>
...
```

Etiqueta *existingRecords*

Incluye el atributo **count** que especifica el número de registros existente en el momento **toDate**. Sólo contabiliza los registros públicos. Este campo no tiene contenido. Este campo siempre se incluye en la respuesta.

- Ejemplo:

```
...
<existingRecords count="97834"/>
...
```

Bloque *Records*

Su estructura dependerá del parámetro **detail**. Incluirá un atributo **count** con el número de registros en la situación correspondiente.

Adicionalmente, si **detail** = *N o D*, se incluye una lista con el detalle de la identificación de los registros afectados. Y, en el caso **detail** = *D*, se añade un campo con el timestamp de la última operación que haya afectado al recurso o al registro.

- Ejemplos

  1) *detail* = *N* (normal)

En este caso, cada bloque record, correspondiente a un registro, sólo incluye el elemento id. Este tiene un atributo type, que indica el tipo de identificador. Por ahora siempre se indicará **type=isbn"** (es el ISBN13 o EAN del libro)

```
...
<Records count="3">
   <record>
      <id type="isbn"> 9788496479685</id>
   </record>
   <record>
      <id type="isbn">9788496479692</id>
   </record>
   <record>
```

```
      <id type="isbn">9788496479608</id>
   </record>
</Records>
...
```

2) *detail* = *D* (date)

En este caso, para cada registro, se indica el timestamp de la última operación realizada sobre el mismo. En caso de que se trate del timestamp de la última modificación del recurso se usará la etiqueta `<resourceTouchDate>` y si se trata de la modificación de la ficha del libro la etiqueta `<recordTouchDate>`. Este dato se incluye como elemento aparte y no como atributo del elemento **`record`**, para simplificar, en el futuro, la gestión de bloques record más complejos: con más fechas, más metadatos sobre el libro, etc.

En el ejemplo, se supone que se ha usado el parámetro **`fallback`**=Y, por lo que aparecen mezcladas fechas de modificación de recursos, cuando figuran, y de modificación de registros, en caso contrario.

```
...
<Records count="3">
   <record>
      <id type="isbn">9788496479685</id>
      <recordTouchDate>2007-10-01T06:34:32Z</recordTouchDate>
   </record>
   <record>
      <id type="isbn">9788496479692</id>
      <resourceTouchDate>2007-10-01T00:00:00Z</resourceTouchDate>
   </record>
   <record>
      <id type="isbn">9788496479608</id>
      <resourceTouchDate>2007-10-01T00:00:00Z</resourceTouchDate>
   </record>
</Records>
...
```

Bloque *error*

Incluye dos elementos: **`code`** y **`text`**. El primero indica el valor numérico del código de error y el segundo el texto correspondiente.

- Ejemplo

```
<getResourceStatusXResponse>
   <error>
      <code>1420</code>
      <text>Formato de fecha incorrecto: fromDate</text>
   </error>
</getResourceStatusXResponse>
```

Apéndice I: Esquemas XML de las respuestas

En este Anexo se incluyen los esquemas XML de las respuestas de las distintas llamadas de la DAPI (API de DILVE).

Igualmente se incluye el esquema **`fichaLibro.xsd`** específico para el contenido de las respuestas de **`getRecordsX`** en formato CEGAL.

Estos esquemas están disponibles en la web de DILVE.

getRecordListX

Disponible e*n:*

http://www.dilve.es/dilve/api/xsd/getRecordListXResponse/1.0/getRecordListXResponse.xsd

y e*n,*

http://www.dilve.es/dilve/api/xsd/getRecordListXResponse/getRecordListXResponse.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DILVE API: getRecordListX response schema -->
<!-- (c)2009 DILVE - www.dilve.es - -->
<!-- Revision history -->
<!-- 1.0 - 2009/10/01: Initial version -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
xmlns:g="http://www.dilve.es/dilve/api/xsd/getRecordListXResponse"
targetNamespace="http://www.dilve.es/dilve/api/xsd/getRecordListXResponse"
version="1.0">

   <xsd:element name="getRecordListXResponse">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element minOccurs="0" ref="g:header"/>
            <xsd:element minOccurs="0" ref="g:totalRecords"/>
            <xsd:element minOccurs="0" ref="g:records"/>
            <xsd:element minOccurs="0" ref="g:error"/>
         </xsd:sequence>
         <xsd:attribute name="version" use="required" type="xsd:string"
fixed="1.0"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="header">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:listDate"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="totalRecords">
      <xsd:complexType>
         <xsd:attribute name="count" use="required" type="xsd:integer"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="records">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" ref="g:record"/>
         </xsd:sequence>
         <xsd:attribute name="count" use="required" type="xsd:integer"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="record">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:id"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="id">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="type" use="required" type="xsd:string"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="error">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:code"/>
            <xsd:element ref="g:text"/>
         </xsd:sequence>
```

```
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="code" type="xsd:string"/>
   <xsd:element name="text" type="xsd:string"/>

   <xsd:element name="listDate" type="xsd:string"/>

</xsd:schema>
```

getRecordStatusX

Disponible en:

http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse/1.0/getRecordStatusXResponse.xsd

y en,

http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse/getRecordStatusXResponse.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DILVE API: getRecordStatusX response schema -->
<!-- (c)2009 DILVE - www.dilve.es - -->
<!-- Revision history -->
<!-- 1.0 - 2009/10/01: Initial version -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse"
xmlns:g="http://www.dilve.es/dilve/api/xsd/getRecordStatusXResponse"
version="1.0">

   <xsd:element name="getRecordStatusXResponse">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element minOccurs="0" ref="g:header"/>
            <xsd:element minOccurs="0" ref="g:existingRecords"/>
            <xsd:element minOccurs="0" ref="g:newRecords"/>
            <xsd:element minOccurs="0" ref="g:changedRecords"/>
            <xsd:element minOccurs="0" ref="g:deletedRecords"/>
            <xsd:element minOccurs="0" ref="g:error"/>
         </xsd:sequence>
         <xsd:attribute name="version" use="required" type="xsd:string"
fixed="1.0"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="header">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:fromDate"/>
            <xsd:element ref="g:toDate"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="existingRecords">
      <xsd:complexType>
         <xsd:attribute name="count" use="required" type="xsd:integer"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="newRecords" type="g:recordsType"/>
   <xsd:element name="changedRecords" type="g:recordsType"/>
   <xsd:element name="deletedRecords" type="g:recordsType"/>

   <xsd:complexType name="recordsType">
      <xsd:sequence>
         <xsd:element minOccurs="0" maxOccurs="unbounded" ref="g:record"/>
      </xsd:sequence>
      <xsd:attribute name="count" use="required" type="xsd:integer"/>
   </xsd:complexType>
```

```
   <xsd:element name="record">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:id"/>
            <xsd:element minOccurs="0" ref="g:touchDate"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="id">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="type" use="required" type="xsd:string"/>
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="error">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:code"/>
            <xsd:element ref="g:text"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="code" type="xsd:string"/>
   <xsd:element name="text" type="xsd:string"/>

   <xsd:element name="touchDate" type="xsd:string"/>
   <xsd:element name="fromDate" type="xsd:string"/>
   <xsd:element name="toDate" type="xsd:string"/>

</xsd:schema>
```

getRecordsX

Disponible en:

http://www.dilve.es/dilve/api/xsd/getRecordsXResponse/1.0/getRecordsXResponse.xsd

y en,

http://www.dilve.es/dilve/api/xsd/getRecordsXResponse/getRecordsXResponse.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<!-- DILVE API: getRecordsX response schema -->
<!-- (c)2009 DILVE - www.dilve.es - -->
<!-- Revision history -->
<!-- 1.0 - 2009/10/01: Initial version -->
<!-- 1.1 - 2010/12/20: ONIX 3.0 (reference and short -->
<!-- 1.2 - 2012/06/21: change namespace ONIX 3.0 -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
    xmlns:g="http://www.dilve.es/dilve/api/xsd/getRecordsXResponse"
    xmlns:onix_short="http://www.editeur.org/onix/2.1/short"
    xmlns:onix_ref="http://www.editeur.org/onix/2.1/reference"
    xmlns:onix3.0_short="http://ns.editeur.org/onix/3.0/short"
    xmlns:onix3.0_ref="http://ns.editeur.org/onix/3.0/reference"
    xmlns:fichaLibros="http://www.dilve.es/dilve/api/xsd/fichaLibros"
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
      version="1.2">

   <!-- **imports**: Change locations of imported schemas as needed -->
   <xsd:import namespace="http://www.editeur.org/onix/2.1/short"
schemaLocation="C:\dilve\onix2.1\ONIX_BookProduct_Release2.1_short.xsd"/>
   <xsd:import namespace="http://www.editeur.org/onix/2.1/reference"
schemaLocation="C:\dilve\onix2.1\ONIX_BookProduct_Release2.1_reference.xsd"/>
   <xsd:import namespace="http://ns.editeur.org/onix/3.0/short"
```

```
schemaLocation="C:\dilve\onix3.0\ONIX_BookProduct_3.0_short.xsd"/>
   <xsd:import namespace="http://ns.editeur.org/onix/3.0/reference"
schemaLocation="C:\dilve\onix3.0\ONIX_BookProduct_3.0_reference.xsd"/>
   <xsd:import namespace="http://www.dilve.es/dilve/api/xsd/fichaLibros"
schemaLocation="http://www.dilve.es/dilve/api/xsd/fichaLibros/1.0/fichaLibros.xs
d"/>
   <xsd:import namespace="http://www.openarchives.org/OAI/2.0/oai_dc/"
schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc.xsd"/>
   <!-- imports end-->

   <xsd:element name="getRecordsXResponse">
      <xsd:complexType>
            <xsd:sequence>
            <xsd:choice minOccurs="0">
                     <xsd:element ref="onix_short:ONIXmessage" minOccurs="1"
maxOccurs="1"/>
                     <xsd:element ref="onix_ref:ONIXMessage" minOccurs="1"
maxOccurs="1"/>
                     <xsd:element ref="onix3.0_short:ONIXmessage" minOccurs="1"
maxOccurs="1"/>
                     <xsd:element ref="onix3.0_ref:ONIXMessage" minOccurs="1"
maxOccurs="1"/>
                     <xsd:element ref="fichaLibros:FICHALIBROS" minOccurs="1"
maxOccurs="1"/>
                     <xsd:element ref="oai_dc:dc" minOccurs="1"
maxOccurs="unbounded"/>
                     <xsd:element ref="g:data" minOccurs="1" maxOccurs="1"/>
            </xsd:choice>
            <xsd:element ref="g:error" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
         <xsd:attribute name="version" use="required" type="xsd:string"
fixed="1.2"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="error">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:code" minOccurs="1" />
            <xsd:element ref="g:text" minOccurs="1" />
            <xsd:element ref="g:identifier" minOccurs="0"
maxOccurs="unbounded"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="code" type="xsd:string"/>
   <xsd:element name="identifier" type="xsd:string"/>
   <xsd:element name="text" type="xsd:string"/>
   <xsd:element name="data" type="xsd:string"/>

</xsd:schema>
```

fichaLibros

Disponible en:

http://www.dilve.es/dilve/api/xsd/fichaLibros/1.0/fichaLibros.xsd

y en,

http://www.dilve.es/dilve/api/xsd/fichaLibros/fichaLibros.xsd

```
<?xml version="3.0" encodiog="utf-8"?>--- DILVE API: fichalibro response schema -->---
Adapted from CEGAL                    -->--- (c)2009 DIL-E - www.dilve.es -        -->---
Revision history                      -->--- 1-0 - 2009/10/01: Initial version     -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
targetNamespace="http://www.dilve.es/dilve/api/xsd/fichaLibros"
attributeFormDefault="unqualified" version="3.0">

    <xsd:element name="FICHALIBROS">
```

```
        <xsd:complexType>
            <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="LI”RO" maxOccurs="unbounDed">

                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="ISBN" type="xsd:string"/>
                            <xsd:element name="VERSION_SINLI" type="xsd:string"/>
                            <xsd:element name="DETALLE" type="xsd:string"/>
                            <xsd:element name="RESUMEN" type="xsd:string" minOccurs="0"/>
                            <xsd:element name="INDICE" type="xsd:string" minOccurs="0"/>
                            <xsd:element name="PORTADA" minOccurs="0">
                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="IMAGEN_PORTADA"
type="xsd:string"/>
                                        <xsd:element name="FORMATO_PORTADA"
type="xsd:string"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                            <xsd:element name="AUTORES" minOccurs="0">
                                <xsd:complexType>
                                    <xsd:sequence>

                                        <xsd:element name="AUTOR" minOccurs="0"
maxOccurs="unbounded">
                                            <xsd:complexType>
                                                <xsd:sequence>
                                                    <xsd:element name="NOMBRE"
type="xsd:string"/>
                                                    <xsd:element name="BIOGRAFIA"
type="xsd:string" minOccurs="0"/>
                                                </xsd:sequence>
                                            </xsd:complexType>
                                        </xsd:element>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="ERRORES" minOccurs="0">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="ERROR" maxOccurs="unbounded">

                                <xsd:complexType>
                                    <xsd:sequence>
                                        <xsd:element name="ISBN" type="xsd:string"
minOccurs="0"/>
                                        <xsd:element name="CODIGO" type="xsd:integer"/>
                                        <xsd:element name="DESCRIPCION" type="xsd:string"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>

                    </xsd:complexType>
                </xsd:element>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

getResourceX

La llamada getResourceX devuelve directamente el archivo solicitado, en el formato correspondiente: JPEG, PDF, etc. por lo que no tiene un esquema XML asociado.

getResourceStatusX

Disponible en:

http://www.dilve.es/dilve/api/xsd/getResourceStatusXResponse/1.0/getResourceStatusXRespons.xsd

y en,

http://www.dilve.es/dilve/api/xsd/getResourceStatusXResponse/getResourceStatusXResponse.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- DILVE API: getResourceStatusX response schema -->
<!-- (c)2009 DILVE - www.dilve.es - -->
<!-- Revision history -->
<!-- 1.0 - 2020/05/06: Initial version -->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
targetNamespace="http://www.dilve.es/dilve/api/xsd/getResourceStatusXResponse"
xmlns:g="http://www.dilve.es/dilve/api/xsd/getResourceStatusXResponse"
version="1.0">

   <xsd:element name="getResourceStatusXResponse">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element minOccurs="0" ref="g:header"/>
            <xsd:element minOccurs="0" ref="g:existingRecords"/>
            <xsd:element minOccurs="0" ref="g:Records"/>
            <xsd:element minOccurs="0" ref="g:error"/>
         </xsd:sequence>
         <xsd:attribute name="version" use="required" type="xsd:string"
fixed="1.0"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="header">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:fromDate"/>
            <xsd:element ref="g:toDate"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="existingRecords">
      <xsd:complexType>
         <xsd:attribute name="count" use="required" type="xsd:integer"/>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="Records" type="g:recordsType"/>

   <xsd:complexType name="recordsType">
      <xsd:sequence>
         <xsd:element minOccurs="0" maxOccurs="unbounded" ref="g:record"/>
      </xsd:sequence>
      <xsd:attribute name="count" use="required" type="xsd:integer"/>
   </xsd:complexType>

   <xsd:element name="record">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:id"/>
            <xsd:element minOccurs="0" ref="g:resourceTouchDate"/>
            <xsd:element minOccurs="0" ref="g:recordTouchDate"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="id">
      <xsd:complexType>
         <xsd:simpleContent>
            <xsd:extension base="xsd:string">
               <xsd:attribute name="type" use="required" type="xsd:string"/>
```

```
            </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="error">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element ref="g:code"/>
            <xsd:element ref="g:text"/>
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>

   <xsd:element name="code" type="xsd:string"/>
   <xsd:element name="text" type="xsd:string"/>

   <xsd:element name="resourceTouchDate" type="xsd:string"/>
   <xsd:element name="recordTouchDate" type="xsd:string"/>
   <xsd:element name="fromDate" type="xsd:string"/>
   <xsd:element name="toDate" type="xsd:string"/>

</xsd:schema>
```

Apéndice II: Códigos de error de DAPI

Introducción

- Se asignará un código de error único para cada tipo de error o situación de error genéricos.
- No se crearán códigos de error específicos asociados a parámetros concretos.
- Si la llamada http contiene varios errores distintos, solo se devolverá uno de ellos. No se intentará hacer un análisis completo de la llamada.
- Los códigos numéricos de error se agrupan por "familias" de errores conceptualmente semejantes, de forma que la adición de nuevos códigos de error se podrá hacer de forma ordenada y estructurada.

Familias de errores

Los códigos de error correspondientes a llamadas http en DILVE van desde el 1000 al 1900.

Estos se agrupan en hasta diez familias de error: de la 10xx a la 19xx

Las familias son:

- **10xx**: La llamada se ha completado, con éxito, total o parcialmente. Al usuario no se le ha devuelto un mensaje de error.
- **11xx**: Errores de identificación o permisos
- **12xx**: Faltan parámetros
- **13xx**: Sobran parámetros
- **14xx**: Valores de parámetros incorrectos / inconsistentes
- **15xx**: No se encuentra la información solicitada
- **16xx**: Reservado
- **17xx**: Reservado
- **18xx**: Reservado
- **19xx**: Problemas en el sistema / servidor / aplicación

Código	Error / Estatus	Comentario
1000	Llamada finalizada correctamente	
1001	No se han devuelto todos los registros solicitados por no encontrarse todos en DILVE	Se han devuelto algunos de los registros solicitados pero otros no se han encontrado en DILVE, aunque los identificadores eran correctos. Si solo se pide un ISBN y no se encuentra, se devolverá el estatus 1501.
1002	Se ha devuelto algún registro, pero no todos, ya que había algún identificador incorrecto	Si solo se pide un ISBN y está mal formado, se devolverá el estatus 1401
1003	Algún registro no se ha podido extraer, por problemas en la aplicación	Se ha encontrado un problema al formatear algún registro, pero se ha devuelvo correctamente, al menos un registro. Si no se ha podido devolver ninguno, se almacenará el estatus.
1004	Algún registro no se ha podido extraer porque los metadatos no son codificables en el formato solicitado	

Código	Error / Estatus	Comentario
1100	*Problemas de identificación / permisos*	
1101	Usuario o clave incorrecto	
1102	No tiene permiso para acceder a este recurso	Es posible que, en algún momento, se implementen jerarquías de permisos.
1120	Solicitud denegada. Consulte con Asistencia DILVE.	
1200	*Faltan parámetros*	
1201	Falta un parámetro obligatorio	
1202	Falta un parámetro dependiente	
1203	Figura un parámetro dependiente de otro que falta	
1300	*Sobran parámetros*	
1301	Parámetro desconocido	
1302	Parámetro incompatible con el resto	
1400	*Valores erróneos / inconsistentes*	
1401	Identificador mal formado	Si se solicitan varios ISBN, y se ha podido devolver alguno, pero hay alguno incorrecto, se devolverá el estatus 1002
1410	Valor de parámetro incorrecto	
1411	Valor de parámetro inconsistente con otros parámetros	
1420	Formato de fecha incorrecto	
1421	Valor de fecha incorrecto	
1422	Valor de fecha inconsistente	
1423	Rango de fechas inconsistente	
1424	Valor de tipo de editorial incorrecto	
1500	*No se encuentra información*	
1501	No se encuentra en DILVE el registro identificado por el ISBN (u otro identificador)	Si se solicitan varios ISBN (todos ellos correctos) y se ha podido devolver alguno, se generará el estatus 1001
1502	No existe el archivo de recurso solicitado	Si no existe el registro correspondiente, se devolverá el estatus 1501
1503	No existe la programación	
1504	No existe el mapa CSV	
1505	Libro digital. Metadatos no codificables en el formato solicitado	
1506	El recurso solicitado se modificó anteriormente	
1900	*Estatus del servidor*	
1901	DILVE en mantenimiento	

Código	Error / Estatus	Comentario
1902	Servidor sobrecargado. Intente más tarde.	Se dará un estatus http 503 también
1950	No se han podido extraer los registros solicitados, por problemas en la aplicación.	

Mapeos de errores

Código	fichalibro / CEGAL	getRecord...X y getResourceStatusX	getResourceX
1000	- (200)	- (200)	- (200)
1001	[x] (200) (a)	- (200)	- (200)
1002	[x] (200) (a)	- (200)	- (200)
1003	[x] (200) (a)	- (200)	- (200)
1004	[x] (200) (a)	- (200)	- (200)
1101	[2] (200)	[cod] (200) (c)	(403) (b)
1102	[1] (200)	[cod] (200) (c)	(403) (b)
1120	[1] (200)	[cod] (200) (c)	(403) (b)
1201	[7] (200)	[cod] (200) (c)	(400) (b)
1202	[7] (200)	[cod] (200) (c)	(400) (b)
1203	[7] (200)	[cod] (200) (c)	(400) (b)
1301	[7] (200)	[cod] (200) (c)	(400) (b)
1302	[7] (200)	[cod] (200) (c)	(400) (b)
1401	[x] (200) (a)	[cod] (200) (c)	(400) (b)
1410	[7] (200)	[cod] (200) (c)	(400) (b)
1411	[7] (200)	[cod] (200) (c)	(400) (b)
1420	[7] (200)	[cod] (200) (c)	(400) (b)
1421	[7] (200)	[cod] (200) (c)	(400) (b)
1422	[7] (200)	[cod] (200) (c)	(400) (b)
1423	[7] (200)	[cod] (200) (c)	(400) (b)
1501	[x] (200) (a)	[cod] (200) (c)	(404) (b)
1502	[x] (200) (a)	[cod] (200) (c)	(404) (b)
1503	[x] (200) (a)	[cod] (200) (c)	(404) (b)
1504	[x] (200) (a)	[cod] (200) (c)	(404) (b)
1505	[x] (200) (a)	[cod] (200) (c)	(404) (b)
1506	[x] (200) (a)	[cod] (200) (c)	(404) (b)

Código	fichalibro / CEGAL	getRecord...X y getResourceStatusX	getResourceX
1901	[1] (200 / 500?)	(503) (b)	(503) (b)
1902	[1] (503)	(503) (b)	(503) (b)
1950	[x] (200) (a)	(400) (b)	(400) (b)

(a) En fichalibro / CEGAL: se devolverá un bloque ERRORS con los ISBN que falten. Los códigos serán [0] para los que no se hayan podido extraer por problemas en la aplicación, [4] para los ISBN no válidos, [5] para los ISBN no encontrados, [6] para los ISBN de libros electrónicos sin soporte físico especificado.

(b) En el caso de estatus http distinto de 200, en la cabecera HTTP DILVE_ERROR se dará una indicación del problema encontrado.

(c) Se devolverá un XML con el código de error que figura en la primera columna.