

Proyecto Interciclo SOA

Esteban Rosero & Yandry Romero & Ángel Ruiz.
Enero 2020.

Universidad Politécnica Salesiana.
Carrera de Computación.
Sistemas Distribuidos

Tabla de Contenidos

ii

Capítulo 1 Informe del proyecto Interciclo SOA.....	1
Shazam API	1
Arquitectura del sistema	1
Implementación.....	2
Conclusiones	9
Recomendaciones	9

Capítulo 1

Informe del proyecto Interciclo SOA

Shazam API

Con esta API podemos identificar cualquier canción, descubrir nuevas canciones, letras, videos y listas de reproducción. A esta API lo obtenemos de la página web de RapidAPI donde nos proporciona todo lo necesario para poder realizar nuestra implementación y consumir el web Service. La página web donde fue obtenida la información es: <https://rapidapi.com/apidojo/api/shazam/endpoints>

Arquitectura del sistema

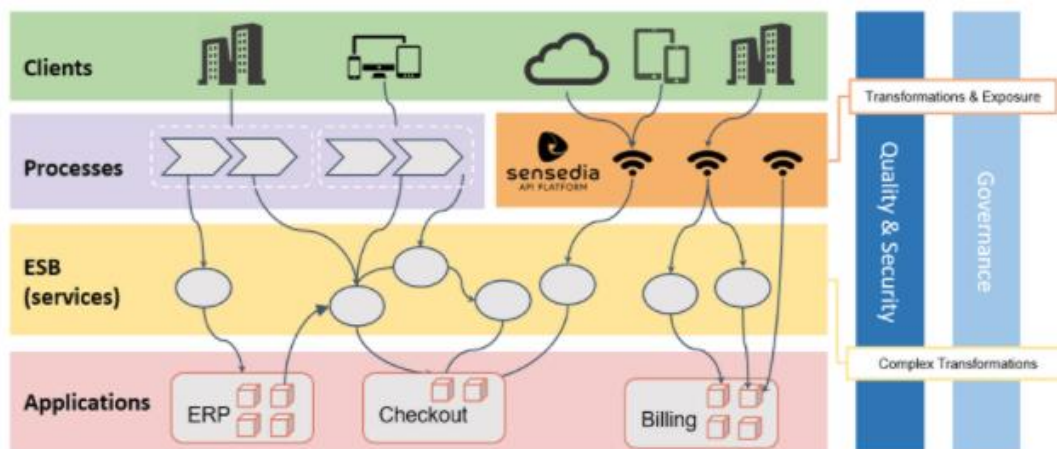
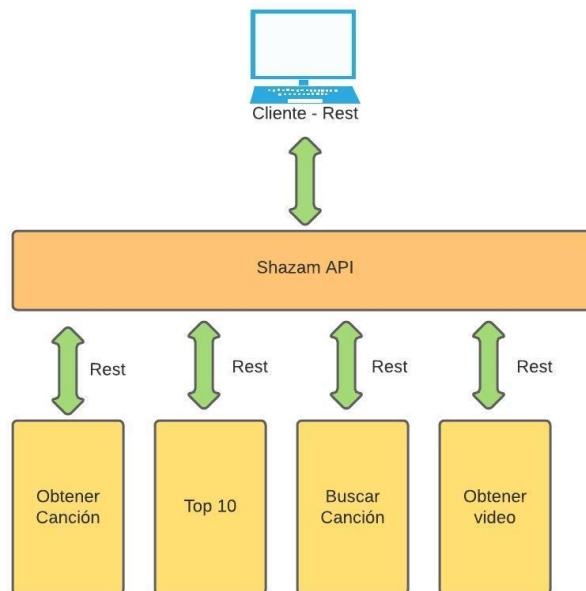


Imagen Obtenida de SENSEDIA <https://sensedia.com/es/apis/repensando-el-esb->

[arquitectura-soa-con-apis/](https://sensedia.com/es/apis/repensando-el-esb-arquitectura-soa-con-apis/)



Implementación

Para la implementación de nuestro SOA como primer paso a través de Anypoint Studio creamos nuestro proyecto mule que se encargara de consumir el servicio rest.

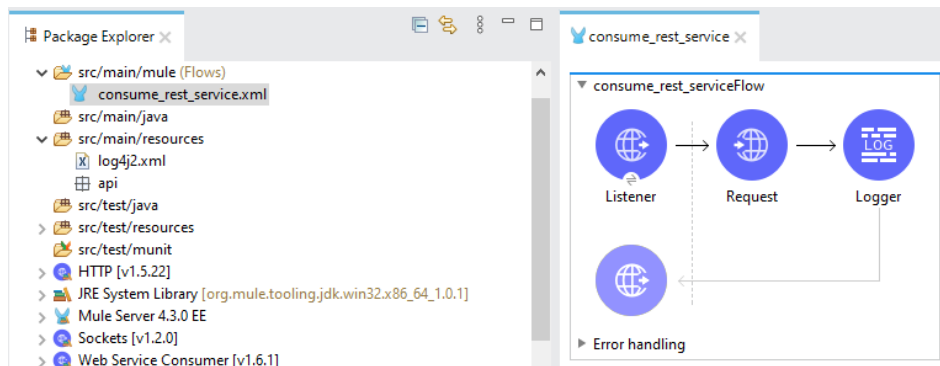


Fig. 1 Proyecto en Mule

Luego para consumir el servicio de la API de Shazam debemos ir creando nuestra estructura rest en Mule como primer paso es definir un Listener donde vamos a pasar los parámetros por donde va a obtener los datos y que datos nos va a consumir.

En este caso vamos a obtener las canciones y le pasamos como parámetro una llave requerida por la API y también vamos a configurar nuestro conector, en nuestro Request aquí definimos nuestro servicio Rest donde vamos a realizar la petición al servidor de los datos que estamos necesitando y mostrarlos por pantalla.

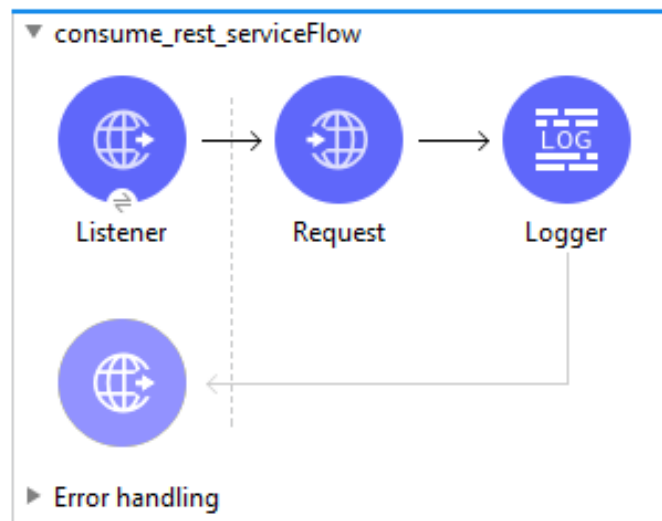


Fig 2. Arquitectura Mule Obtener Canción

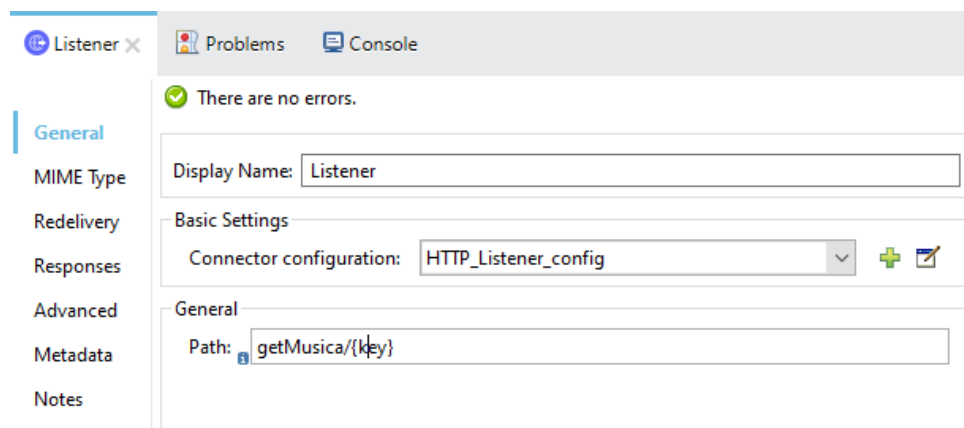


Fig. 3 Configuración Listener obtener Canción

Ya que tenemos configurado nuestro Listener, configuramos nuestro servicio rest para que nos consuma desde el servidor de nuestra API donde primero debemos definir que es un servicio rest de tipo get y luego configurar nuestro host para que recupere los datos, posterior a esto enviamos a mostrar los datos recuperados.

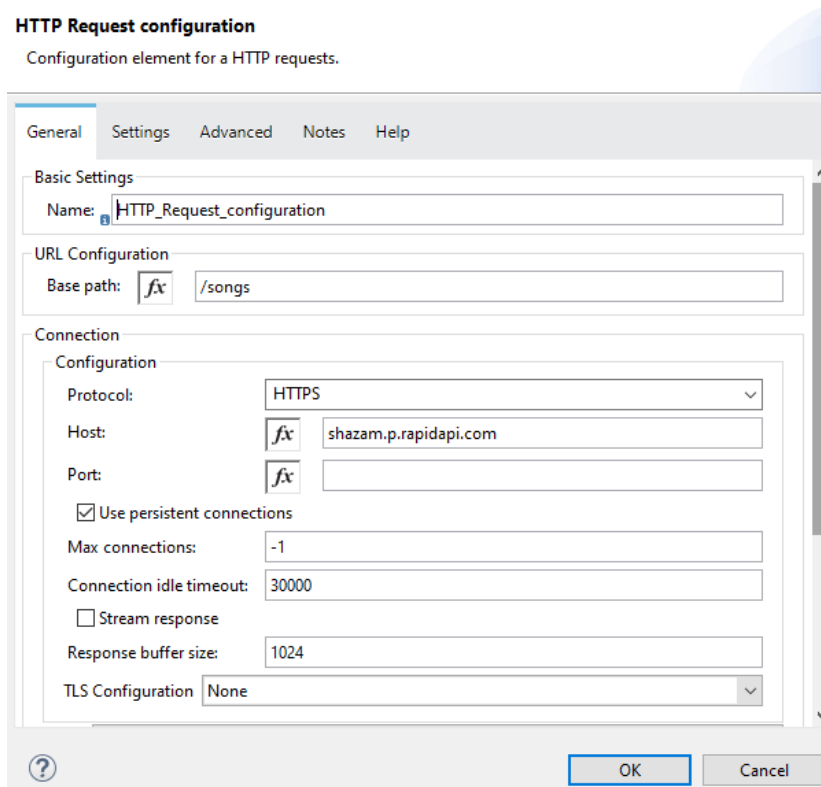


Fig 4. Configuración Request

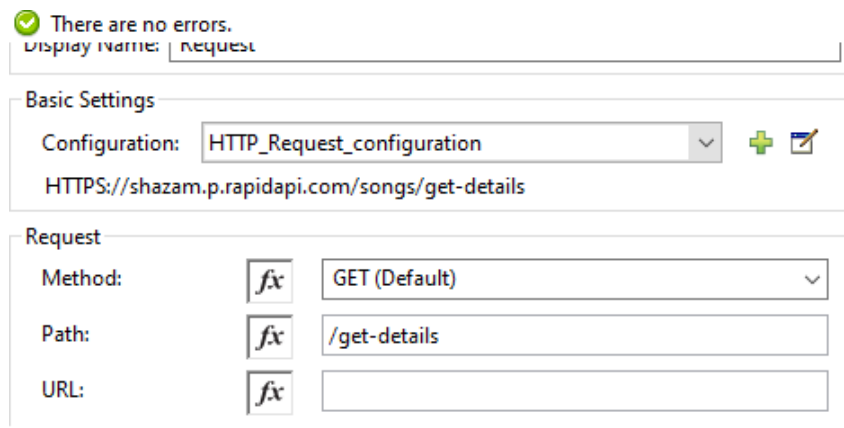


Fig. 5 Configuración Básica Rest

Para el presente proyecto interciclo consumimos tres servicios de nuestra API como en le punto anterior explicamos el obtener canción también implementamos 3 más que son: Obtener el top de canciones según las fecha que seleccionamos, También el

obtener la canción según los parámetros de nombre y limite de similitudes esto sería el buscar canción, y por último consumimos el obtener url del video para mostrarlo en nuestra página web.

La implementación en Mule del consumo rest solo nos muestra los datos obtenidos en forma de lista pero ahora vamos a implementarlo en java para que a través del orquestador consuma estos servicios y los podremos mostrar en una interfaz creativa en la web.

En java para la implementación de nuestro código estamos usando la librería de Faces que nos facilita el trabajo y el manejo de datos, como punto inicial creamos nuestro proyecto y creamos nuestro controlador en el cual vamos a definir todos los parámetros.

```
private String urlYT = "https://www.youtube.com/embed/tgbNymZ7vqY";
private boolean renderizado = false;
private String cancion;
private String letra;
private String imagen;
private Date fecha;
private String top1;
private String top2;
private String top3;
private String top4;
private String top5;
private String top6;
private String top7;
private String top8;
private String top9;
private String top10;
private String urlDescarga;
```

Fig 6. Parámetros del controlador

Para nuestra primera funcionalidad que es buscar el top de las canciones según la fecha a través de un try y catch para el manejo de excepciones recuperamos los datos y los almacenamos en un array para que nos muestre el top 10 según la fecha que seleccionamos, algo muy importante de esto es enviar nuestra url de donde se esta recuperando los datos para que pueda obtener la información.

recolectar los datos y nosotros a través de java solo escogemos lo que nos interesa para así mostrar en nuestra web service.

```
System.out.println("KEY: " + key);
StringBuilder resultado = new StringBuilder();
URL url = new URL("http://localhost:8081/getMusica/" + key);

URLConnection conexion = (URLConnection) url.openConnection();
conexion.setRequestMethod("GET");

BufferedReader rd = new BufferedReader(new InputStreamReader(conexion.getInputStream()));
String linea;

while ((linea = rd.readLine()) != null) {
    resultado.append(linea);
}

rd.close();
String yt = resultado.toString();
String[] lines = resultado.toString().split("\n");
for (String l : lines) {
    if (l.contains("https://youtu.be")) {
        yt = l;
    }
}

boolean condicion = true;
String letraCancion = resultado.toString();
String[] lines3 = resultado.toString().split(":\\n");
```

Fig 9. Obtener Música

La API que nosotros implementamos en nuestro proyecto fue la de login la cual implementamos en nuestro controlador para luego ser consumida por nuestro web service. A continuación, se puede observar el funcionamiento de nuestro proyecto.

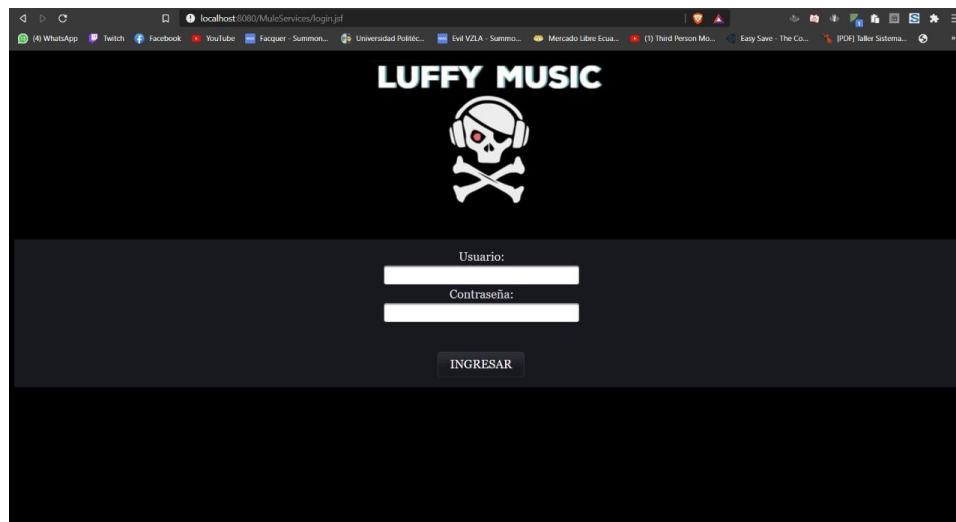



Fig 10. Pantalla Principal

Fecha: TOP 10

Lean



LETRA

¡WUH!, HOLA MI GIRL (FALSETTO BABEH)", "FALSETTO (READY), SAMMY, TOWY", """, "A LAS TRES SE VA PA'L BEAUTY, A LAS CUATRO VA PA'L MALL", "PA' ESO DE LAS CINCO ESTÁ A VAPOR", "SE HIZO LAS BOOBIES Y EL BOOTY, VISTE DE CHRISTIAN DIOR", "ACTIVA A TO'A SUS AMIGAS CON UN CALL", """, "LA BEBECITA BEBE LEAN Y BEBE WHISKY", "FUMA MARIHUANA Y TAMBIÉN SE METE PICKY", "SENDA BELLA CONA FRIKITONA, FRIKI,

Fig 11. Pantalla Canciones

Conclusiones

La implementación de este proyecto SOA nos ayudo mucho a mejorar nuestros conocimientos para así poder consumir servicios de una API en nuestro caso es la Shazam, a través del conector ESB nos facilita el consumo de las diferentes api e implementar las diversas funcionalidades que necesitamos para nuestros proyectos.

Recomendaciones

Revisar la documentación existente de Mule para la implementación de consumo rest y cómo utilizar el orquestador para implementarlo con java. Así mismo verificar la documentación que brinda el API para consumir correctamente las diferentes funcionalidades.