Seneca FX API

Introducción

Séneca FX implementa un API accesible mediante el protocolo HTTP con un esquema de Urls de tipo **RESTful**, accesible en la ruta de primer nivel **/api**.

La información proporcionada por este API se entrega formateada como documento **Xml** o **Json** a criterio del solicitante, sin más que especificar la extensión correspondiente al formato deseado en la Url, siendo el formato por defecto Xml.

Así, el método general para acceder al API de Séneca FX será realizar una consulta HTTP con una Url que debe tener la siguiente estructura:

```
http://<host>/api/<resource>[<format>][?<options>]
```

donde <*resource*> sigue las normas RESTful para la denominación de los recursos, <*format*> puede ser `.xml` o `.json`, y <*options*> podrá contener uno o más parámetros en aquellas consultas para las que se haya implementado algún tipo de opción.

La consulta más sencilla al API de Séneca FX la proporciona el recurso `echo`, que permite comprobar si el servidor está operativo y el API habilitado.

En un navegador tecleamos:

```
http://<host>/api/echo?PING
```

a lo que el servidor responderá con en formato XML -ya que no se ha indicado el formato en la Url-,

```
<response version="1.0.1">
  <header>
    <timestamp>2018-02-13 15:33:14 +0100</timestamp>
    <timezone>(GMT+01:00) Europe/Madrid</timezone>
    <request>/api/echo?PING</request>
    <result>
      <status>
      200
      </status>
      <message>
      PONG
      </message>
    </result>
  </header>
</response>
```

y en caso de que indiquemos que deseamos la respuesta en formato JSON

```
http://<host>/api/echo.json?PING
```

a lo que el servidor responderá en formato JSON,

```
{
  "version": "1.0.1",
  "header": {
    "timestamp": "2018-02-13 15:33:14 +0100",
    "timezone": "(GMT+01:00) Europe/Madrid",
    "request": "/api/echo.json?PING",
    "result": {
      "status": 200,
      "message": "PONG"
    }
  }
}
```

En los siguientes apartados se comentarán la estructura general de las respuestas, los códigos de estado y una relación de los recursos disponibles a través del API de Séneca FX.
Formato de la respuestas del API

El API de Séneca FX entregará como respuesta un documento XML o JSON, que se compone de los siguientes elementos :

- elemento `response`, es el elemento raiz en el formato XML, no se utiliza en JSON
- elemento `header`, contiene una referencia a Url y hora de la consulta que provoca la respuesta por parte del API, un código de estado inspirado en los códigos HTTP y opcionalmente un mensaje descriptivo del resultado de la operación.
- elemento `data`, contiene la información solicitada.

Como ejemplo, la respuesta - abreviada - del API cuando solicitamos el registro de una sesión, en este caso la que tiene el ID 250, tiene el siguiente aspecto:

(consulta)

```
http://<host>/api/items/250.xml
```

(respuesta)

```
<response version="1.0.1">
   <header>
      <timestamp>2018-02-13 12:52:18 +0100</timestamp>
      <timezone>(GMT+01:00) Europe/Madrid</timezone>
      <request>/api/items/250</request>
      <result>
         <status>200</status>
         <message/>
      </result>
    </header>
    <data>
```

```
      <record>
          <item id="250" uuid="32561571-6f30-41eb-99b8-f3bc33c79815"
                href="http://server.spica.es/library/items/9-lexislatura-
comision-6-industria-2015-10-06"
                poster="http://server.spica.es/p/items/250.jpg"/>
      </record>
    </data>
</response>
```

y en el caso de realizar la misma consulta pero indicando el formato Json para la respuesta:

(consulta)

```
http://<host>/api/items/250.json
```

(respuesta) ```json "version": "1.0.1", "header": { "timestamp": "2018-02-13 13:10:00 +0100", "timezone": "(GMT+01:00) Europe/Madrid", "request": "/api/items/250.json", "result": { "status": 200, "message": "" } }, "data": { "record": { "item": { "id": 250, "uuid": "32561571-6f30-41eb-99b8-f3bc33c79815", "href": "http://server.spica.es/library/items/9-lexislatura-comision-6-industria-2015-10-06", "poster": "http://server.spica.es/p/items/250.jpg" }, }, } ```
>NOTA: En lugar de las extensiones .xml y .json, se puede utilizar la cabecera HTTP `Accept` con los valores `application/xml` y `application/json`.
API response *status* vs HTTP *status*

Las respuestas del API llevan un código de estado que sigue el estándar HTTP - 200 para "Ok", etc. - pero a diferencia del estándar REST, este no se proporciona como valor *Status* de la respuesta HTTP, que siempre será 200, sino como valor del elemento `/response/header/result/status` en el cuerpo de la respuesta del API.

Por ejemplo, si realizamos la siguiente consulta a un recurso no existente/no implementado:

```
http://<host>/api/resource_name_that_is_not_implemented.xml
```

obtendremos una repuesta con el valor HTTP *Status* igual a 200, donde la sección `/response/header/result` sería la siguiente

```
<result>
  <status>
  404
  </status>
  <message>
  La URL que está utilizando podría contener un error tipográfico, la
página podría haberse movido, o puede que solo esté temporalmente fuera de
línea
  <message>
</result>
```

en donde el código de estado del API para la respuesta es 404, y debe interpretarse según la guía del estándar HTTP ("Not Found").

Además del código de estado, el API puede incluir el elemento `/response/header/result/message` con un texto explicativo del motivo concreto que ha provocado la respuesta, especialmente útil cuando diferentes tipos de errores se agrupan en un mismo código de estado.

API response `data`

Sin duda la parte más importante de la respuesta del API es el elemento `/response/data` ya que contiene la información que le solicitamos. Este elemento, que se incluye únicamente en respuestas con código 200, aunque no en todas, presenta tres posibilidades,

- que se incluya *un sólo registro*, como elemento `record`, cuando la ruta del recurso haga referencia un identificador.
- que se incluya *una lista de registros*, como elemento `list`, cuando la ruta de la consulta haga referencia a una colección, aún cuando tenga un único elemento.
- que se incluya *un arbol (o descendiente en un arbol) de registros*, como elemento `tree`, cuando la ruta de la consulta haga referencia a una colección o a un registro cuyo modelo de datos sea del tipo jerárquico.

`response/data/record`

Este caso se dará cuando la ruta del resurso REST termine con un identificador (con alguna excepción). Un ejemplo sería la consulta de una sesión ya vista anteriormente (*/api/items/250*), pero que ahora se mostrará completa en contraste con la respuesta a la consulta de una legislatura, simplemente para resaltar que la estructura del contenido de un elemento `response/data/record` puede ser más o menos compleja según el tipo de recursos solicitado.

```
http://<host>/api/items/250.json
```


```
{
  "version": "1.0.1",
  "header": {
    "timestamp": "2018-02-14 18:05:42 +0100",
    "timezone": "(GMT+01:00) Europe/Madrid",
    "request": "/api/items/250.json",
    "result": {
      "status": 200,
      "message": ""
    }
  },
  "data": {
    "record": {
      "item": {
        "id": 250,
        "uuid": "32561571-6f30-41eb-99b8-f3bc33c79815",
        "type": "asset",
        "visibility": null,
        "audience": 5,
        "rating": null,
        "published": "true",
        "featured": null,
```

```
        "favouritable": false,
        "favourite_id": null,
        "signeable": false,
        "user_signed": false,
        "href": "http://127.0.0.1:8980/library/items/actos-
institucionales-9-lexislatura-comision-6-industria-enerxia-comercio-e-
turismo-2015-10-06",
        "poster": "http://127.0.0.1:8980/p/items/250.jpg",
        "timestamp": {
          "created_at": "2016-05-06T14:33:39+02:00",
          "updated_at": "2016-11-28T17:44:20+01:00"
        },
        "event": {
          "type": "actofp",
          "date": "2015-10-06T16:30:35+02:00"
        },
        "media": {
          "count": 1,
          "duration": 7617,
          "duration2": "02:06:57",
          "size": 732464530
        },
        "xactofp": {
          "xmeeting_id": 250,
          "xlegs_id": 12,
          "xlegs_name": "9ª Lexislatura",
          "xgbody_id": 216,
          "xgbody_name": "Comisión 6ª Industria, Enerxía, Comercio e
Turismo."
        },
        "metadata": {
          "event": "actofp",
          "title": "Comisión 6ª Industria, Enerxía, Comercio e Turismo",
          "description": null,
          "date": "2015-10-06T00:00:00+02:00",
          "legs": "9ª Lexislatura",
          "gbody": "Comisión 6ª Industria, Enerxía, Comercio e Turismo.",
          "actofp": {
            "legs": {
              "id": 12,
              "name": "9ª Lexislatura"
            },
            "gbody": {
              "id": 216,
              "name": "Comisión 6ª Industria, Enerxía, Comercio e
Turismo."
            }
          },
          "classifiers": [
            {
              "taxonomy_id": 9,
              "taxonomy_name": "Actos Institucionales",
              "id": 216,
              "uuid": "a7b35a63-9035-4e6c-aafa-2482504a2a93",
```

```
                "name": "Comisión 6ª Industria, Enerxía, Comercio e
Turismo.",
                "tname": "Comisión 6ª Industria, Enerxía, Comercio e
Turismo.",
                "qname": "/9ª Lexislatura/Comisión 6ª Industria, Enerxía,
Comercio e Turismo.",
                "emblem": "C6-IX"
              }
            ]
          }
        }
      }
    }
}
```

y para la legislatura 216

```
http://<host>/api/legs/216.json
```

```
{
  "version": "1.0.1",
  "header": {
    "timestamp": "2018-02-14 18:06:41 +0100",
    "timezone": "(GMT+01:00) Europe/Madrid",
    "request": "/api/legs/216",
    "result": {
      "status": 200,
      "message": ""
    }
  },
  "data": {
    "record": {
      "leg": {
        "id": 12,
        "name": "9ª Lexislatura",
        "tname": "9ª Lexislatura",
        "emblem": "IX",
        "smartlist": true,
        "slug": "actos-institucionales-9-lexislatura"
      }
    }
  }
}
```

`response/data/list`

Cuando la ruta REST termine con una nombre de colección, generalmente el plural del mismo nombre utilizado la consulta de un registro individual, se obtendrá un elemento `response/data/list` que contendrá la relación de los registros individuales de la colección.

Así, como contraste con la respuesta a la consulta de una sesión, se muestra el resultado de la consulta de todas las sesiones, en la que se suprime el identificador al final de la ruta del recurso,

```
http://<host>/api/items
```

```
<response version="1.0.1">
  <header>
    <timestamp>2018-02-13 14:03:30 +0100</timestamp>
    <timezone>(GMT+01:00) Europe/Madrid</timezone>
    <request>/api/items</request>
    <result>
      <status>200</status>
      <message/>
    </result>
  </header>
  <data>
    <list page="1" page_records="20" page_offset="0" total_records="150"
total_pages="8" page_size="20">
      <items>
        <item id="7638" uuid="b3b945ea-33d8-4215-8015-2ca78a523e18"
              href="http://r620.spica.es/library/items/probas-pleno-2018-
02-02"
              poster="http://r620.spica.es/p/items/7638.jpg"/>

        .
        .
        .

        <item id="7615" uuid="5fb347d9-bd8c-431c-bf35-56e60d7ff06"
              href="http://r620.spica.es/library/items/comision-5-
sanidade-politica-social-emprego-2018-01-17"
              poster="http://r620.spica.es/p/items/7615.jpg"/>
      </items>
    </list>
  </data>
<response/>
```

> Hay que tener en cuenta que en resultado de consultas sobre elementos de una colección depende de los filtros explicitos que pudiese incluir la consulta, pero también de aquellos 'fijos' más los que se deriven del 'contexto' de la consulta, tales como el usuario que la realiza.

El número de registros para la repuesta a una consulta de este tipo puede ser lo suficientemente elevado como para hacer inviable incluirlos todos de una sola vez y en consecuencia, el API aplica -siempre- un algoritmo de paginación a este tipo de consultas, independientemente de si el número de resultados es más o menos elevado.

Para gestionar está paginación, el elemento `data/list` incluye los siguientes atributos:

- `page`, es el número de página del documento entregado por el API. Si no se especifica un página en particular en la consulta se entregará la primera página. Para obtener una página distinta se pude incluir la opcion *page*
- `page_size`, es el número máximo de registros en una página. El servidor utiliza un tamaño de página predeterminado y si deseamos uno distinto podemos incluirlo como opción *page_size* en la

consulta.

- `page_records`, es el numero de registros en el documento
- `page_offset`, es el ordinal del primer registro en la lista de resultados objecto de la consulta
- `total_records`, es el número de registros objeto de la consulta, una vez aplicados los filtros que correspondan
- `total_pages`, es el número de páginas objeto de la consulta, una vez aplicados los filtros que correspondan

Como ejemplo, para obtener la tercera página de la lista de sesiones, con 100 sesiones por página, la consulta sería la siguiente

```
http://<host>/api/items?page=3&page_size=100
```

Hay que tener en cuenta que si la página solicitada o el número de registros por página solicitado están 'fuera de rango' el servidor utilizará el tamaño de página por defecto y sobre esta base devolverá la primera o última página (según lo que se haya solicitado). Debido a esto, la página solicitada o el número de elementos en la página podrían no coincidir con los solicitados y como consecuencia para una correcta gestión de las paginas es necesario analizar los valores de pagicación devueltos por el servidor en cada respuesta que contenga un elemento `data/list`.

Otro aspecto a tener en cuenta, es que una consulta sobre la lista de elementos de una colección vacía - especialmente si se aplican filtros - generará una respuesta con cero registros y no una repuesta con código de estado 404 (Not Found).

`response/data/tree`

Si el modelo de datos del recurso solicitado es del tipo jerárquico, obtendremos una elemento que será un híbrido de elementos *record* y *list* con sus descendientes, registros y colecciones de registros y se distinguirá de estos por su etiqueta específica *tree*.

La única consulta que actualmente retorna este tipo de datos se utiliza en el para obtener la lista de carpetas (virtuales)

```
http://<host>/api/folders
```

{ "version": "1.0.1", "header": { "timestamp": "2018-02-15 15:50:48 +0100", "timezone": "(GMT+01:00) Europe/Madrid", "request": "/api/folders", "result": { "status": 200, "message": "" } }, "data": { "tree": { "folders": [{ "id": 6, "uuid": "$actofp$", "parent_id": 1, "name": "Actos Institucionales", "audience": 5, "items_count": 0 }, { "id": 2, "uuid": "$public$", "parent_id": 1, "name": "Archivo General", "audience": 5, "items_count": 0 }] } } }

El API con usuario invitado.

El API de Séneca FX entrega información en función de nivel de acceso del solicitante.

Si se realizan consultas sin identificarse, el servidor utilizará las reglas de acceso para el usuario *invitado* y estas únicamente permiten acceso a información pública, exactamente la misma que se puede obtener a través de la Web sin iniciar sesión. Además, el único método HTMP admitido como usuario invitado es GET;, con lo cual, no se podrán hacer cambios en la base de datos.

Un detalle importante es que el usuario invitado puede deshabilitarse, en cuyo caso la utilización del API pasa obligatóriamente por proporcionar unas credenciales de usuario de Séneca FX válidas.

El API con sesión de usuario.

Si se ha iniciado sesión en la Web de Séneca FX (de forma interactiva) los accesos al API tendrán en cuenta automáticamente el contexto de usuario de la sesión en curso. Por este motivo, no es posible utilizar un navegador para iniciar sesión en la Web de Séneca con un usuario y consultar al API con uno distinto.

Además, el API de Séneca FX nunca solicitará una contraseña, por lo que no se podrán realizar consultas con Urls del tipo

```
http://<nombre_de_usuario>@<host>/api/....
```

Sin embargo, si accedemos al API desde un utilidad de consola o mediante un lenguaje de programación, podremos incluir el nombre de usuario y clave de acceso sin que el servidor la solicite, o directamente, proporcionar el valor valor correcto en la cabecera HTTP 'Authorization'.

Como ejemplo, utilizando CURL,

```
curl -H "Authorization: Basic c3BpY2E6c3BpY2E=" http://<host>/api/...
```

donde la cadena "c3BpY2E6c3BpY2E=" es el Base 64 de la cadena formado por el nombre de usuario y password (estandar HTTP), o también,

```
curl -u <usuario>  http://<host>/api/...
```

donde CURL solicitará la contraseña (no esperará que el servidor se la pida) y generará la cabecera "Authorization" con la cadena correcta.

Los accesos al API se autentificarán utilizando el valor de la cabecera "Authorization", pero, tendremos que incluirla en todas las consultas ya que el API de Séneca FX no genera cookies que permitan mantener la sesión iniciada.

En cualquier caso, cuando se accede al API mediante algún tipo de script es recomendable desactivar las cookies.

Utilidades de consola para acceso al API

Como parte de la distribución de Séneca FX se incluye varios scriots que simplifican el acceso al API desde la consola, en la carpeta */opt/spica/seneca/seneca/api*.

Los scripts incluidos son:

_ seneca-api.sh _

Esta utilidad sirve para realizar consultas y mostrar el resultado en el terminal o guardarlo en un archivo. Está escrito en BASH y puede ejecutarse tanto en Linux como en Cygwin (Windows). Una función importante de este script que se puede incluir un archivo de configuración con el nombre de host, puerto, usuario, password y formato deseado para la consulta con lo cual la consulta típica con el CURL pasaría de

```
curl -H "Authorization: Basic XXXXXXXXX"
http://127.0.0.1:8980/api/items.json
```

pasaría a ser

```
seneca-api.sh -c ./test.cfg /api/items
```

supuesto que el archivo test.cfg define las siguientes variables de entorno

```
SAPI_HOST=127.0.0.1
SAPI_PORT=8980
SAPI_USER=UUUUUUUU
SAPI_PASS=PPPPPPPP
```

_ **seneca-api-batch.sh** _

Este script procesa un archivo de texto, ejecutando un consulta al API por para cada línea que comienza con "GET;". En el directorio ./test se incluyen varios ejemplos con el nombre ./test/test-set1.txt, ./test/test-set3.txt, etc.

_ **seneca-api-test.sh** _

Basándose en el anterior, realiza un bloque de consultas preestablecidas (archivos ./test/test-set*.txt) que permiten testear las funciones del API almacenando los resultados de las consultas en la carpeta *test.out*.

Para ejecutar los test del API :

```
$ cd /opt/spica/seneca/seneca/api
$ chmod +x *.sh
$ ./seneca-api.test.sh
$ ls -l ./test.out
200-api_echo_PING.json
200-api_gbodies_273.json
200-api_gbodies_273_people.json
200-api_gbodies_273_positions_gbody.json
200-api_gbodies.json
200-api_gbodies_PL-X.json
200-api_gbodies_PL-X_people.json
200-api_gbodies_PL-X_positions_gbody.json
200-api_items_2018-02-01-1.json
200-api_items_7514.json
.
.
.
```

Recursos del API

*** Legislaturas ***

Recurso	Descripción
/api/legs	Lista de Legislaturas
/api/legs/:legs_id	Detalle de una Legislatura
/api/legs/current	Detalle de Legislatura actual

El valor *:legs_id* puede ser del identificador o el código (a.k.a. *emblem*).
Se puede utilizar el código especial *current* para la legislatura actual.

*** Organos Parlamentarios ***

Recurso	Descripción
/api/gbodies	Lista de Órganos Parlamentarios (todas las Legislaturas)
/api/gbodies/:gbody_id	Detalle de Órgano Parlamentario
	|

/api/legs/:legs_id/gbodies | Lista de Órganos Parlamentarios de una Legislatura /api/legs/current/gbodies| Órganos Parlamentarios de la Legislatura Actual

El valor *:legs_id* puede ser del identificador o el código (a.k.a. *emblem*).
El valor *:gbody_id* puede ser del identificador o el código (a.k.a. *emblem*).

*** Grupos Políticos ***

Recurso	Descripción
/api/politics	Lista de Grupos Políticos
/api/politics/:politics_id	Detalle de Grupo Político
	|

El valor *:politics_id* puede ser del identificador o el código (a.k.a. *emblem*).

*** Cargos (en Organo Parlamentario) ***

Recurso	Descripción
/api/positions_gbody	Lista de Cargos
/api/positions_gbody/:position_id	Detalle de Cargo
/api/gbogies/:gbody_id/positions_gbody	Cargos en Órganos Parlamentario

El valor *:position_id* puede ser del identificador o el código (a.k.a. *emblem*).
El valor *:gbody_id* puede ser del identificador o el código (a.k.a. *emblem*).

*** Títulos ***

Recurso	Descripción
/api/positions_title	Lista de Títulos de Oradores
/api/positions_title/:position_id	Detalle de Título de Orador

El valor *:position_id* puede ser del identificador o el código (a.k.a. 'emblem').

*** Oradores ***

Recurso	Descripción
/api/people	Lista de Oradores (todos o filtrado)
/api/people/:person_id	Detalle de Orador
/api/people/:person_id/politics	Lista de Grupos Políticos de un Orador
/api/people/:person_id/positions_title	Lista de Títulos de un Orador
/api/people/:person_id/positions_gbody	Lista de Cargos de un Orador
/api/politics/:politics_id/people	Lista de Oradores de un Grupo Parlamentario
/api/gbodies/:gbody_id/people	Lista de Oradores con cargo en un Órgano Parlamentario
/api/legs/:legs_id/people	Lista de Oradores con cargo en algún Órgano Parlamentario de una Legislatura

El valor de *:person_id* puede ser del identificador del orador, o el código (a.k.a 'pericode') de un Orador.
El valor *:politics_id* puede ser del identificador o el código (a.k.a. *emblem*) de un Grupo Político.
El valor *:gbody_id* puede ser del identificador o el código (a.k.a. *emblem*) de un Organo Parlamentario.
El valor *:legs_id* puede ser del identificador o el código (a.k.a. *emblem*) de una Legislatrura.

*** Sesiones (a.k.a. Items) ***

Recurso	**Descripción**
/api/items	Todas las sesiones
/api/items/*:item_id*	Detalle de sesión
/api/items/last	Última sesión
/api/items/fecha*[-:*n*]	Detalle de sesión por fecha y ordinal
/api/items/[:*gbody_id*-]:*fecha*[-:*n*]	Detalle de sesión por Organo Parlamentario, fecha y ordinal

El valor de *:item_id* puede ser el identificador de la sesión, el código especial *last* (para la última sesión), la fecha y ordinal (opcional en el caso de 1) y por último, el código del Organo Parlamentario más la fecha mas el ordinal.

Ejemplos:

- Obtener detalle de una sesión conociendo únicamente la fecha y el Organo Parlamentario.

```
/api/items/2018-02-01
```

devolverá el detalle de la primera sesión del 1 Ene. 2018 (siempre y cuando haya alguna).

- Si además de la fecha, conocemos el identificador o código del Organo Parlamentario, Ej. 'PL-X', se puede indicar de la siguiente manera

```
/api/items/PL-X:2018-02-01
```

- Aun así, puede haber más de una sesión en cuyo caso se devolverá la primera. En estos casos podemos añadir el ordinal de la sesión para esa fecha, ej. en el caso anterior, para obtener la segunda

```
/api/items/PL-X:2018-02-01-2
```

*** Filtrar la lista de sesiones ***

El recurso */api/items* puede devolver una lista de sesiones filtrada con las siguientes opciones

Opción	**Descripción**
collection_id	Identificador de una colección que puede ser *recents* o *featured*.
legs_id	Identificador/Código de una Legislatura

Opción	Descripción
gbody_id	Identificador/Código de un Órgano Parlamentario
date	Fecha con formato YYYY-MM-DD
date_from,date_to	Rango de fechas, con formato YYYY-MM-DD
title	Texto que debe aparecer en el título de la Sesión

Ejemplos:

- Todas las sesiones en una fecha

```
/api/items?date=2018-01-01
```

- Todas las sesiones en un rango de fechas de la legislatura 'X'

```
/api/items?legs_id=X&date_from=2018-01-1&date_to=2018-01-31
```

- Todas las sesiones del Pleno de la legislatura actual, en una fecha

```
/api/items?legs_id=current&gbody_id=PL-X&date=2018-01-1
```

*** Listas de reproducción, media y marcadores ***

El *playlist* es un recurso especial que incluye la información relativa a una sesión o una grabación, con los marcadores y los recursos de descarga y streaming de la media.

Opción	Descripción
/api/items/:item_id/playlist	Recurso específico para generar la página de reproducción de un Sesión.
/api/items/:item_id/masters	Lista de todas la grabaciones de una Sesión
/api/items/:item_id/markers	Lista de marcadores de todas la grabaciones de una Sesión
/api/items/:item_id/sections	Lista de secciones (ptos. orden del día) de todas la grabaciones de una Sesión
/api/items/:item_id/speeches	Lista de intervenciones de todas la grabaciones de una Sesión
/api/items/:item_id/annotations	Lista de marcadores de todas la grabaciones de una Sesión

El valor de *:item_id* puede ser el identificador de la sesión, el código especial *last* (para la última sesión), la fecha y ordinal (opcional en el caso de 1) y por último, el código del Organo Parlamentario más la fecha

mas el ordinal.

Si se prefiere obtener la misma información, pero relativa a una única grabación

Opción	**Descripción**
/api/masters/:master_id	Detalle de una grabación
/api/masters/:master_id/playlist	Recurso específico para generar la página de reproducción de un Sesión.
/api/masters/:master_id/markers	Lista de marcadores de todas la grabaciones de una Sesión
/api/masters/:master_id/sections	Lista de secciones (ptos. orden del día) de todas la grabaciones de una Sesión
/api/masters/:master_id/speeches	Lista de intervenciones de todas la grabaciones de una Sesión
/api/masters/:master_id/annotations	Lista de marcadores de todas la grabaciones de una Sesión

donde el valor de *:master_id* puede ser del identificador de la grabación o la etiqueta.