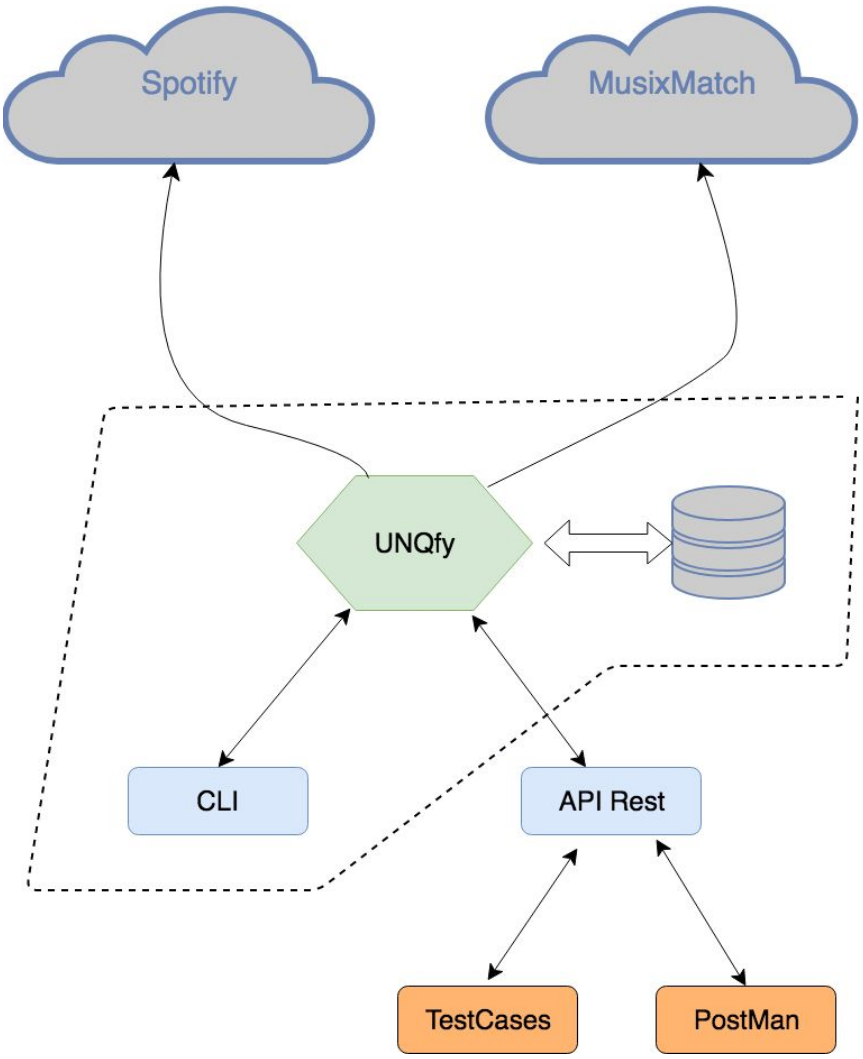
Visado 2 - REST / Asincronismo

Introducción

Basándose en su implementación de UNQfy (lo que esta encerrado en lineas punteadas) del visado anterior ahora ud. deberá agregar la siguitente funcionalidad:

1- Enriqueser su modelo con datos provenientes de Spotify y MusixMatch
2- Desarrollar una API rest para interactuar con UNQfy.



Fase 1 - Enriquecer el modelo

En el modelo UNQfy debe ser posible preguntar por todos los albumes de un artista dado. No es necesario que los álbumes tengan los temas, al menos debe estar el título. getAlbumsForArtist(artistName).

Spotify

Implemente un comando en UNQfy llamado *populateAlbumsForArtist(artistName)* . Cuando UNQfy recibe este comando debe consultar los albumes de dicho artista en Spotify, en base a los datos recibidos debe instanciar los álbumes correspondientes y asociarlos al artista, de manera tal que en la siguiente invocación a getAlbumsForArtist(artistName) se incluyan todos los álbumes recibidos de Spotify.

Debe implementar esta parte del enunciado utilizando requests (con callbacks o promesas). Para conectarse a Spotify es necesario pasar por un proceso de autenticación basado en OAuth. Se provee un snippet de código para facilitar esta tarea.

La API de Spotify se encuentra documentada en:
https://beta.developer.spotify.com/documentation/web-api/reference/

En particular usted deberá utilizar el siguiente entry point:
https://beta.developer.spotify.com/documentation/web-api/reference/artists/get-artists-albums/

G**enerar Credenciales para la API de Spotify:**

- Copiarse el archivo 'generateSpotifyCredentials.js' (provisto por la cátedra) en el mismo directorio donde se encuentra el código de su unqfy.
- Ejecutar: npm install --save express request
- Ejecutar: node generateSpotifyCredentials.js

Una vez ejecutado deberá:

- Hacer click en link impreso en la terminal
- Ese link lo llevará a loguearse en spotify (deberá crear una cuenta si no la tiene)
- Una vez logueado, volver a la terminal. Se habrá generado un archivo spotifyCreds.json con el access_token

Para poder enviar un request autorizado a spotify deberá enviar el string 'BEARER ' + access_token en el Header 'Authorization' en cada request. Utilizando request-promise:

```
const rp = require('request-promise');
const options = {
     url: 'ALGUN_ENDPOINT_DE_SPOTIFY'
     headers: { Authorization: 'Bearer ' + 'ACCESS_TOKEN' },
     json: true,
};
rp.get(options).then((response) => //hacer algo con response);
```

MusixMatch

MusixMatch es una aplicación (con una API Rest) que provee las letras (lyrics) de muchas canciones.

UNQfy ahora debe soportar pedirle la letra a un determinado track/tema.
Cuando un tema recibe el mensaje *getLyrics()* debe retornar el string con la letra de esa canción. Obviamente, la primera vez que se le envía ese mensaje a un track no tiene el lyric disponible, entonces debe ir a buscarlo a MusixMatch, almancenarlo en una variable de instancia y retornarlo.

MusixMatch utiliza *apikey* para validar la conexión. Se provee una API key válida para que la use.
La API de MusixMatch se encuentra documentada en este sitio.
https://developer.musixmatch.com/documentation

En particular, para resolver este ejercicio, ud utilizará el siguiente entrypoint:
https://developer.musixmatch.com/documentation/api-reference/track-lyrics-get

https://developer.musixmatch.com/documentation/api-reference/track-search

Nota: Tanto para obtener los albumes de un artista en Spotify, como para obtener la letra de un tema en MusixMatch, es necesario tener el *id* correspondiente.
Para ello las búsquedas en ambos servicios retornan listas de objetos con sus IDs.
Por ej. si se quiere obtener los álbumes del artista Charly García, es necesario primero hacer una búsqueda de artista por el string "Charly García" como resultado obtendrá una serie de objetos con sus IDs (y otros campos) como:

- Charly García (4923943242)
- Charly García y Sui Generis (32454534532)
- Charly M. García (423094832048)

De todos esos seleccionaremos de manera automática siempre el primer artista (o tema según corresponda).

Fase 2 - Desarrollo de la API Rest para UNQfy.

La API Rest de UNQfy debe permitir agregar, buscar y eliminar los objetos core del sistema: artistas, albums, tracks y playlists. **Para este visado, vamos a implementar la búsqueda, agregado y eliminado de Artistas y álbumes solamente,**

A continuación se describe la API. Debe respetar la definición para que los tests implementados con postman puedan ser utilizados.

Artistas:

Acción	**Endpoint**	**Argumentos en Request**	**Respuesta**

Agregar un artista	`POST /api/artists`	**`Body:`**
		`{`
		`'name': 'Guns n Roses' ,`
		`'country': 'USA'`
		`}`
			`Status Code: 200`
			`{`
			`"id": <artistId>`
			`"name": 'Guns n' Roses"`
			`"country":"USA",`
			`"albums": []`
			`}`

Obtener un artista	`GET /api/artists/<id>`		`Status Code: 200`
			`{`
			`"id": <artistId>`
			`"name": 'Guns n' Roses"`
			`"country":"USA",`
			`"albums":[<album1>, <album2> ]`
			`}`

Borrar un Artista	`DELETE /api/artists/<id>`		`Status Code: 200`
Buscar	`GET /api/artists`	**`Query:`** `name=<artistName>`	`Status Code: 200`

artistas			[<artista1>, <artista2>]

Albums:

Acción	**Endpoint**	**Argumentos en Request**	**Respuesta**

Agregar un album a un artista	POST /api/albums	**Body:**
		{
		artistId: artistId
		'name': 'Appetite For Destruction' ,
		year: 1987
		}
			Status Code: 200
			{
			"id": <albumId>
			"name": 'Appetite For Destruction",
			"year": 1987,
			"tracks": []
			}

Obtener un album	GET /api/albums/<id>		Status Code: 200
			{
			"id": <albumId>
			"name": 'Appetite For Destruction",
			"year": 1987,
			"tracks": [],
			}

Borrar un Album	DELETE /api/albums/<id>		Status Code: 200
Buscar albumes	GET /api/albums	**Query:** name="<albumName>"	Status Code: 200 [<album1>, <album2>]

NOTA:

- Las búsquedas (de albumes y artistas) no deben ser sensibles a mayúsculas. Es decir, si existe el album con nombre “ALBUM 1” y se busca el string “aLbu” debe retornar el album “ALBUM 1”.
- Cuando se obtienen los artistas. El campo albums del artista debe tener una lista de objetos albums como los retornados en el Endpoint GET /albums/<albumId>

Los código de error también forman parte de la API. Su implementación debe usar los códigos de error de la siguiente manera:

Acción	**Status Code Respuesta**	**Body Respuesta**

Se intenta agregar un artista/album duplicado	409	{
		status: 409,
		errorCode: "RESOURCE_ALREADY_EXISTS"
		}

Se intenta agregar un álbum a un artista inexistente	404	{
		status: 404,
		errorCode: "RELATED_RESOURCE_NOT_FOUND"
		}

URL invalida/inexistente	404	{
		status: 404,
		errorCode: "RESOURCE_NOT_FOUND"
		}

Borrar/Obtener un Artista/Album inexistente	404	{
		status: 404,
		errorCode: "RESOURCE_NOT_FOUND"
		}

Se envía un Json invalido en el Body	400	{
		status: 400,
		errorCode: "BAD_REQUEST"
		}

Falta un parametro en el JSON al agregar artista/Album	400	{
		status: 400,
		errorCode: "BAD_REQUEST"
		}

Fallo inesperado	500	{
		status: 500,
		errorCode: "INTERNAL_SERVER_ERROR"
		}

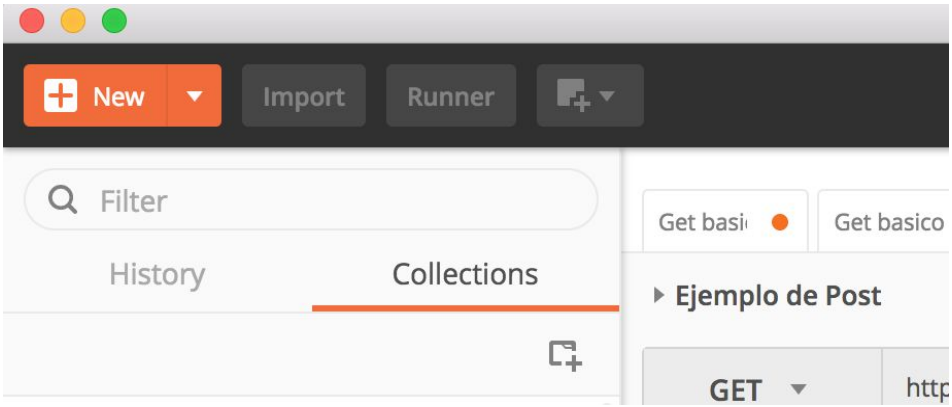
Notas/Aclaraciones:

- Deberá agregar un campo ID (numérico) a sus objetos en unqfy. Esto es, cada vez que se crea un objeto deberán asignarle un identificador único
- Recuerde persistir el objeto unqfy en las operaciones que modifican el mismo.
- Tenga en cuenta que para retornar un objeto de UNQfy como un JSON, probablemente quiera implementar el método toJSON en el objeto a devolver.

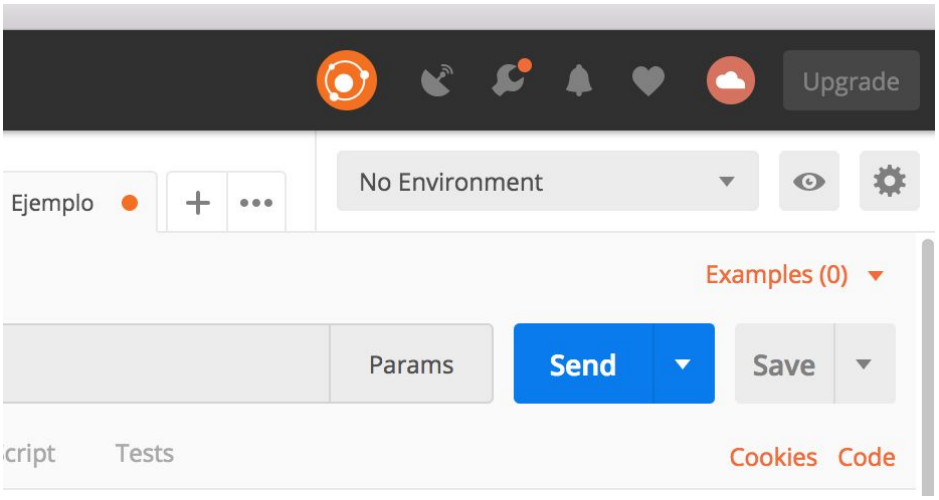
CORRIENDO LOS TESTS DE INTEGRACIÓN

Realice por única vez:

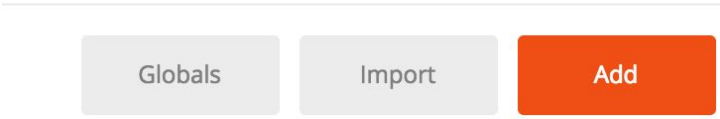
- Importe en postman la colección `UNQfy.postman_collection.json`



- Importe en postman el ambiente `UNQfy.postman_environment.json`.

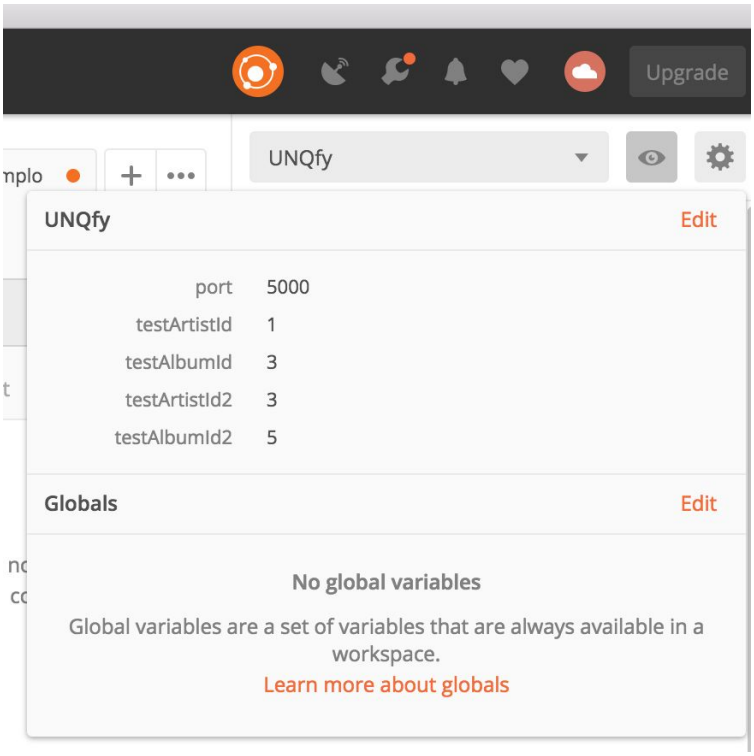


Click en el boton de stettings del environment



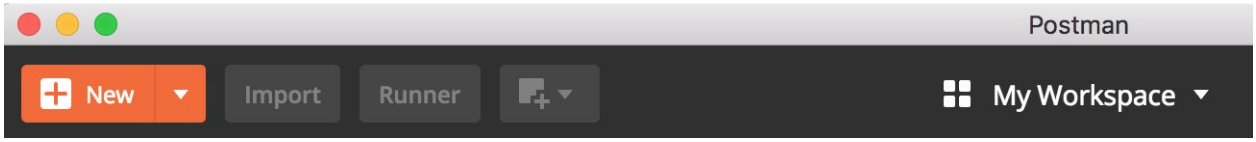
Botón import, seleccionar el archivo y luego seleccionar ese environment con como el activo.

- En el ambiente UNQfy importado deberá configurar el puerto donde tiene corriendo su servicio (variable port).

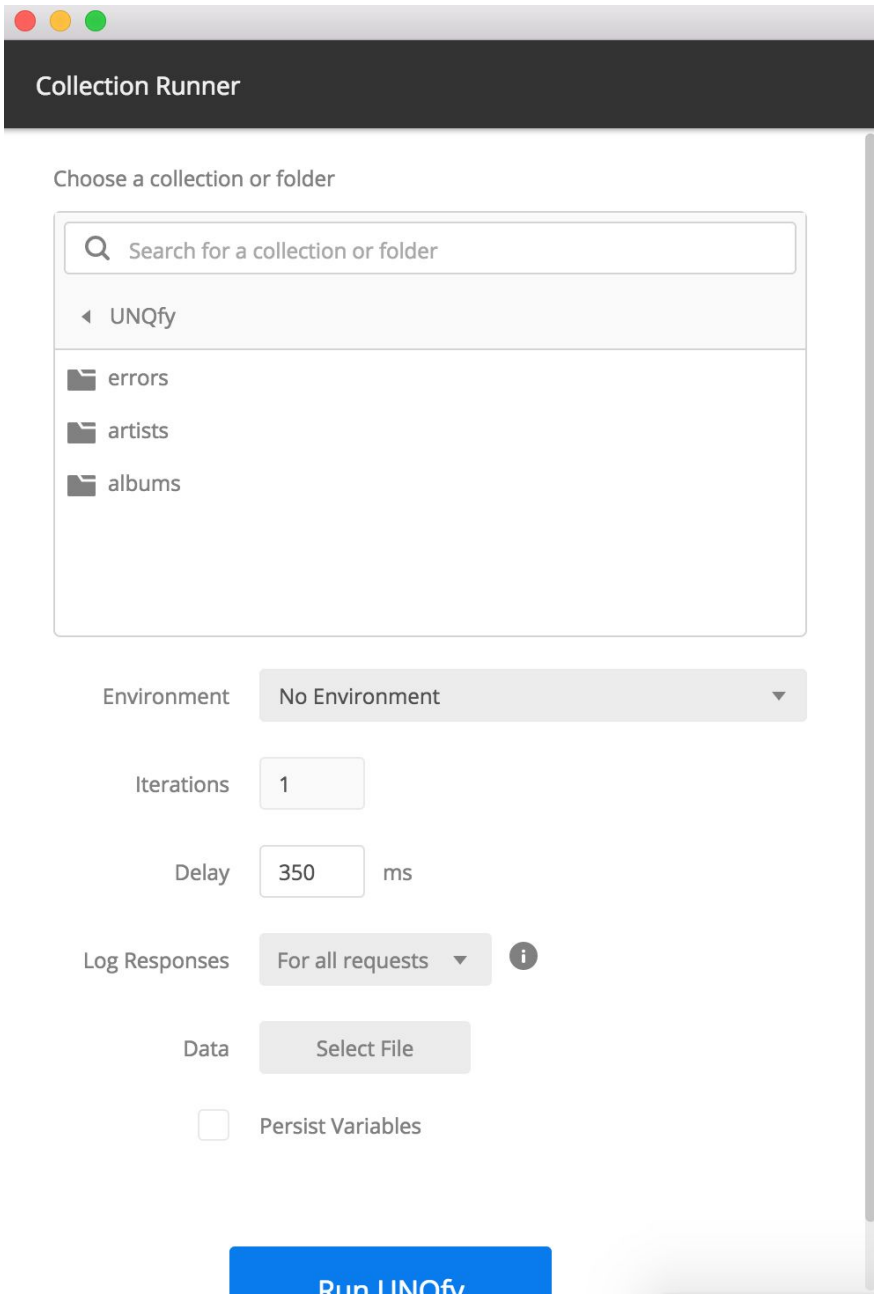


Cada vez que quiera correr los tests:

- Borre su archivo de estado en UNQfy y reinicie el servicio.
- Click en el boton “runner” en postman



- Elija la colección UNQfy.
- En el campo delay, ingrese “350 ms”.



NOTAS/Aclaraciones:

- Los tests modifican el estado de unqfy, es por eso que deberá borrar su estado antes de correr cada test.