

Apex Integrating Services

Apex Rest Callouts

AnimalLocator.apxc

```
public class AnimalLocator {  
  
    public static String getAnimalNameById (Integer i){  
  
        Http http = new Http();  
  
        HttpRequest request = new HttpRequest();  
  
        request.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/'+i);  
  
        request.setMethod('GET');  
  
        HttpResponse response = http.send(request);  
  
  
        Map<String, Object> result =  
(Map<String, Object>)JSON.deserializeUntyped(response.getBody());  
  
        Map<String, Object> animal = (Map<String, Object>)result.get('animal');  
  
        System.debug('name:'+string.valueOf(animal.get('name')));  
  
        return string.valueOf(animal.get('name'));  
  
    }  
  
}
```

AnimalLocatorTest.apxc

```
@isTest  
  
private class AnimalLocatorTest {  
  
    @isTest  
  
    static void animalLocatorTest1() {  
  
        Test.setMock(HttpCalloutMock.class, new AnimalLocatorMock());  
  
        String actual = AnimalLocator.getAnimalNameById(1);  
  
        String expected = 'moose';  
  
        System.assertEquals(actual, expected);  
  
    }  
  
}
```

```
}
```

AnimalLocatorMock.apxc

@isTest

global class AnimalLocatorMock implements HttpCalloutMock {

```
    global HttpResponse respond(HttpRequest request){
        HttpResponse response = new HttpResponse();
        response.setHeader('contentType', 'application/json');
        response.setBody('{"animal":{"id":1,"name":"moose","eats":"plants","says":"bellows"}}');
        response.setStatusCode(200);
        return response;
    }
}
```

Apex Soap Callouts

ParkLocator.apxc

public class ParkLocator {

```
    public static List <String> country(String country){
        ParkService.ParksImplPort prkSvc = new ParkService.ParksImplPort();
        return prkSvc.byCountry(country);
    }
}
```

```
}
```

ParkServiceMock.apxc

@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();
        response_x.return_x = new List<String>{'Yosemite','Sequoia','Crater Lake'};
        response.put('response_x',response_x);

    }
}

```

ParkLocatorTest.apxc

```

@Test
private class ParkLocatorTest {

    @Test static void testCallout () {
        Test.setMock(WebServiceMock.class,new ParkServiceMock());
        String country = 'United States';
        List<String> expectedParks = new List<String>{'Yosemite','Sequoia','Crater Lake'};

        System.assertEquals(expectedParks,ParkLocator.country(country));
    }
}

```

Apex Web Services

AccountManager.apxc

```
@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,FirstName,LastName FROM Contacts)
```

```
            FROM Account
```

```
            WHERE Id = :accountId];
```

```
        return result;
```

```
    }
```

```
}
```

AccountManagerTest.apxc

```
@isTest
```

```
private class AccountManagerTest {
```

```
    @isTest
```

```
    static void testGetAccount() {
```

```
        Account a = new Account(Name='TestAccount');
```

```
        insert a;
```

```
        Contact c = new Contact(AccountId=a.Id,FirstName='Test',LastName='Test');
```

```
        insert c;
```

```
RestRequest request = new RestRequest();

request.requestURI =
'https://yourInstance.salesforce.com/services/apexrest/Accounts/'+a.id+'/contacts';

request.httpMethod = 'GET';

RestContext.request = request;


Account myAcct = AccountManager.getAccount();

System.assert(myAcct != null);

System.assertEquals('TestAccount', myAcct.Name);
}
}
```