



Consulta de servicios web REST

CUENTA DE DESARROLLADOR PARA USAR LA API DE LAST.FM

Para poder realizar el programa que se va a desarrollar, dado que va a trabajar sobre la API del Web Service de la plataforma musical Last.FM, crea una cuenta de desarrollador:

<https://www.last.fm/api>

OBTENER UN CLIENT_ID DE JAMENDO PARA USAR EN LAS CONSULTAS DE LA APLICACIÓN A DESARROLLAR

- Con la cuenta de desarrollador creada anteriormente, crear una aplicación desde <https://www.last.fm/api/account/create>
- Al crear la aplicación, se te otorgará un API Key que deberás usar en todas las consultas. Se pueden consultar las aplicaciones en <https://www.last.fm/api/accounts>
- Accede a la documentación de la API para consultar toda la información relativa al uso de la API: <https://www.last.fm/api/intro>

En cada uno de los métodos hay información sobre cómo realizar la consulta.

En el método para consultar los artistas más populares por países (<https://www.last.fm/api/show/geo.getTopArtists>), se muestra un ejemplo de peticiones, así como los parámetros obligatorios y opcionales. Por ejemplo

```
http://ws.audioscrobbler.com/2.0/?  
method=geo.gettopartists&country=spain&api_key=YOUR_API_KEY&format=json
```

Example URLs

JSON: /2.0/?method=geo.gettopartists&country=spain&api_key=YOUR_API_KEY&format=json

XML: /2.0/?method=geo.gettopartists&country=spain&api_key=YOUR_API_KEY

Params

country (Required) : A country name, as defined by the ISO 3166-1 country names standard

limit (Optional) : The number of results to fetch per page. Defaults to 50.

page (Optional) : The page number to fetch. Defaults to first page.

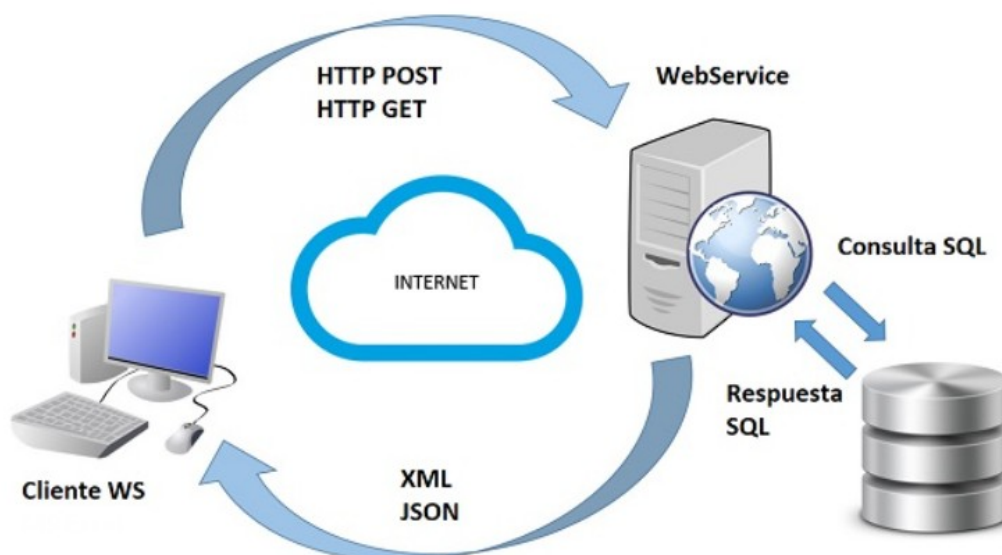
api_key (Required) : A Last.fm API key.

QUÉ ES UN WEB SERVICE Y COMO FUNCIONA

Con **web service** nos referimos a un servicio que nos ofrece un servidor web para proporcionarnos información en un documento con un formato de representación estandarizado (XML, JSON, otros) sobre los contenidos que se manejan en el sitio web que corresponda.

Muchos sitios web disponen de un **web service**. Cada **web service** dispone en servidor de varios métodos que responden a peticiones de los clientes. A estos métodos se los denomina en conjunto **API Web Service**.

Lo más importante de un **web service** es que, con su interface de diálogo con el cliente, le oculta la comunicación con la fuente de datos, generalmente con la base de datos. Un **web service** recibe peticiones **http** de un cliente para obtener o descargar datos o para subirlos y convierte esas peticiones en consultas a la base de datos.



Plataformas de descarga de música como **Spotify, Last.Fm o Jamendo** disponen de una API para que los desarrolladores realicen aplicaciones clientes que obtengan información sobre sus contenidos. Por ejemplo, que desde la **aplicación cliente** se realice una conexión con el servicio web para conocer los albums publicados por un artista. El servicio web responderá con un documento que incluye información sobre cada album registrado en la plataforma para ese artista (nombre del artista, título del album, enlace a la imagen de la carátula, enlace de descarga del album, etc.).

La **API de un Web Service** es la interface de diálogo del cliente con el web service, **es decir**, es el conjunto de métodos que puede usar el cliente para comunicarse con el **web service**. Muchas plataformas web **publican su API**, es decir dan información a los desarrolladores de cómo acceder a sus métodos y cual es el formato de los documentos de respuesta, por ejemplo, *Youtube, Netflix, Facebook, Google Maps, Aemet*, etc. La API de un otras plataformas u otros sitios web tienen su API aunque no está publicada, lo que no tiene porqué significar que los desarrolladores no la puedan utilizar.



Todo servicio web:

- Debe utilizar un protocolo de transporte para la comunicación de las aplicaciones cliente con el servicio web, normalmente HTTP o HTTPS.
- Para desarrollarlos se ha utilizado una tecnología. Fundamentalmente, se usan dos tecnologías, **SOAP** y **REST** (la más habitual).
- Intercambia los mensajes (peticiones desde el cliente y respuestas al cliente) con uno o varios formatos de representación, principalmente **JSON** y **XML**.

Se habla de **consumir un servicio web** para indicar que una aplicación cliente obtiene datos de un servicio web y/o sube datos a un servicio web.

EJERCICIO 1

Accede con el navegador web a <https://aplicaciones.ivanlorenzo.es/ad/dam2.json> y a <https://aplicaciones.ivanlorenzo.es/ad/dam2.xml>. Examina los ficheros.

Realiza ahora un programa que usa la clase **URLConnection** para representar en pantalla el texto de cada uno de los ficheros.

EJERCICIO 2

Realiza un programa que, consumiendo el web service de Last.FM y usando conexiones establecidas mediante **URLConnection**, presenta un menú como el siguiente:

- 1.- Datos de artistas más populares en tu país
- 2.- 30 mejores temas de un artista
- 3.- Datos de 20 artistas más populares de un usuario concreto
- 4.- Obtener 10 artistas similares a uno dado

Y en función de la opción seleccionada realiza lo siguiente:

Opción 1: Usando **SAX**, pide un número N que indica cuantos artistas se quieren mostrar y escribe un listado con los N artistas más populares de tu país. Se usará el método **geo.gettopartists** de la API de Last.FM.

En el listado, se numeran los artistas desde 1 y de cada artista se escribe su nombre y url de imagen.

**Hoja02_XML_05**

Introduce número de artistas que quieres obtener

10

1 - David Bowie	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
2 - Radiohead	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
3 - Queen	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
4 - Coldplay	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
5 - The Rolling Stones	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
6 - The Beatles	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
7 - Muse	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
8 - Daft Punk	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
9 - The Cure	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png
10 - Arctic Monkeys	Descargar imagen: https://lastfm.freetls.fastly.net/i/u/300x300/2a96cbd8b46e442fc41c2b86b821562f.png

Ten en cuenta que el método **parse** del parseador SAX puede recibir un flujo **InputStream** en lugar de un objeto **File** (como hemos hecho hasta ahora).

Opción 2: Usando SAX, pide el nombre de un artista y se escribirán los 30 mejores temas usando el método **artist.gettoptracks**.

De cada tema se mostrará el ranking, el nombre y la URL del track.

Opción 3: En este punto obtendremos el formato JSON. Deberás usar también la librería **GSON** para obtener los datos. El método usado es **user.gettopartists**.

Se escribirá únicamente el nombre de los 20 artistas más populares de un usuario dado.

Opción 4: Los datos vendrán el formato JSON. Se pide el nombre de un artista y se obtiene únicamente el nombre de los 10 artistas más parecidos a él.

Se usará el método **artist.getsimilar**