

Apex Integration Services

Apex REST Callouts

AnimalLocator

```
public class AnimalLocator {

    public static String getAnimalNameById(Integer x){
        Http http = new Http();
        HttpRequest req = new HttpRequest();
        req.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/' + x);
        req.setMethod('GET');
        Map<String, Object> animal= new Map<String, Object>();
        HttpResponse res = http.send(req);
        if (res.getStatusCode() == 200) {
            Map<String, Object> results = (Map<String,
Object>)JSON.deserializeUntyped(res.getBody());
            animal = (Map<String, Object>) results.get('animal');
        }
        return (String)animal.get('name');
    }

}
```

AnimalLocatorTest

```
@isTest

private class AnimalLocatorTest{

    @isTest static void AnimalLocatorMock1() {
```

```

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

    string result = AnimalLocator.getAnimalNameById(3);

    String expectedResult = 'chicken';

    System.assertEquals(result,expectedResult );

}

}

```

AnimalLocatorMock

@isTest

global class AnimalLocatorMock implements HttpCalloutMock {

```

    global HTTPResponse respond(HTTPRequest request) {

        HttpResponse response = new HttpResponse();

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

        response.setBody('{ "animals": ["majestic badger", "fluffy bunny", "scary bear",
"chicken", "mighty moose"]}');

        response.setStatusCode(200);

        return response;

    }

}

```

Apex SOAP Callouts

ParkLocator

```
public class ParkLocator {  
    public static string[] country(string theCountry) {  
        ParkService.ParksImplPort parkSvc = new ParkService.ParksImplPort(); // remove  
space  
        return parkSvc.byCountry(theCountry);  
    }  
}
```

ParkLocatorTest

```
@isTest  
private class ParkLocatorTest {  
    @isTest static void testCallout() {  
        Test.setMock(WebServiceMock.class, new ParkServiceMock ());  
        String country = 'United States';  
        List<String> result = ParkLocator.country(country);  
        List<String> parks = new List<String>{'Yellowstone', 'Mackinac National Park',  
'Yosemite'};  
        System.assertEquals(parks, result);  
    }  
}
```

ParkServiceMock

```
@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>{'Yellowstone', 'Mackinac National Park',
        'Yosemite'};

        response.put('response_x', response_x);
    }
}

```

Apex Web Services

AccountManager

```

@RestResource(urlMapping='/Accounts/*/contacts')
global class AccountManager {
    @HttpGet

```

```

global static Account getAccount() {
    RestRequest req = RestContext.request;
    String accId = req.requestURI.substringBetween('Accounts/', '/contacts');
    Account acc = [SELECT Id, Name, (SELECT Id, Name FROM Contacts)
                  FROM Account WHERE Id = :accId];
    return acc;
}
}

```

AccountManagerTest

@isTest

```

private class AccountManagerTest {

    private static testMethod void getAccountTest1() {
        Id recordId = createTestRecord();

        RestRequest request = new RestRequest();
        request.requestUri = 'https://na1.salesforce.com/services/apexrest/Accounts/' +
recordId + '/contacts' ;
        request.httpMethod = 'GET';
        RestContext.request = request;
        Account thisAccount = AccountManager.getAccount();
        System.assert(thisAccount != null);
        System.assertEquals('Test record', thisAccount.Name);

    }

    static Id createTestRecord() {

```

```
Account TestAcc = new Account(  
    Name='Test record');  
insert TestAcc;  
Contact TestCon= new Contact(  
    LastName='Test',  
    AccountId = TestAcc.id);  
return TestAcc.Id;  
}  
}
```