**What you need?**

JDK 10+

Apache Maven 3.6+

IntelliJ/Eclipse IDE

**Clone Example Projects:**

Clone the “consumer” [appointment-service] & “provider” [patient-service] projects from the below GithHub repos and checkout “**cdc\_mod3\_day1\_begin**” branch.

<https://github.com/pradip79/appointment-service.git>

<https://github.com/pradip79/patient-service.git>

**Provider Side:**

**Step1**: Write the consumer contract in “src/test/resources/contracts/shouldReturnPatient.groovy”

1. Create the below contract in “src/test/resources/contracts/shouldReturnPatient.groovy”

**import** org.springframework.cloud.contract.spec.Contract  
  
Contract.make **{** description(**"Should return a patient"**)  
 request **{** method GET()  
 url(**"/patient-service/patient"**)**{** queryParameters **{** parameter(**"MRN"**, **"2009120401"**)  
 **}  
 }  
 }** response **{** status(200)  
 headers **{** contentType applicationJson()  
 **}** body(  
 **firstName**: **"John"**,  
 **lastName**: **"Smart"**,  
 **dateOfBirth**: **"02/02/1997"**,  
 **registrationDate**: **"04/12/2009"**,  
 **address**: **"15 Foreshore Road, Philadelphia, PA, 19101"**,  
 **mrn**: **"2009120401"** )  
 **}  
}**

1. Add spring-cloud-contract-verifier dependency in the pom.xml:

<**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-contract-verifier</**artifactId**>  
 <**version**>3.0.0</**version**>  
 <**scope**>test</**scope**>  
</**dependency**>

1. Add spring-cloud-contract dependency in “dependencyManagement” section:

<**dependencyManagement**>  
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-dependencies</**artifactId**>  
 <**version**>2020.0.0</**version**>  
 <**type**>pom</**type**>  
 <**scope**>import</**scope**>  
 </**dependency**>  
 </**dependencies**>  
</**dependencyManagement**>

1. Add spring-cloud-contract-maven plugin

<**plugin**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-contract-maven-plugin</**artifactId**>  
 <**version**>3.0.0</**version**>  
 <**extensions**>true</**extensions**>  
 <**configuration**>  
 <**testFramework**>JUNIT5</**testFramework**>  
 <**testMode**>MOCKMVC</**testMode**>   
 </**configuration**>  
</**plugin**>

**Step2**: Run “mvn clean test” 🡪 Verify the build failed but below contract verifier test generated:

Graphical user interface, text, application, chat or text message

Description automatically generated

**Step3**: Write a base test class and configure it to be the base class for the generated contract verifier test

1. Add the below test class in “src/test/java”

**package** com.testannotation.patientservice;  
  
**import** com.testannotation.patientservice.controller.PatientController;  
**import** io.restassured.module.mockmvc.RestAssuredMockMvc;  
**import** org.junit.jupiter.api.BeforeEach;  
**import** org.junit.jupiter.api.extension.ExtendWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.test.context.SpringBootTest;  
**import** org.springframework.cloud.contract.verifier.messaging.boot.AutoConfigureMessageVerifier;  
**import** org.springframework.test.context.junit.jupiter.SpringExtension;  
**import** org.springframework.test.web.servlet.setup.MockMvcBuilders;  
**import** org.springframework.test.web.servlet.setup.StandaloneMockMvcBuilder;  
  
@ExtendWith(SpringExtension.**class**)  
@SpringBootTest (webEnvironment = SpringBootTest.WebEnvironment.***RANDOM\_PORT***)  
@AutoConfigureMessageVerifier  
**public class** BaseContractTest {  
  
 @Autowired  
 **private** PatientController **patientController**;  
  
 @BeforeEach  
 **public void** setUp(){  
 StandaloneMockMvcBuilder standaloneMockMvcBuilder  
 = MockMvcBuilders.*standaloneSetup*(**patientController**);  
 RestAssuredMockMvc.*standaloneSetup*(standaloneMockMvcBuilder);  
 }  
}

1. Add below highlighted configuration in “spring-cloud-contract-maven” plugin.

<**plugin**>  
 <**groupId**>org.springframework.cloud</**groupId**>  
 <**artifactId**>spring-cloud-contract-maven-plugin</**artifactId**>  
 <**version**>3.0.0</**version**>  
 <**extensions**>true</**extensions**>  
 <**configuration**>  
 <**testFramework**>JUNIT5</**testFramework**>  
 <**testMode**>MOCKMVC</**testMode**>  
 <**baseClassForTests**>  
 com.testannotation.patientservice.BaseContractTest  
 </**baseClassForTests**>  
 </**configuration**>  
</**plugin**>

1. Run “mvn clean install” 🡪 Verify build is successful and below message appears in console

Installing .\.\target\patient-service-0.0.1-SNAPSHOT-stubs.jar to \.\.m2\repository\com\testannotation\patient-service\0.0.1-SNAPSHOT\patient-service-0.0.1-SNAPSHOT-stubs.jar

**Consumer Side:**

**Step1:**

Add the new integration test similar to the existing Wiremock test (but without the @BeforeEach method) to be executed against the generated & installed stub from the provider build.

**package** com.testannotation.appointmentservice;  
**import** com.testannotation.appointmentservice.client.PatientClient;  
**import** com.testannotation.appointmentservice.model.Patient;  
**import** org.junit.jupiter.api.Test;  
**import** org.junit.jupiter.api.extension.ExtendWith;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.boot.test.context.SpringBootTest;  
**import** org.springframework.cloud.contract.stubrunner.spring.AutoConfigureStubRunner;  
**import** org.springframework.cloud.contract.stubrunner.spring.StubRunnerProperties;  
**import** org.springframework.http.ResponseEntity;  
**import** org.springframework.test.context.junit.jupiter.SpringExtension;  
**import static** org.assertj.core.api.BDDAssertions.*then*;  
  
@ExtendWith(SpringExtension.**class**)  
@SpringBootTest  
@AutoConfigureStubRunner(ids = **"com.testannotation:patient-service:+:7070"**,  
 stubsMode = StubRunnerProperties.StubsMode.***LOCAL***)  
**public class** AppointServiceIntegrationTestSCC {  
 @Autowired  
 PatientClient **patientClient**;  
  
 @Test  
 **public void** givenMRN\_whenSearchForPatient\_shouldReturnAPatient(){  
 *//Given* String MRN = **"2009120401"**;  
  
 *//When* ResponseEntity<Patient> response = **patientClient**.searchPatient(MRN);  
  
 *//Then  
 then*(response.getStatusCode().equals(200));  
 *then*(response.getBody().getMRN().equals(MRN));  
 *then*(response.getBody().getDateOfRegistration().equals(**"04/12/2009"**));  
 }  
}

**Step2:**

1. Update the port# in the application.properties to point to the port# provided in the above test (7070)
2. Run the test and see it fail for NullPointerException at below line:

*then*(response.getBody().getDateOfRegistration().equals(**"04/12/2009"**));

The failure is because based on the contract the response body is as below:

**firstName**: **"John"**,  
 **lastName**: **"Smart"**,  
 **dateOfBirth**: **"02/02/1997"**,  
 **registrationDate**: **"04/12/2009"**,  
 **address**: **"15 Foreshore Road, Philadelphia, PA, 19101"**,  
 **mrn**: **"2009120401"**

But our consumer side POJO (Patient.java) has the below field:

**private** String **dateOfRegistration**

1. Refactor & rename the field to match to the defined contract and rerun the test 🡪 See it pass.