APEX INTEGRATION SERVICES

1. Apex REST Callouts

Challenge

```
public class AnimalLocator

{

        public static String getAnimalNameById(Integer id)

        {

                Http http = new Http();

                HttpRequest request = new HttpRequest();

                request.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/'+id);

                request.setMethod('GET');

                HttpResponse response = http.send(request);

                Map<String, Object> results = (Map<String, Object>)
                JSON.deserializeUntyped(response.getBody());

                Map<string,object> animals = (map<string,object>) results.get('animal');

                return string.valueof(animals.get('name')) ;

        }

}
```

```
@isTest

global class AnimalLocatorMock implements HttpCalloutMock

{

        global HttpResponse respond(HttpRequest request)

        {

                HttpResponse response = new HttpResponse();

                response.setHeader('Content-Type', 'application/json');

                response.setBody('{"animal":{"id":1,"name":"chicken","eats":"grains","says":"quack quack"}}');

                 response.setStatusCode(200);

                 return response;

        }

}
```

```
@isTest

private class AnimalLocatorTest

{

    @isTest static void AnimalsCalloutsTest()

    {

        Test.setMock(HttpCalloutMock.class, new AnimalLocatorMock());

        String actual = AnimalLocator.getAnimalNameById(1);

        String expected = 'chicken';

        System.assertEquals(actual, expected);

    }

}
```

2. Apex SOAP Callouts

Challenge

```
public class ParkLocator
{
    public static String[] country(String country)
    {
        ParkService.ParksImplPort prkSvc = new ParkService.ParksImplPort();
        String[] parksname = prkSvc.byCountry(country);
        return parksname;
    }
}

@isTest
private class ParkLocatorTest
{
    @isTest static void testCallout()
    {
        Test.setMock(WebServiceMock.class, new ParkServiceMock());
        List<String> actualParks = new List<String>();
        List<String> expectedParks = new List<String>{'Park1','Park2','Park3'};
        actualParks = ParkLocator.country('India');
        System.assertEquals(expectedParks, actualParks);
    }
}

@isTest
global class ParkServiceMock implements WebServiceMock
{
    global void doInvoke(
    Object stub,
    Object request,
        Map<String, Object> response,

        String endpoint,

        String soapAction,

        String requestName,

        String responseNS,

        String responseName,

        String responseType)

  {

        ParkService.byCountryResponse response_x = new ParkService.byCountryResponse();

        List<String> myStrings = new List<String> {'Park1','Park2','Park3'};

        response_x.return_x = myStrings;

        response.put('response_x', response_x);

  }

}
```

3. Apex Web Services

Challenge

```
@RestResource(urlMapping='/Accounts/*/contacts')
global with sharing class AccountManager
{
        @HttpGet
        global static Account getAccount()
        {
                RestRequest request = RestContext.request;
                String accountId = request.requestURI.substringBetween('Accounts/','/contacts');
                Account result = [SELECT Id,Name,(SELECT Id,Name FROM Contacts) FROM Account WHERE Id = :accountId];
                return result;
        }
}

@isTest
private class AccountManagerTest
{
        @isTest
        static void testGetAccount(){
        Account a = new Account(Name = 'TestAccount');
        insert a;
        Contact c = new Contact(LastName = 'Test', FirstName = 'Test', AccountId = a.Id);
        insert c;

        RestRequest request = new RestRequest();
        request.requestUri = 'https://playful-koala-n2mg4w-dev-ed.my.salesforce.com/services/apexrest/Accounts/' + a.Id +'/contacts';
        request.httpMethod = 'GET';
        RestContext.request = request;
        Account myAcc = AccountManager.getAccount();
        System.assert(myAcc != null);
        System.assertEquals('TestAccount', myAcc.Name);
  }
}
```