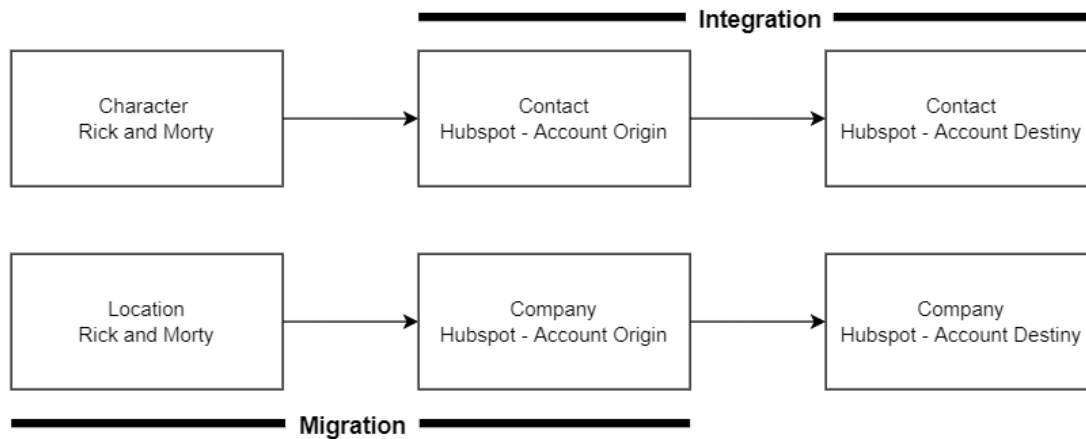# Backend Developer Test 2022

Context

The back-end developer team performs regular migrations and integrations between third-party platforms to Hubspot. The difference between migration and integration is that migrations send data from one platform to another only once. In case of modifications after migration, it will not be reflected in the new platform; In contrast, integrations are a continuous connection between platforms that constantly allows changes made on one platform to be seen on the other.

In our example, we have a public API platform to get the objects' information. This platform is the Rick and Morty API. We will work with two objects: the first is the character that will be homologated by the contact object in Hubspot; the other is the location that will be homologated by the company object in Hubspot.

Exercise

In the **Rick and Morty API,** you will find the different **characters and locations** that have appeared in the series. One of our clients wants to carry out a Backend project with this API. First, you want to **migrate** all characters whose ID is a prime number, plus ID number 1, which is dear Rick Sanchez. In addition to all the locations that are associated with these characters. To do this, our engineering team has mapped the properties and homologation, finding that the **characters** will be **contacts** and the **locations** will be **companies** on the **Hubspot platform**. Also, as a security measure, our client wants to have a **mirror account** of the first one. To do this, we will perform an **integration** between two Hubspot accounts. This integration will be one way only and based on **creating or updating** contacts in the source account to the mirror account.

Characteristics:

- **Platform to Migrate:** Rick and Morty API
- **Source Platform:** Source_Platform_Developer_#
- **Mirror Platform:** Mirror_Platform_Developer_#

NOTE: We will give them access to the accounts to your email. Be sure to check your inbox or spam folder for the invitation email to join the accounts.

Migration Part

Below you will find the conditions to do the migration part:

1. Using the **Rick and Morty API**, **migrate character #1** and all **characters whose ID is a prime number** as **contacts** on the Hubspot platform. The properties mapping for this object is:

Rick And Morty Character		Hubspot Contacts			
Key	*Type*	*Property Name*	*Internal Name*	*Type*	*Values*
id	int	Character ID	character_id	Single Line Text	N/A
name	string	First Name	firstname	Single Line Text	N/A
name	string	Last Name	lastname	Single Line Text	N/A
status	string	Status Character	status_character	Dropdown	Alive Dead unknown
species	string	Character's Species	character_species	Single Line Text	N/A
gender	string	Character's Gender	character_gender	Dropdown	Female Male Genderless unknown

2. You will find a key with the character's **location** in each character. Create this location like a **Company** on the Hubspot Platform. The properties mapping for this object is:

Rick And Morty Location		Hubspot Companies			
Key	*Type*	*Property Name*	*Internal Name*	*Type*	*Values*
id	int	Location ID	location_id	Single Line Text	N/A
name	string	Company name	name	Single Line Text	N/A
type	string	Location's Type	location_type	Single Line Text	N/A

dimension	string	Dimension	dimension	Single Line Text	N/A
created	string	Creation Date	creation_date	Date Picker	N/A

3. Create the corresponding **associations** between characters (contacts) and locations (companies).

Integration Part

1. Create **two endpoints**. One of them will be in charge of **creating or updating contacts** on the mirror platform. The second will do the same process **(creation and update) for the companies**.

2. Two workflows were created in the source account to detect this process, and the triggers for each case were configured.

3. **Modify the workflows** created in the source account in the action "**Send a webhook**" in the "**Webhook URL**" option by the **endpoints URLs**.

What We Need to Do

1. Create a **private app** in each account that allows you to read and write the contact and company objects.

2. **Create an API** that allows you to carry out the proposed exercise (data migration and integration).

3. **Deploy the API to a server**. You can use [Heroku](), [Render](), [Qoddi](), [Deta Cloud](), or any free cloud service of your choice.

4. **Configure endpoint URLs** in the workflows configured in the source account.

5. **Test the solution.**

Deliverables

1. **Diagram** where you explain the migration and integration process.

2. A **git repository** with a well-documented Readme file explains how we should run the solution.

Important Notes:

1. Review the attached documentation and videos.

2. Use Node.js as the Backend Programming Language.

3. If you get stuck, please don't hesitate in asking to ask by sending us a video showing us how far you got and what is the problem you are encountering.

Video

Please watch the video for a better context of the test

https://www.loom.com/share/c26599ec15ff425a94ebb732838fadaa.

Resources:

- **Private Apps:** https://developers.hubspot.com/docs/api/private-apps

- **Hubspot API**: https://developers.hubspot.com/docs/api/overview
    - **CRM API - Contacts:** https://developers.hubspot.com/docs/api/crm/contacts

    - **CRM API - Companies:** https://developers.hubspot.com/docs/api/crm/companies

    - **CRM API - Associations:** https://developers.hubspot.com/docs/api/crm/associations

- **HubSpot Node.js Client API**: https://www.npmjs.com/package/@hubspot/api-client

- **Rick and Morty API:** https://rickandmortyapi.com/documentation/#rest

- **Rick and Morty js Client API:** https://javascript.rickandmortyapi.com/