Magdalena Larmet
Salesforce
Developer

Legarant

DEPLOYER UNE APPLICATION
SALESFORCE

Magdalena Larmet
Salesforce
Developer

DEPLOYER UNE APPLICATION SALESFORCE

Sommaire

Rappels du projet

Rappel du projet

22-08-2023	18-09-2023	À DÉFINIR
Date démarrage	Date de fin	Coût du projet

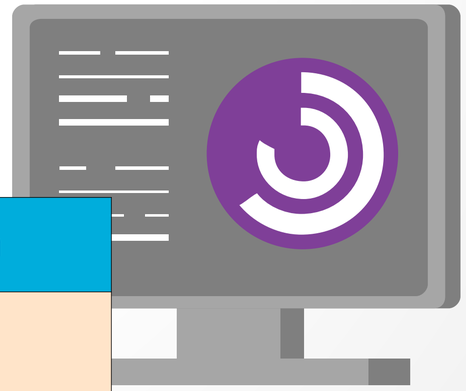
BESOINS EXPRIMÉS

Suite au rachet la société AXG, LEGARANT souhaite conserver son CRM Salesforce, et intégrer les données importantes du CRM AXG à celui-ci.

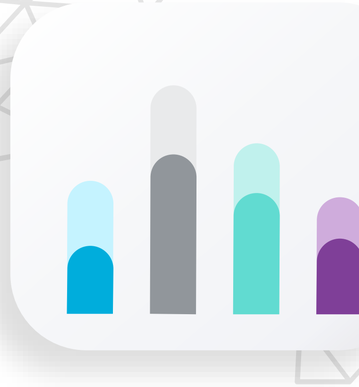
EXIGENCES DU PROJET

- ✓ Respecter les règles de gestion de la société
- ✓ Effectuer les appels API en respectant les standards Salesforce
- ✓ Utiliser une PAAS parmi les suivantes:
  - Heroku,
  - Azure
  - AWS

Rappel des règles de gestion



Règle de Gestion	Salesforce	@Http	Postman
Création d'un contact : quand un contact est créé dans CRM AXG, il faut vérifier si l'email existe déjà. i. S'il existe, mettre l'id de Salesforce dans le CRM AXG ii. S'il n'existe pas, créer un nouveau contact dans Salesforce et prendre l'id.	createNewContact()	httpPost	CreateContact
Modification d'un contact : quand un champ est modifié sur le Contact dans le CRM AXG, le changement doit être envoyé	updateContact()	httpPatch	UpdateContact
Suppression d'un contact : un contact ne doit pas être supprimé dans Salesforce, il faut le mettre en désactivé	deactivateContact()	httpDelete	DisableContact
Création d'un contrat : quand un contrat est créé dans le CRM AXG, les informations sont envoyées vers Salesforce et l'Id de Salesforce est sauvegardé.	createNewContract()	httpPost	CreateContract
Modification d'un contrat : à chaque modification du contrat, il faut mettre à jour le contrat dans Salesforce en se basant sur l'Id de Salesforce.	updateContract()	httpPatch	UpdateContract



Étapes du projet

STEP 01

ENVIRONNEMENT
SALESFORCE

- Création de l'application Connectée
- Création de Champs Personnalisés
- Flow pour gérer External ID
- Création des classes @RestRessource
- Test des methodes Rest
- Autoriser l'org dans VSC

STEP 02

ENVIRONNEMENT
POSTMAN

- Connecter avec l'org Salesforce (Consumer Key, Consumer Secret)
- Création d'une Collection
- Obtention du Token
- Effectuer des requêtes demandées

STEP 03

ENVIRONNEMENT
HEROKU

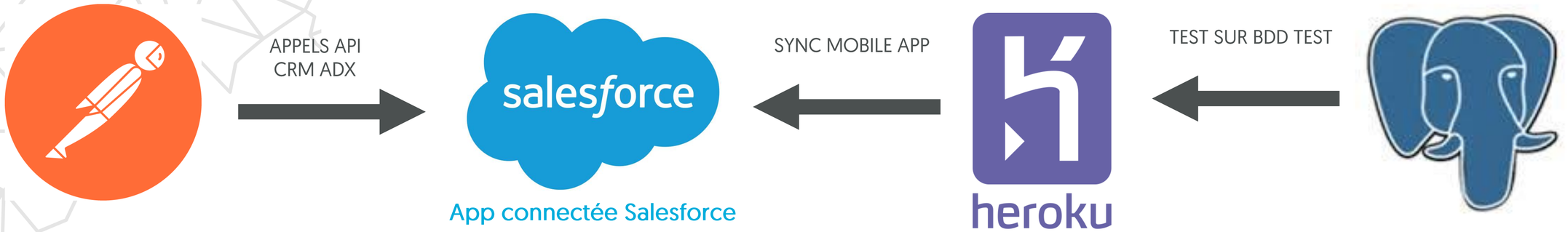
- Création de l'application Heroku
- Connecter avec l'organization SF
- Déployer sur Github et Connecter le Github créé pour le projet

STEP 04

ENVIRONNEMENT
PGADMIN

- Création d'un server en renseignant les Crédentials Heroku
- Création de requêtes tests avant le déploiement

Étapes du projet



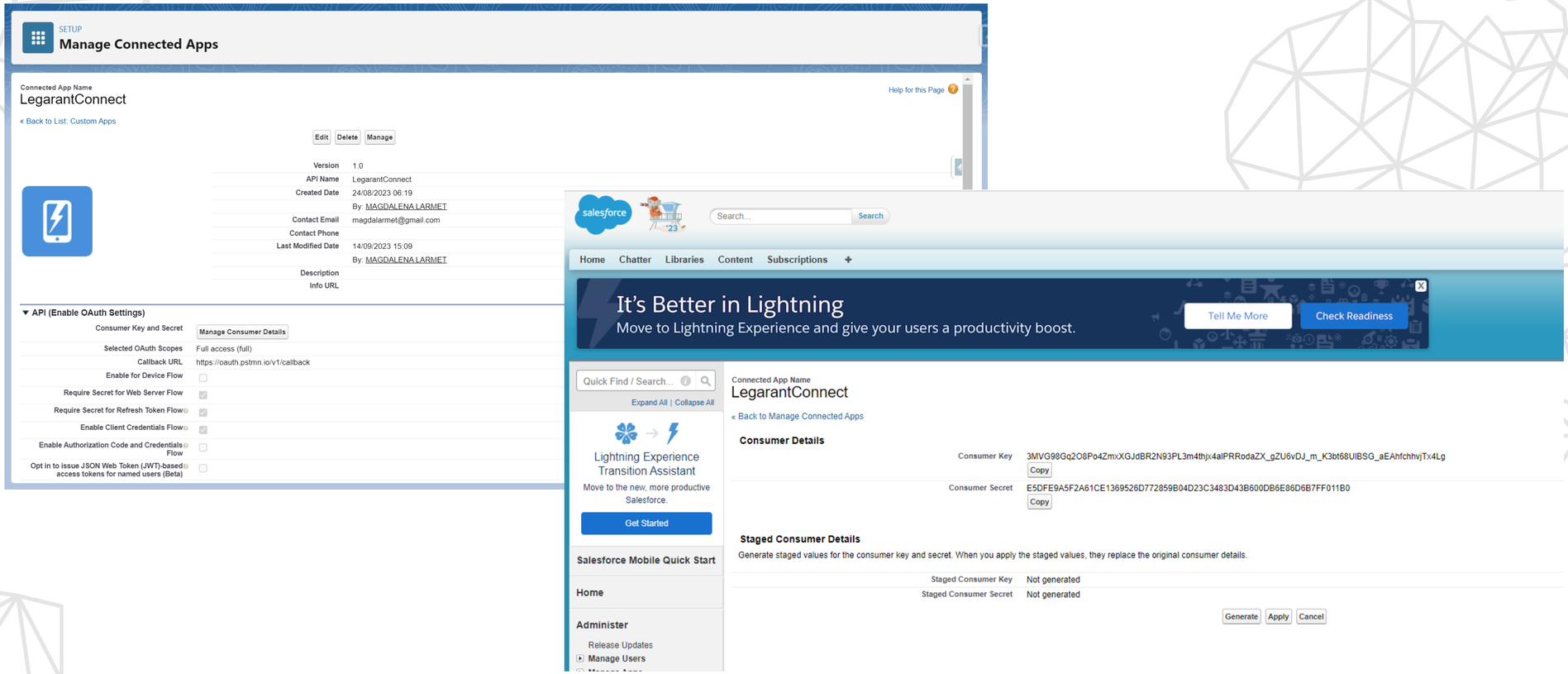
Environnement
Salesforce

Environnement Postman

Connecter Salesforce et Postman

Consumer Key & Consumer Salesforce

LegarantConnect



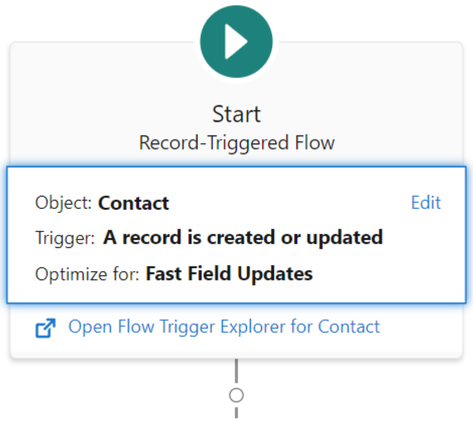
Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

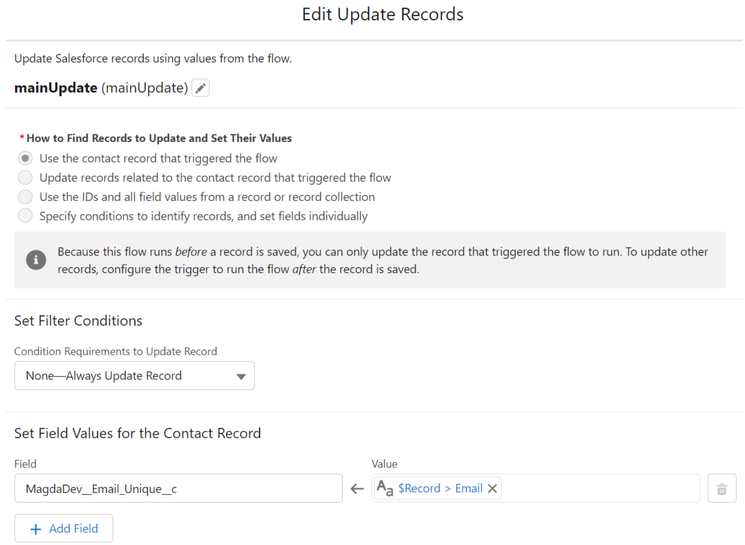
Environnement Salesforce

Flow Builder

Gestion de l'email Unique

À la Création et à l'Update





Le champ email se rempli automatiquement avec l'email du contact qui correspond à l'external ID

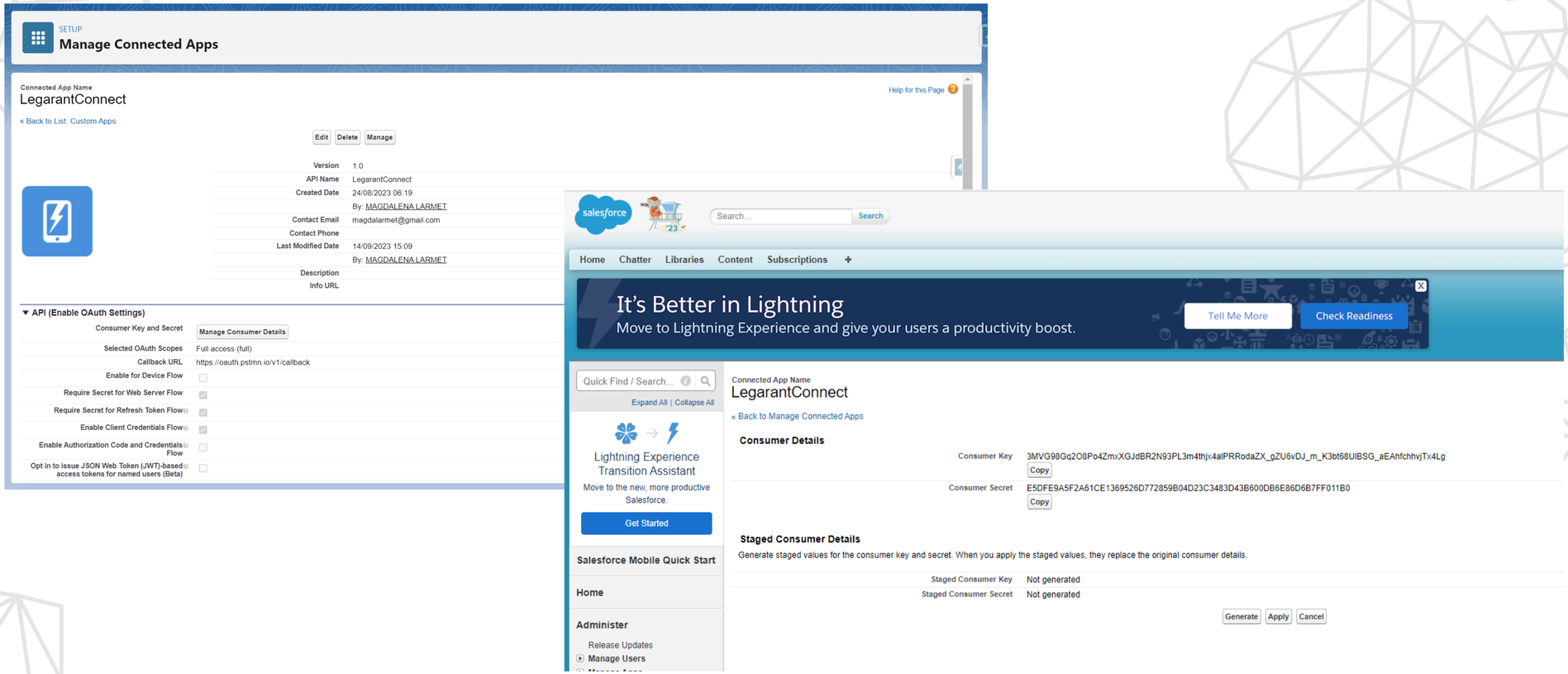
Projet mené pour la Société LEGARANT

Environnement Salesforce

App Manager

Connected App

LegarantConnect



Projet mené pour la Société LEGARANT

Environnement Salesforce

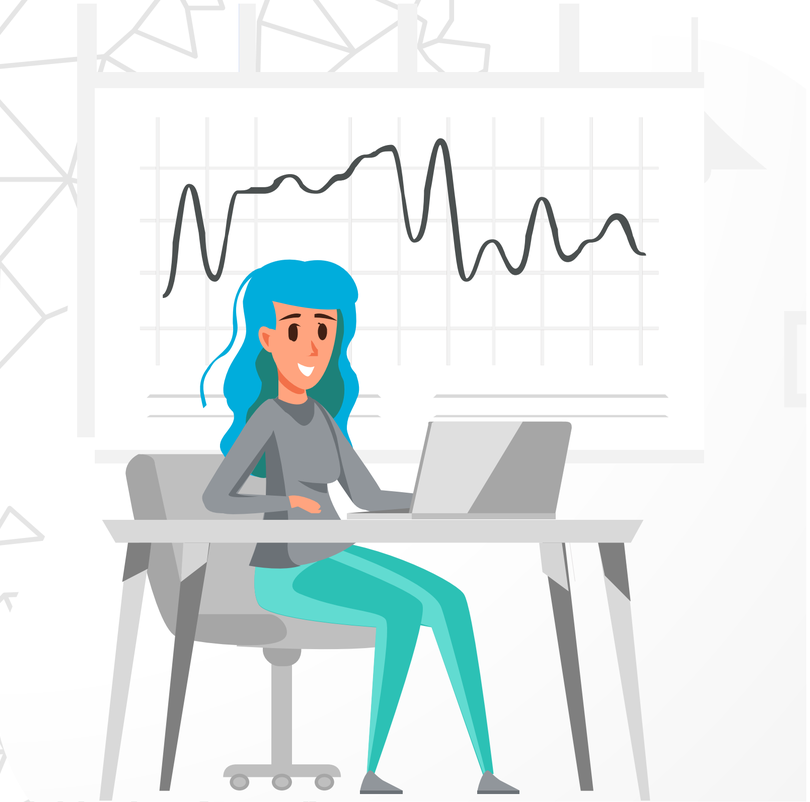
Le code @RestRessources

ContactService

```
@RestResource(urlMapping='/Contact/*')
global with sharing class ContactService {

    @HttpGet
    global static Contact getContactById() {
        RestRequest request = RestContext.request;
        // Get the contactId from the end of the URL
        String contactId = request.requestURI.substring(
                request.requestURI.lastIndexOf('/') + 1);
        //Id contactId = RestContext.request.params.get('Id');

        Contact result = [
                SELECT Salutation, FirstName, LastName, Email, Phone
                FROM Contact
                WHERE Id = :contactId
        ];
        return result;
    }

    @HttpPost
    global static String createNewContact() {
        RestRequest request = RestContext.request;
        Map<String, Object> requestResult = (Map<String, Object>) JSON.deserializeUntyped(request.requestBody.toString());
        List<Object> contacts = (List<Object>) requestResult.get('records');
        List<Contact> contactsToInsert = new List<Contact>();
        for (Object contactObject : contacts) {
            Map<String, Object> parsedContact = (Map<String, Object>) contactObject;
            Contact thisContact = new Contact(
                    Active__c = true,
                    FirstName = (String) parsedContact.get('FirstName'),
                    LastName = (String) parsedContact.get('LastName'),
                    Email = (String) parsedContact.get('Email'),
                    Email_Unique__c = (String) parsedContact.get('Email')
            );
            contactsToInsert.add(thisContact);

        }

        Database.upsert(contactsToInsert, Contact.Email_Unique__c, true);

        Map<String, String> stringMap = new Map<String, String>();
        for (Contact contact : contactsToInsert) {
            if (contact.Id != null) {
            stringMap.put(contact.Id, contact.Email_Unique__c);
            }
        }
        return JSON.serializePretty(stringMap);
    }
```

Environnement Salesforce

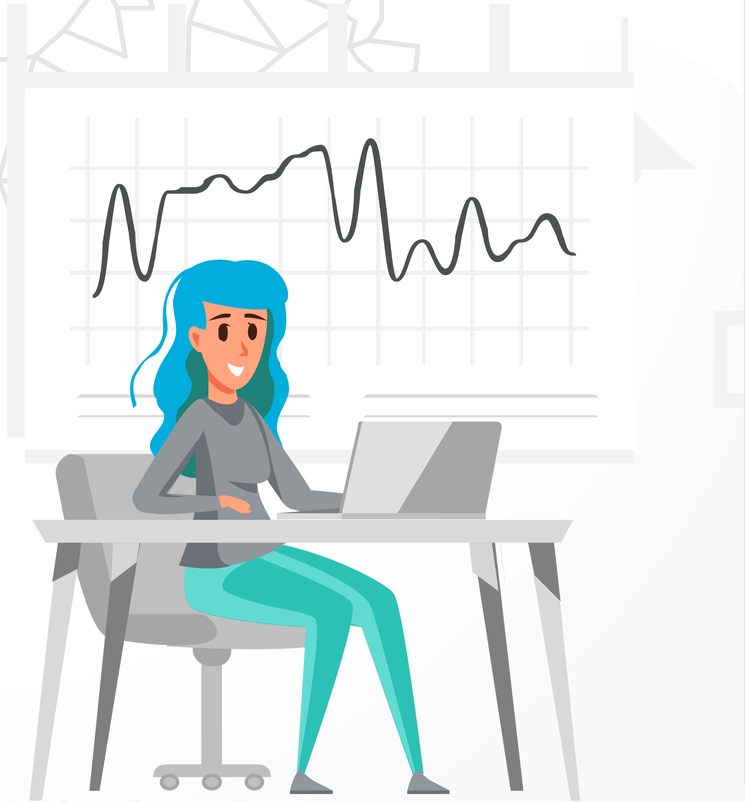
Le code @RestRessources
ContactService



```
@HttpPatch
global static Id updateContact() {

    // Add the Contact Id as a parameter in the URL
    RestRequest request = RestContext.request;
    String contactId = request.requestURI.substring(request.requestURI.lastIndexOf('/') + 1);

    // Check if contact exist
    List<Contact> matchingContacts = [SELECT Id FROM Contact WHERE Id = :contactId];
    if (matchingContacts.isEmpty()) {
        // Case : ID does not exist
        // return an error
        return 'Contact with ID ' + contactId + ' does not exist';
    }

    Contact thisContact = matchingContacts[0];
    //convert JSON data from request
    Map<String, Object> params = (Map<String, Object>) JSON.deserializeUntyped(request.requestBody.toString());
    //update fields on object
    for (String fieldName : params.keySet()) {
        thisContact.put(fieldName, params.get(fieldName));
    }

    update thisContact;
    return thisContact.Id;
}
```

Environnement Salesforce

Le code @RestRessources
ContactServiceTest

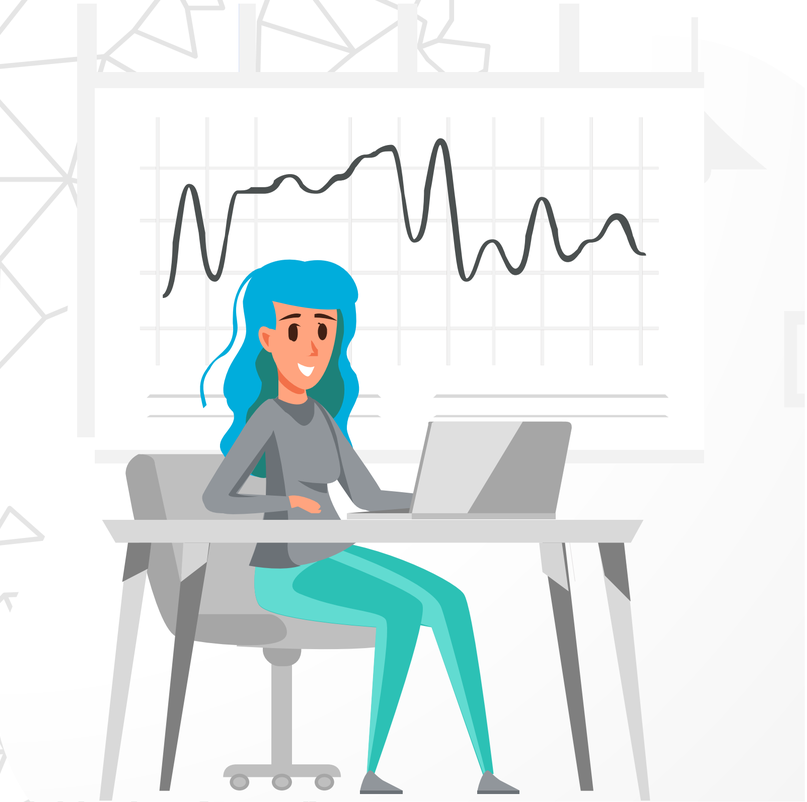


```
@isTest
public class ContactServiceTest {
    private static String contactFirstName = 'Magda';
    private static String contactLastName = 'Test';

     @testSetup
    static void setupForTest(){

    }

    @isTest static void testGetContactById() {
        List<Contact> contactListForThisTest = TestDataFactory.createContacts(2);
        Id recordId = contactListForThisTest.get(0).Id;
         Test.startTest();
        // Set up a test request
        RestRequest request = new RestRequest();
        request.requestUri ='https://legarant4-dev-ed.develop.lightning.force.com/services/apexrest/Contact/'+ recordId;
        request.httpMethod = 'GET';
        RestContext.request = request;

        // Call the method to test
        Contact thisContact = ContactService.getContactById();
        Test.stopTest();

        // Verify results
        System.assert(thisContact != null);
    }
```

Environnement Salesforce

Le code @RestRessources

ContractService



```
    @HttpPost
global static String createNewContract() {
    RestRequest request = RestContext.request;
    Map<String, Object> requestResult = (Map<String, Object>) JSON.deserializeUntyped(request.requestBody.toString());
    List<Object> contracts = (List<Object>) requestResult.get('records');
    List<Contract> contractsToInsert = new List<Contract>();

    for (Object contractObject : contracts) {
        Map<String, Object> parsedContract = (Map<String, Object>) contractObject;
        Contract thisContract = new Contract(
               Name = (String) parsedContract.get('Name'),
                Status =  (String) parsedContract.get('Status'),
                AccountId =  Id.valueOf((String) parsedContract.get('AccountId')),
                StartDate = Date.valueOf((String) parsedContract.get('StartDate')),
                ContractTerm = Integer.valueOf((String) parsedContract.get('ContractTerm'))
        );
        contractsToInsert.add(thisContract);
    }
    Database.upsert(contractsToInsert, Contract.Id, true);

    Map<String, Object> stringMap = new Map<String, Object>();
    for (Contract contract : contractsToInsert) {
        if (contract.Id != null) {
            stringMap.put(contract.Id, contract.Name);
        }
    }
    return JSON.serializePretty(stringMap);
}
```

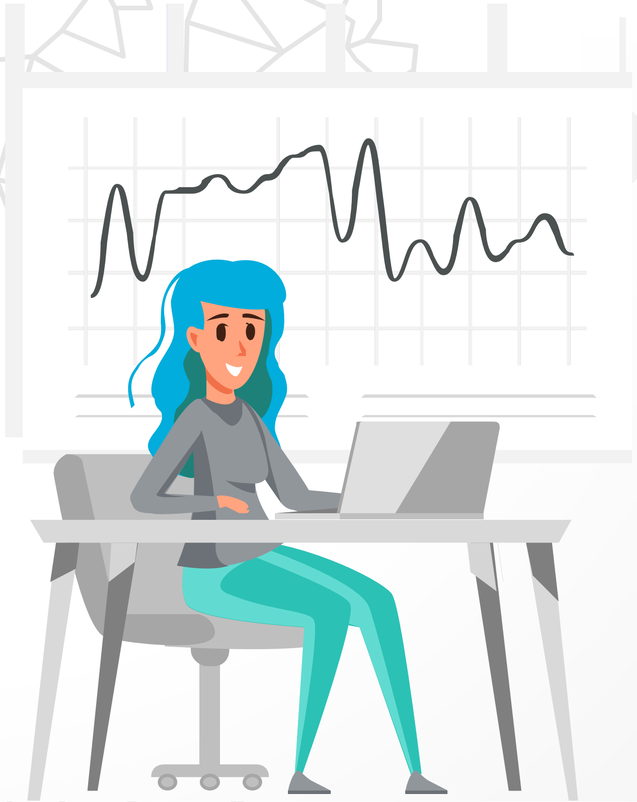
Environnement Salesforce

Le code @RestRessources
ContractService

```
@HttpPatch
global static Id updateContract(){
    // extract the Contract Id as a parameter in the URL
    RestRequest request = RestContext.request;
    String contractId = request.requestURI.substring(request.requestURI.lastIndexOf('/')+1);

    // Check if contact exist
    List<Contract> matchingContracts = [SELECT Id, AccountId FROM Contract WHERE Id = :contractId];
    if (matchingContracts.isEmpty()) {
        // Case : ID does not exist
        // return an error
        return 'Contract with ID ' + contractId + ' does not exist';
    }

    Contract thisContract = matchingContracts[0];
    //convert JSON data from request
    Map<String, Object> requestResult = (Map<String, Object>) JSON.deserializeUntyped(request.requestBody.toString());

    //update fields on object
    for (String fieldName : requestResult.keySet()) {
        thisContract.put(fieldName, requestResult.get(fieldName));
    }

    update thisContract;
    return thisContract.Id;
}
```

Environnement Salesforce

Le code @RestRessources
ContractServiceTest

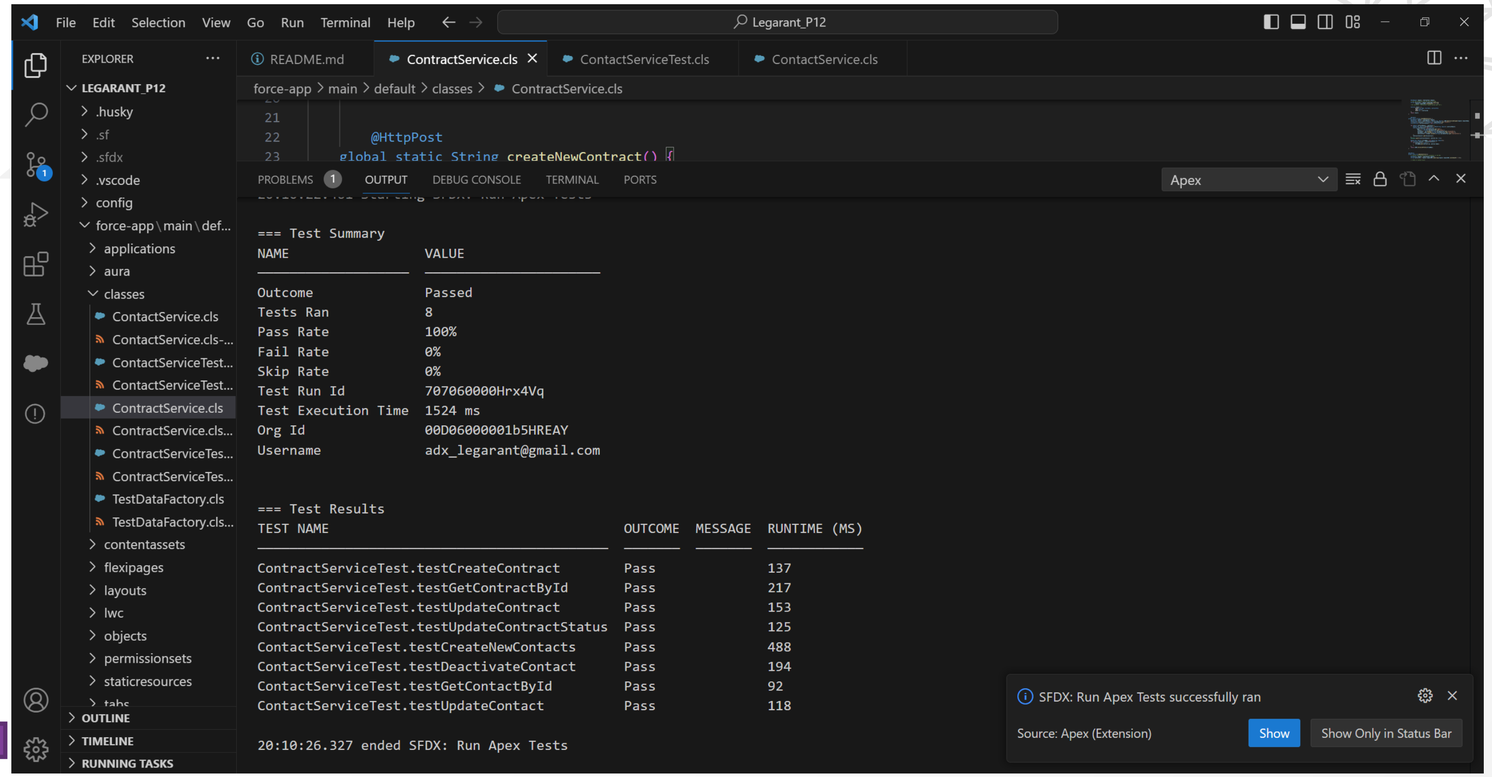


```
@isTest

public class ContractServiceTest {
 @testSetup
   static void setupForTest(){

    }

    @isTest static void testGetContractById() {
        Id contractForThisTest = TestDataFactory.createAccountAndContractRecord();
        Id recordId = contractForThisTest;
         Test.startTest();
        // Set up a test request
        RestRequest request = new RestRequest();
        request.requestUri ='https://legarant-b-dev-ed.develop.lightning.force.com/services/apexrest/Contact/'+ recordId;
        request.httpMethod = 'GET';
        RestContext.request = request;

        // Call the method to test
        Contract thisContract = ContractService.getContractById();
        Test.stopTest();

        // Verify results
        System.assert(thisContract != null);
    }

     @isTest
    static void testCreateContract() {
```

Environnement Salesforce

Le code @RestRessources

Résultat des Test passés

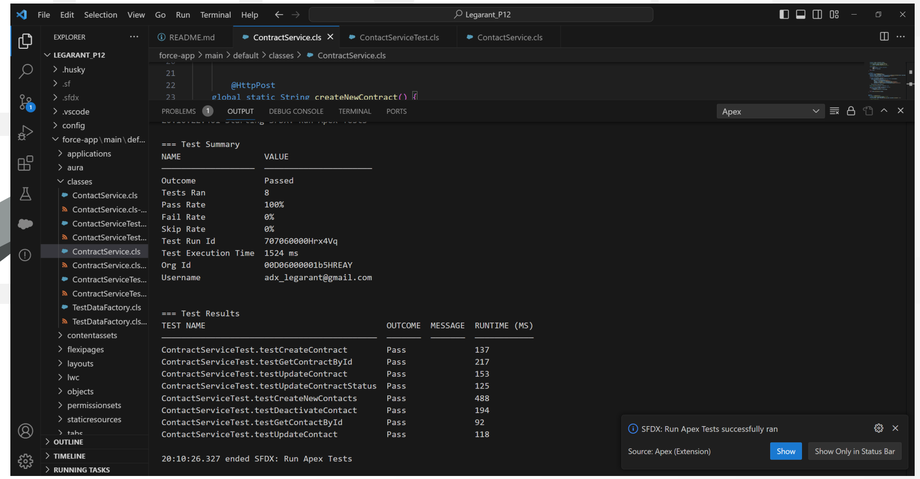


Environnement Salesforce

Le code @RestRessources

Couverture du code

Overall Code Coverage		
Class	Percent	Lines
Overall	**93%**	
ContactService	97%	46/47
ContractService	97%	38/39
TestDataFactory	83%	25/30



2

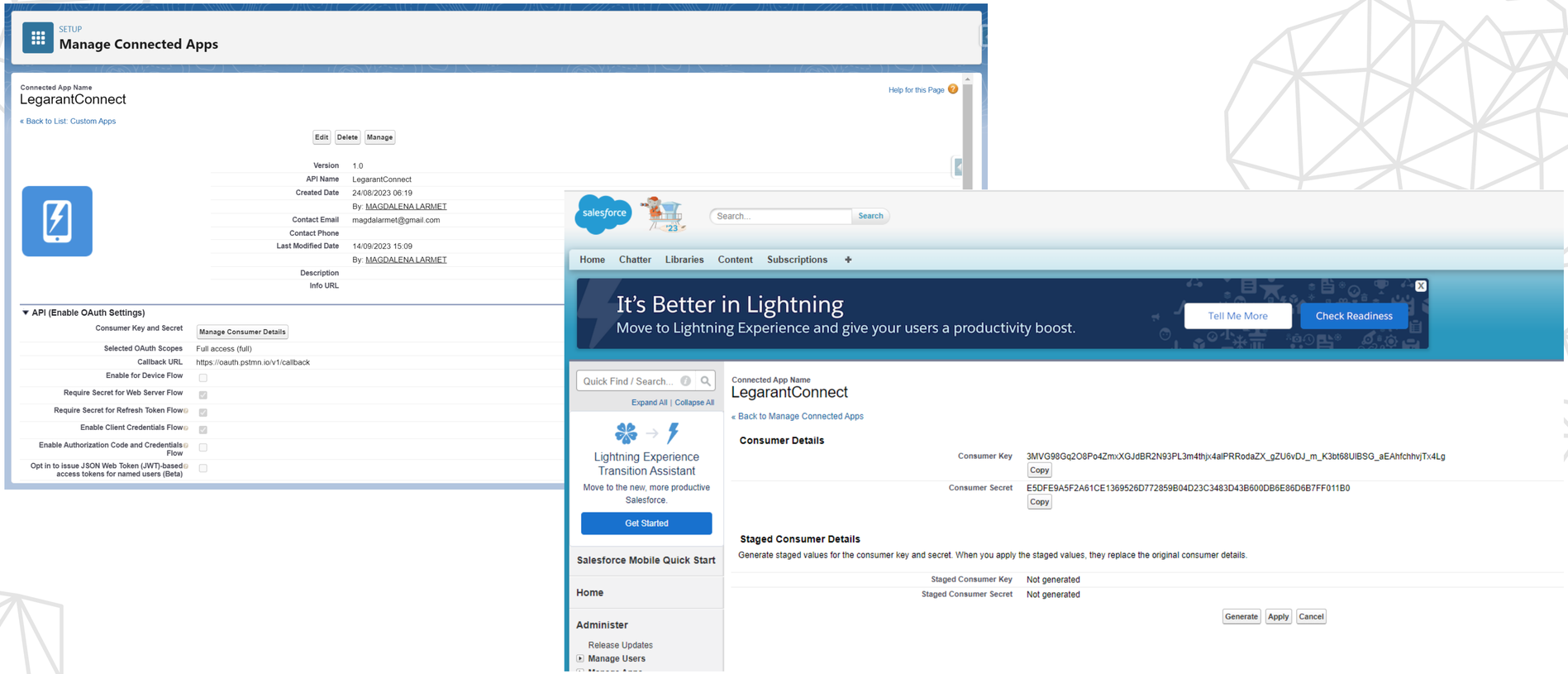
Environnement
Postman

Environnement Postman

Appels d'API

Identifiants de connexion

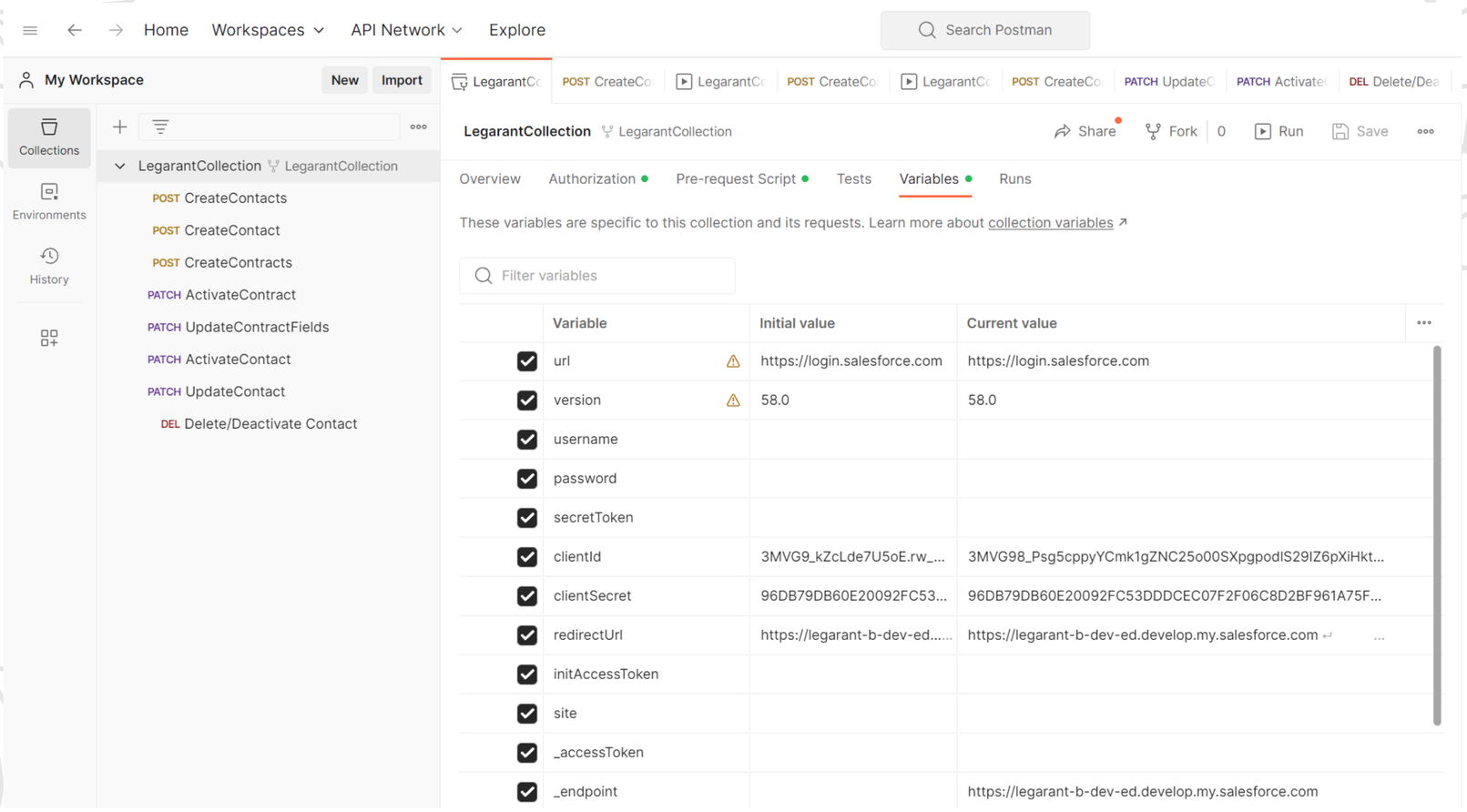
LegarantConnect



Environnement Postman

Application Salesforce

Création des variables



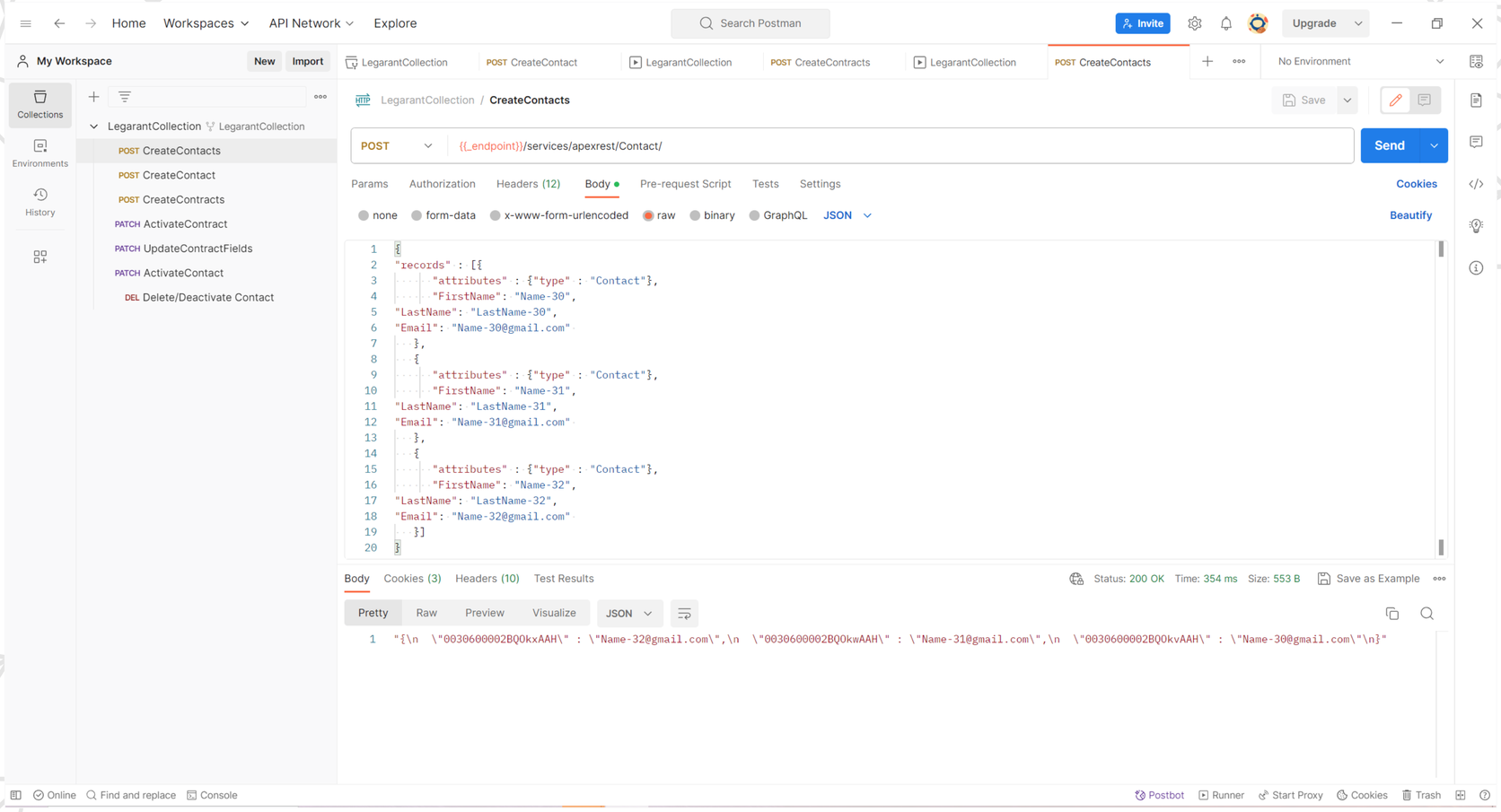
Projet mené pour la Société LEGARANT

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Environnement Postman

La collection Legarant

Création de contacts



Projet mené pour la Société LEGARANT

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Environnement Postman

La collection Legarant

Mise à jour de contacts

PATCH {{_endpoint}}/services/apexrest/:SOBJECT_API_NAME/:RECORD_ID

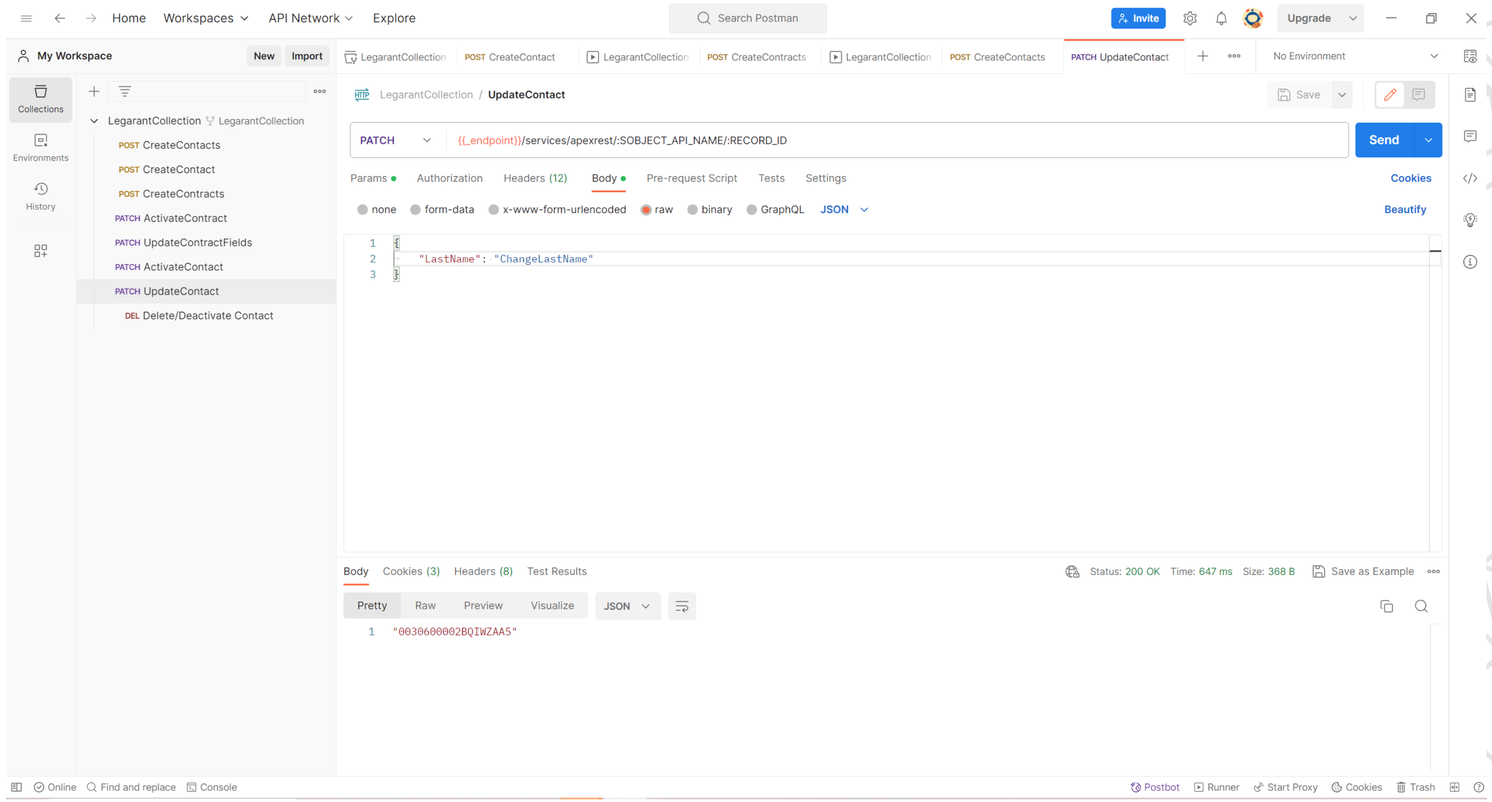
Params • Authorization Headers (12) Body • Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Path Variables

Key	Value
SOBJECT_API_NAME	Contact
RECORD_ID	0030600002BQIWZAA5



Projet mené pour la Société LEGARANT

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Environnement Postman

La collection Legarant

Suppression de contacts

LegarantCollection / **Delete/Deactivate Contact**

DELETE {{_endpoint}}/services/apexrest/:SOBJECT_API_NAME/:RECORD_ID

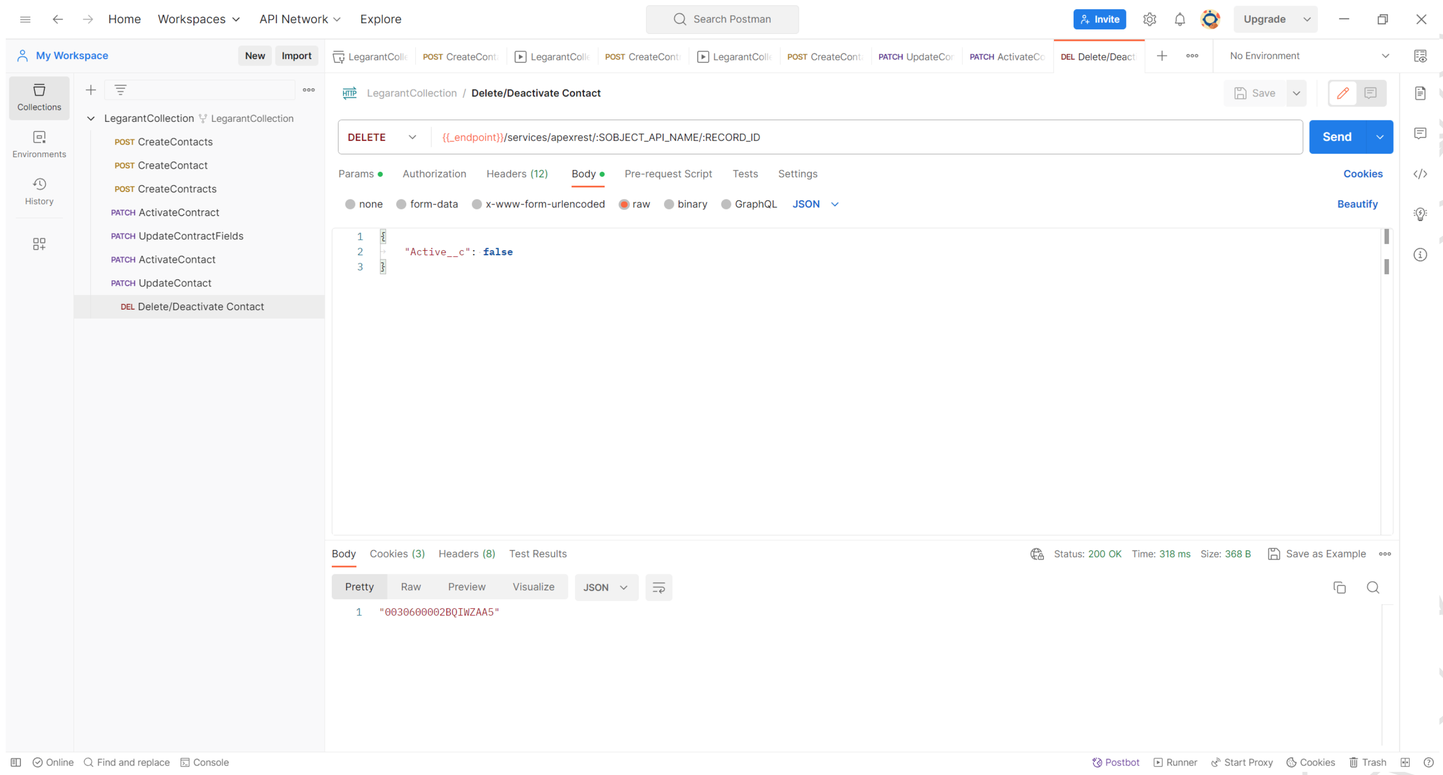
Params • Authorization Headers (12) Body • Pre-request Script Tests Settings

Query Params

Key	Value
Key	Value

Path Variables

Key	Value
SOBJECT_API_NAME	Contact
RECORD_ID	0030600002BQIWZAA5



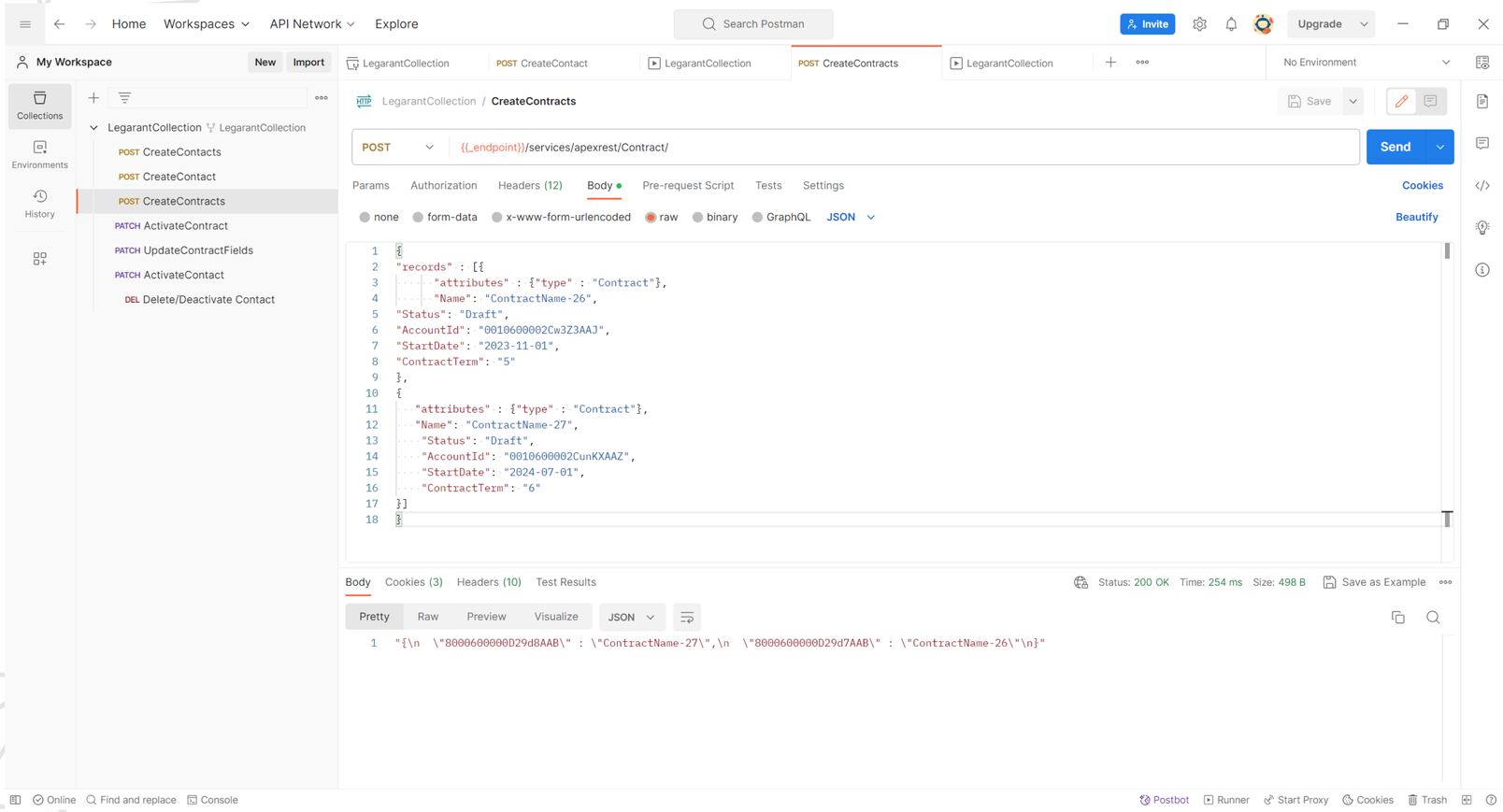
Projet mené pour la Société LEGARANT

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Environnement Postman

La collection Legarant

Création de contrats



Projet mené pour la Société LEGARANT

Magdalena Larmet | DEVELOPPEUR **SALESFORCE**

Environnement Postman

La collection Legarant

Mise à jour de contrats

PATCH {{_endpoint}}/services/apexrest/:SOBJECT_API_NAME/:RECORD_ID

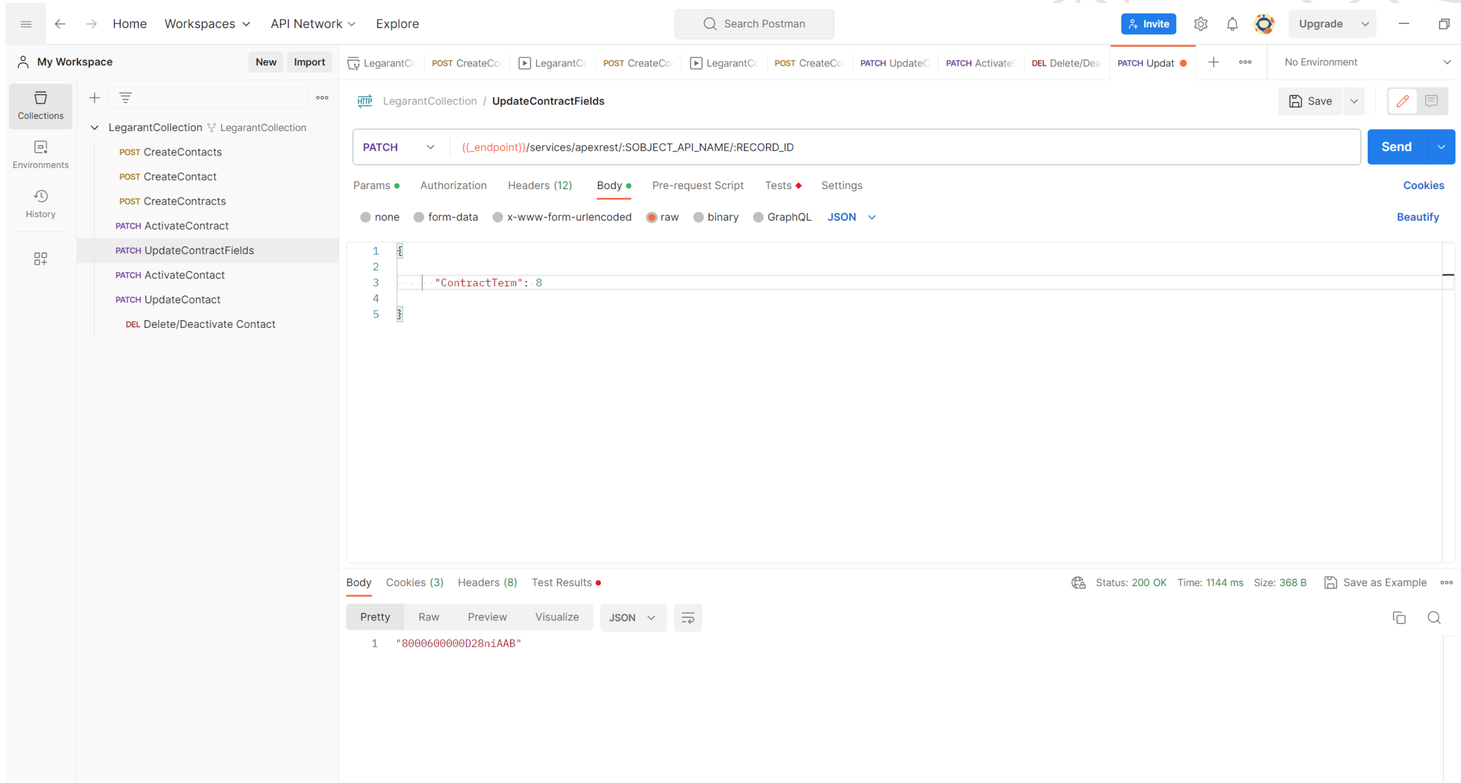
Params • Authorization Headers (12) Body • Pre-request Script Tests • Settings

Query Params

	Key	Value
	Key	Value

Path Variables

	Key	Value
	SOBJECT_API_NAME	Contract
	RECORD_ID	8000600000D28niAAB



Projet mené pour la Société LEGARANT

3

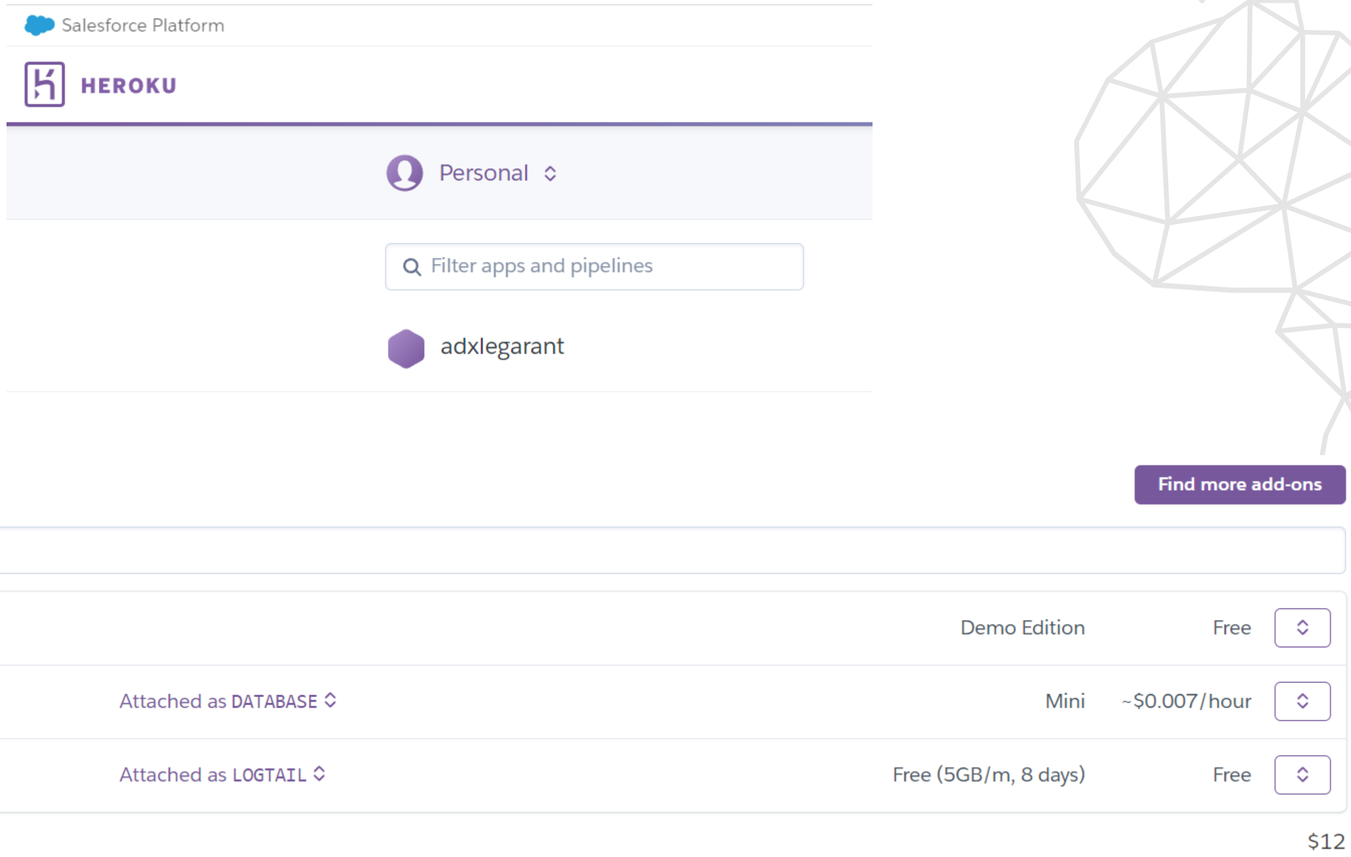
Environnement

Heroku

Environnement Heroku

Paramétrage de l'outil

Création de l'application & ajout des add-ons



Environnement Heroku

Heroku Connect

Mapping des champs

Heroku Connect

Activity

Past Hour

No data available for the selected time period

Mapped Fields

Field Name	Label	Type	Length	Indexed
AccountId	Account ID	reference to: Account	18	
Active__c	Active	boolean	0	
CreatedDate	Created Date	datetime	0	
Email	Email	email	80	Yes
Email_Unique__c	Email Unique	email	80	Yes
FirstName	First Name	string	40	
Id	Contact ID	id	18	Yes
IsDeleted	Deleted	boolean	0	
LastName	Last Name	string	80	
MasterRecordId	Master Record ID	reference to: Contact	18	
Name	Full Name	string	121	
OwnerId	Owner ID	reference to: User	18	
ReportsToId	Reports To ID	reference to: Contact	18	
SystemModstamp	System Modstamp	datetime	0	Yes

Environnement Heroku

Heroku Connect

Mapping des champs Contact

Heroku Connect

MAPPED FIELDS	FREQUENCY	LAST CHECKED SALESFORCE
11	Polls every 10 minutes	2023-09-22 18:27:08 UTC

Activity — Past Hour

No data available for the selected time period

Mapped Fields

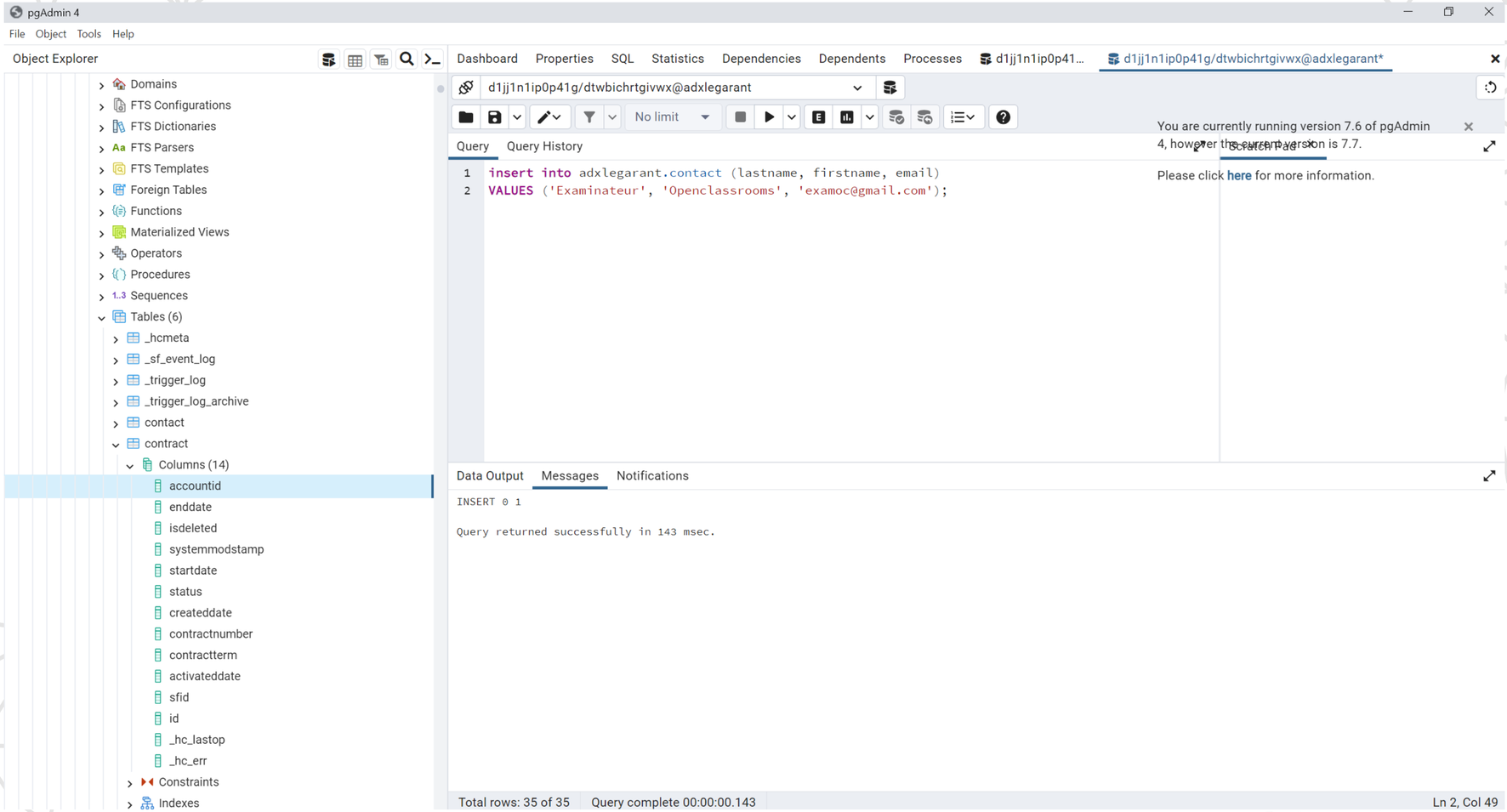
Field Name	Label	Type	Length	Indexed
AccountId	Account ID	reference to: Account	18	
ActivatedDate	Activated Date	datetime	0	
ContractNumber	Contract Number	string	30	
ContractTerm	Contract Term	int	0	
CreatedDate	Created Date	datetime	0	
EndDate	Contract End Date	date	0	
Id	Contract ID	id	18	Yes
IsDeleted	Deleted	boolean	0	
StartDate	Contract Start Date	date	0	
Status	Status	picklist	255	
SystemModstamp	System Modstamp	datetime	0	Yes

Environnement
Salesforce

Environnement PGAdmin

Requetes SQL

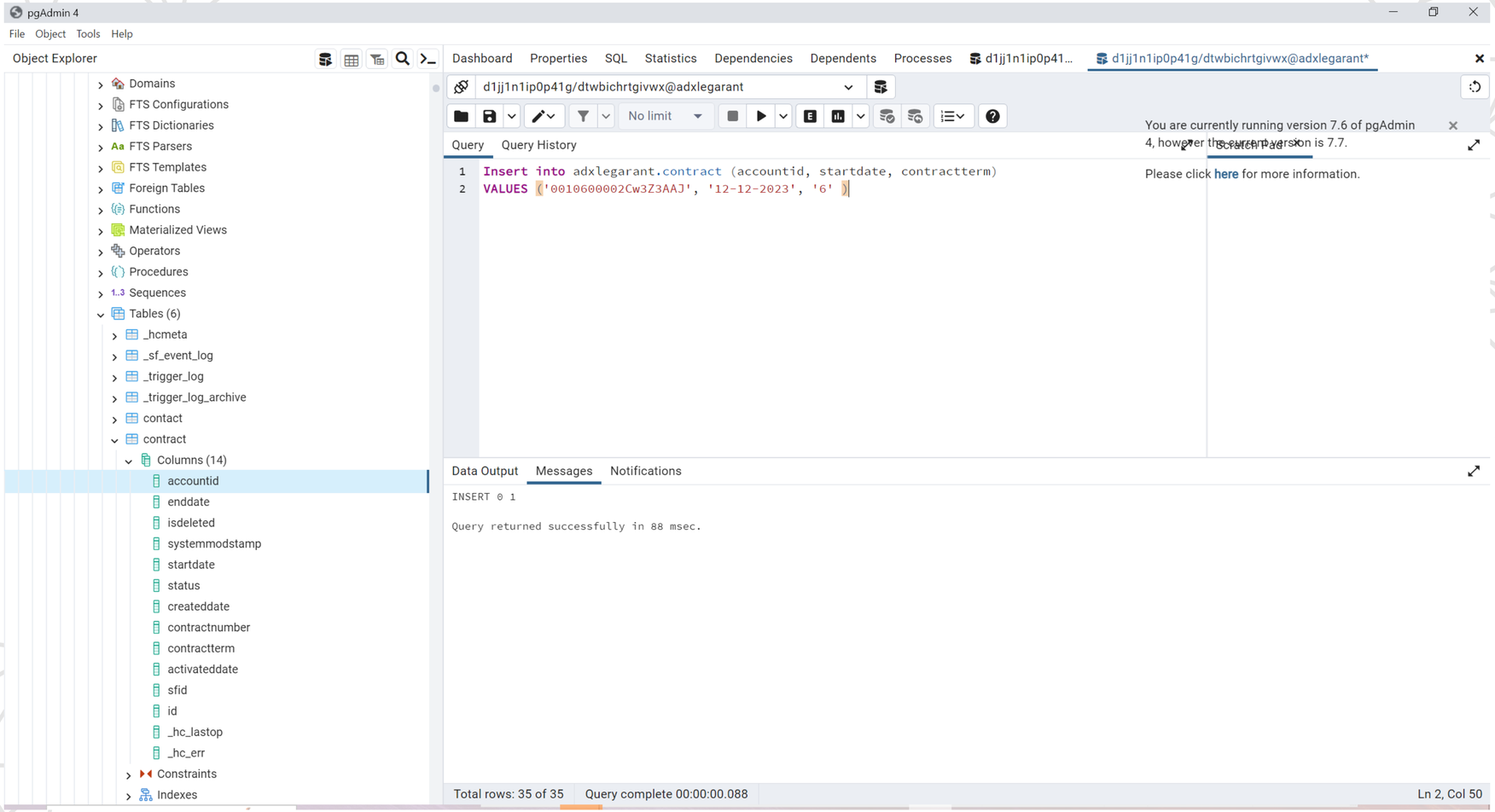
Ajout d'un contact



Environnement PGAdmin

Requetes SQL

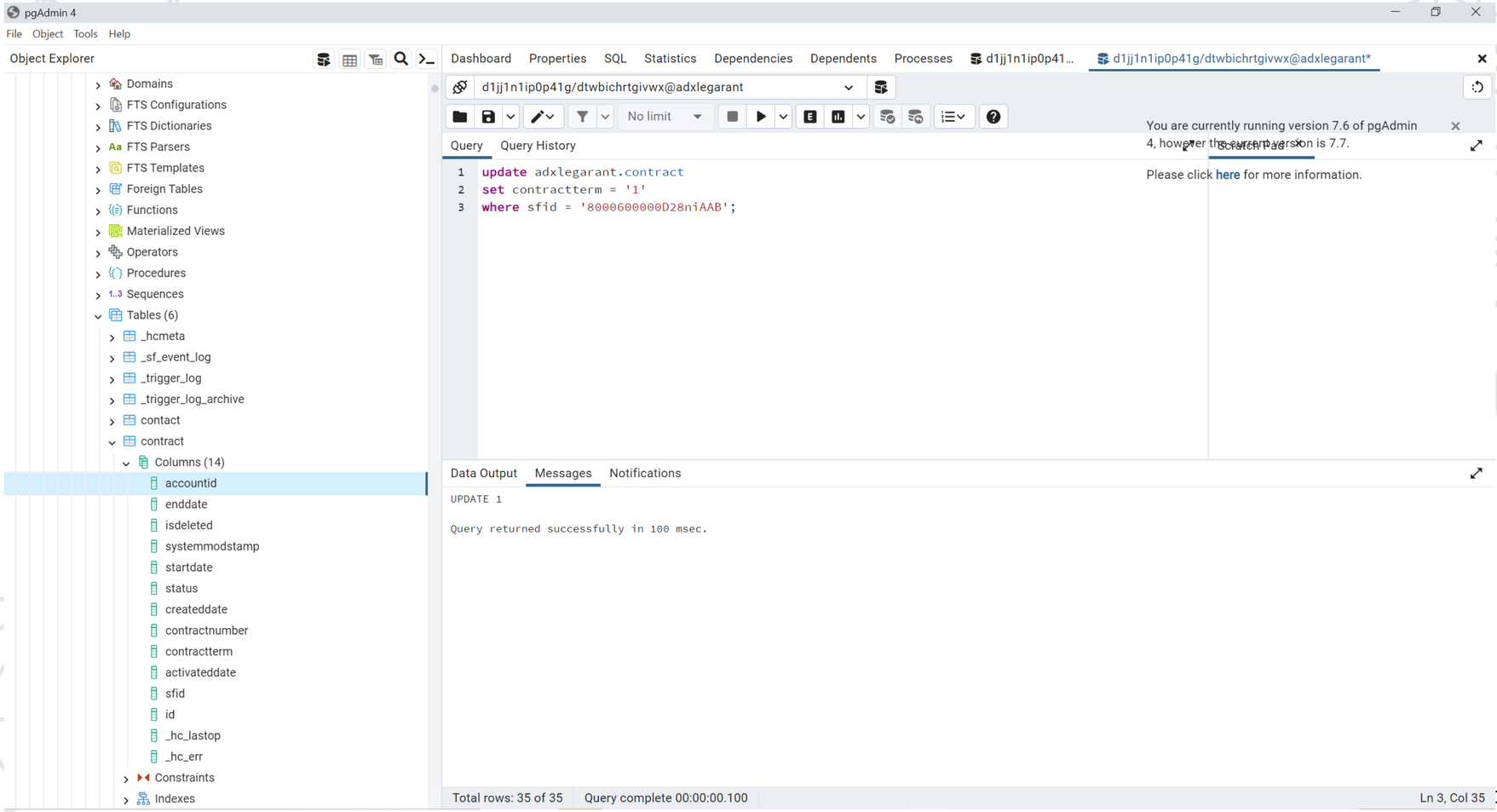
Ajout d'un contrat



Environnement PGAdmin

Requetes SQL

Modification d'un contact



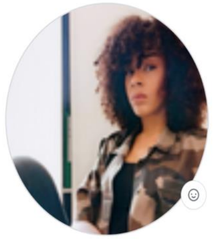
Déploiement de
L'application

Deployer l'application

Repository README.md

https://github.com/Magda-dev/Legarant_P12

Legarant Application Connectée Salesforce



madina_afrolicious
Magda-dev

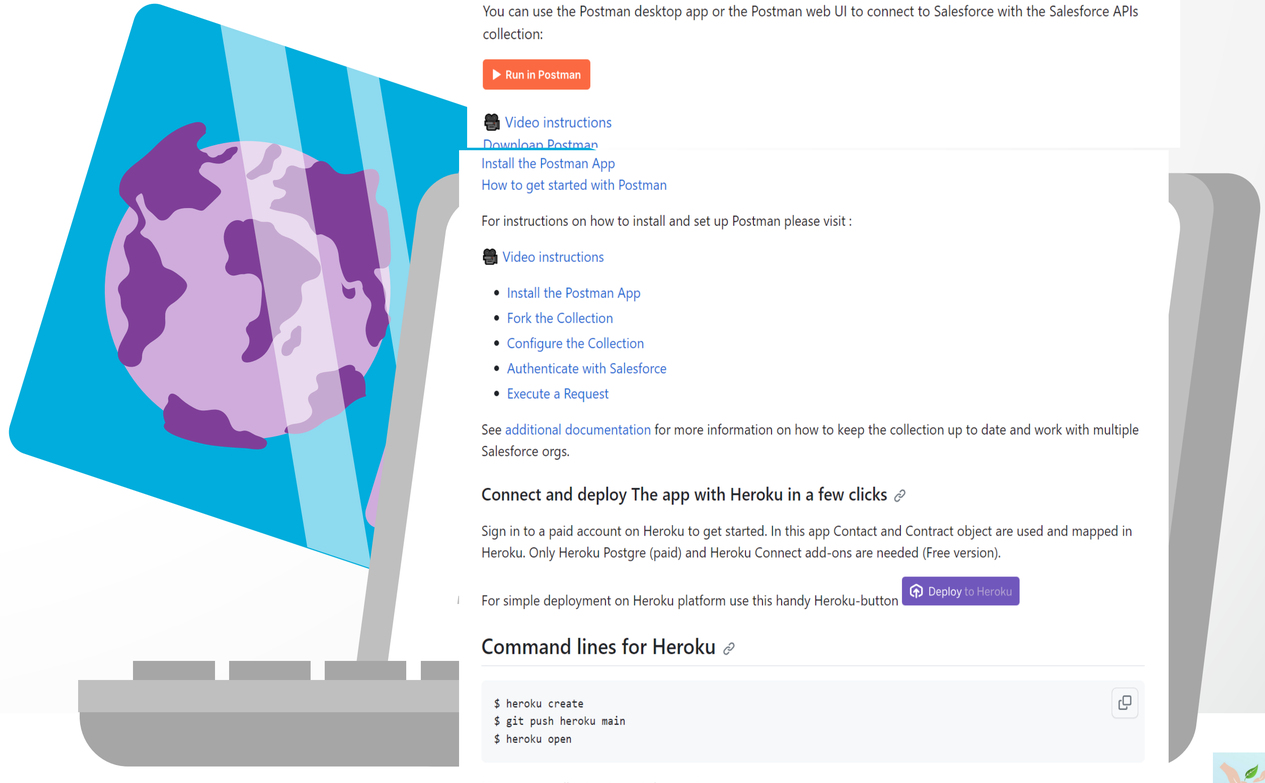
Fullstack Developer | Salesforce Developer

Edit profile

0 followers · 8 following

openclassrooms

Contributors	Release Date	Build	Social
Magdalena Larmet	Snapshot September 2023	BUILD passed	Linkedin Magdalena Larmet



How to deploy the app ?

Install the app package in your Salesforce org

Go to Salesforce and create you own regular org or sandbox. Then simply use the deploy to Salesforce org button to install the package on your own org and follow the 3 steps.

Deploy to Salesforce

Here are some documentation resources to get you started with Salesforce :

Configure Your Salesforce DX Project

The `sfdx-project.json` file contains useful configuration information for your project. See Salesforce DX Project Configuration in the *Salesforce DX Developer Guide* for details about this file.

Read more About Salesforce

- Salesforce Extensions Documentation
- Salesforce CLI Setup Guide
- Salesforce DX Developer Guide
- Salesforce CLI Command Reference

Install Postman

You can use the Postman desktop app or the Postman web UI to connect to Salesforce with the Salesforce APIs collection:

Run in Postman

Video instructions
Download Postman
Install the Postman App
How to get started with Postman

For instructions on how to install and set up Postman please visit :

Video instructions

- Install the Postman App
- Fork the Collection
- Configure the Collection
- Authenticate with Salesforce
- Execute a Request

See additional documentation for more information on how to keep the collection up to date and work with multiple Salesforce orgs.

Connect and deploy The app with Heroku in a few clicks

Sign in to a paid account on Heroku to get started. In this app Contact and Contract object are used and mapped in Heroku. Only Heroku Postgre (paid) and Heroku Connect add-ons are needed (Free version).

For simple deployment on Heroku platform use this handy Heroku-button Deploy to Heroku

Command lines for Heroku

```
$ heroku create
$ git push heroku main
$ heroku open
```

Now you are all set to try out the app !

Magdalena Larmet

Salesforce Developer