

# Apex SOAP Callouts

## Code for ParkService

//Generated by wsdl2apex

```
public class ParkService {
    public class byCountryResponse {
        public String[] return_x;
        private String[] return_x_type_info = new
String[] {'return','http://parks.services/',null,'0','-1','false'};
        private String[] apex_schema_type_info = new
String[] {'http://parks.services/','false','false'};
        private String[] field_order_type_info = new String[] {'return_x'};
    }
    public class byCountry {
        public String arg0;
        private String[] arg0_type_info = new
String[] {'arg0','http://parks.services/',null,'0','1','false'};
        private String[] apex_schema_type_info = new
String[] {'http://parks.services/','false','false'};
        private String[] field_order_type_info = new String[] {'arg0'};
    }
    public class ParksImplPort {
        public String endpoint_x = 'https://th-apex-soap-
service.herokuapp.com/service/parks';
        public Map<String,String> inputHttpHeaders_x;
        public Map<String,String> outputHttpHeaders_x;
        public String clientCertName_x;
        public String clientCert_x;
        public String clientCertPasswd_x;
        public Integer timeout_x;
        private String[] ns_map_type_info = new String[] {'http://parks.services/',
'ParkService'};
        public String[] byCountry(String arg0) {
```

```

        ParkService.byCountry request_x = new ParkService.byCountry();
        request_x.arg0 = arg0;
        ParkService.byCountryResponse response_x;
        Map<String, ParkService.byCountryResponse> response_map_x = new
Map<String, ParkService.byCountryResponse>();
        response_map_x.put('response_x', response_x);
        WebServiceCallout.invoke(
            this,
            request_x,
            response_map_x,
            new String[]{endpoint_x,
                "",
                'http://parks.services/',
                'byCountry',
                'http://parks.services/',
                'byCountryResponse',
                'ParkService.byCountryResponse'}
        );
        response_x = response_map_x.get('response_x');
        return response_x.return_x;
    }
}
}

```

## Code for ParkServiceMock

```

@Test
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> lstOfDummyParks = new List<String> {'Park1','Park2','Park3'};
        response_x.return_x = lstOfDummyParks;

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

```

## Code for ParkLocator

```

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

```

## Code for ParkLocatorTest

```

@Test
private class ParkLocatorTest{
    @Test
    static void testParkLocator() {
        Test.setMock(WebServiceMock.class, new ParkServiceMock());
        String[] arrayOfParks = ParkLocator.country('India');

        System.assertEquals('Park1', arrayOfParks[0]);
    }
}
}

```