Apex Class

@RestResource(urlMapping='/Positions/\*')

global with sharing class PositionManager {

@HttpGet

global static Position\_\_c getPositionById(){

RestRequest request = RestContext.request;

String positionId = request.requestURI.substring(

request.requestURI.lastIndexof('/')+1);

Position\_\_c result = [Select Name, Id from Position\_\_c

where Id=:positionId];

return result;

}

@HttpPost

global static ID createPosition(String Name)

{

Position\_\_c thisPosition = new Position\_\_c(

Name=Name);

insert thisPosition;

return thisPosition.Id;

}

@HttpPut

global static ID upsertPosition(String Name, String Id)

{

Position\_\_c thisPosition = new Position\_\_c(

Id=id,

Name=Name);

upsert thisPosition;

return thisPosition.Id;

}

@HttpDelete

global static void deletePosition(){

RestRequest request = RestContext.request;

String positionId = request.requestURI.substring(

request.requestURI.lastIndexOf('/')+1);

Position\_\_c thisPosition = [select Id from Position\_\_c

where id=:positionId];

delete thisPosition;

}

@HttpPatch

global Static Id UpdatePositionFields(){

RestRequest request = RestContext.request;

String positionId = request.requestURI.substring(

request.requestURI.lastIndexOf('/')+1);

Position\_\_c thisPosition = [select Id from Position\_\_c

where Id=:positionId];

Map<String, Object> params = (Map<String,Object>)JSON.deserializeUntyped(request.requestBody.tostring());

for(String fieldName:params.keySet()){

thisPosition.put(fieldName,params.get(fieldName));

}

update thisPosition;

return thisPosition.Id;

}

}

Apex test class

@IsTest

private class PositionManagerTest {

@isTest static void testGetCaseById() {

Id recordId = createTestRecord();

// Set up a test request

RestRequest request = new RestRequest();

request.requestUri =

'https://yourInstance.salesforce.com/services/apexrest/Positions/'

+ recordId;

request.httpMethod = 'GET';

RestContext.request = request;

// Call the method to test

Position\_\_c thisPosition = PositionManager.getPositionById();

// Verify results

System.assert(thisPosition != null);

System.assertEquals('Test record', thisPosition.Name);

}

@isTest static void testCreateCase() {

// Call the method to test

ID thisPositionId = PositionManager.createPosition(

'Ferocious chipmunk');

// Verify results

System.assert(thisPositionId != null);

Position\_\_c thisPosition = [SELECT Id,Name FROM Position\_\_C

WHERE Id=:thisPositionId];

System.assert(thisPosition != null);

System.assertEquals(thisPosition.Name, 'Ferocious chipmunk');

}

@isTest static void testDeleteCase() {

Id recordId = createTestRecord();

// Set up a test request

RestRequest request = new RestRequest();

request.requestUri =

'https://yourInstance.salesforce.com/services/apexrest/Positions/'

+ recordId;

request.httpMethod = 'GET';

RestContext.request = request;

// Call the method to test

PositionManager.deletePosition();

// Verify record is deleted

List<Position\_\_c> position = [SELECT Id FROM Position\_\_c

WHERE Id=:recordId];

System.assert(position.size() == 0);

}

@isTest static void testUpsertCase() {

// 1. Insert new record

ID position1Id = PositionManager.upsertPosition(

'Ferocious chipmunk', null);

// Verify new record was created

System.assert(position1Id != null);

Position\_\_c position1 = [SELECT Id,Name FROM Position\_\_c

WHERE Id=:position1Id];

System.assert(position1 != null);

System.assertEquals(position1.Name, 'Ferocious chipmunk');

// 2. Update status of existing record to Working

ID position2Id = PositionManager.upsertPosition(

'Ferocious chipmunk', position1Id);

// Verify record was updated

System.assertEquals(position1Id, position2Id);

Position\_\_c position2 = [SELECT Id,Name FROM Position\_\_c

WHERE Id=:position2Id];

System.assert(position2 != null);

System.assertEquals(position2.Name, 'Working');

}

@isTest static void testUpdatePositionFields() {

Id recordId = createTestRecord();

RestRequest request = new RestRequest();

request.requestUri =

'https://yourInstance.salesforce.com/services/apexrest/Cases/'

+ recordId;

request.httpMethod = 'PATCH';

request.addHeader('Content-Type', 'application/json');

request.requestBody = Blob.valueOf('{"Name": "Working"}');

RestContext.request = request;

// Update status of existing record to Working

ID thisPositionId = PositionManager.updatePositionFields();

// Verify record was updated

System.assert(thisPositionId != null);

Position\_\_c thisPosition = [SELECT Id,Name FROM Position\_\_c

WHERE Id=:thisPositionId];

System.assert(thisPosition != null);

System.assertEquals(thisPosition.Name, 'Working');

}

// Helper method

static Id createTestRecord() {

// Create test record

Position\_\_c positionTest = new Position\_\_c(

Name='Test record');

insert positionTest;

return positionTest.Id;

}

}