Karate API Testing Documentation

# 1. Project Setup

Steps:  
  
1. Create a Maven project in STS (Spring Tool Suite) or Eclipse.  
2. Add the following dependencies to your pom.xml:

<dependencies>  
 <dependency>  
 <groupId>com.intuit.karate</groupId>  
 <artifactId>karate-core</artifactId>  
 <version>1.4.1</version>  
 </dependency>  
 <dependency>  
 <groupId>com.intuit.karate</groupId>  
 <artifactId>karate-apache</artifactId>  
 <version>0.9.6</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>com.intuit.karate</groupId>  
 <artifactId>karate-junit5</artifactId>  
 <version>1.4.1</version>  
 <scope>test</scope>  
 </dependency>  
</dependencies>

3. Ensure feature files are placed under src/test/java/[your\_package]/.

# 2. Writing and Running a Basic Test Using JUnit Runner

Feature File: src/test/java/examples/getUsers.feature

Feature: Basic GET API Test  
  
 Scenario: Get user data  
 Given url baseUrl + '/users/1'  
 When method get  
 Then status 200

JUnit Runner: src/test/java/examples/RunKarateTest.java

package examples;  
  
import com.intuit.karate.junit5.Karate;  
  
class RunKarateTest {  
  
 @Karate.Test  
 Karate testAll() {  
 return Karate.run().relativeTo(getClass());  
 }  
}

To run the test:

From terminal: mvn test

From STS:  
1. Right-click the project > Run As > Maven Build…  
2. Enter test in the Goals field  
3. Click Run

# 3. Viewing Karate Test Reports

Reports are available at: target/karate-reports/karate-summary.html

Open karate-summary.html in your browser to view results.

# 4. Environment-Specific Configuration

File: src/test/java/karate-config.js

function fn() {  
 var env = karate.env;  
 if (!env) {  
 env = 'dev';  
 }  
  
 var config = {  
 baseUrl: ''  
 };  
  
 if (env === 'dev') {  
 config.baseUrl = 'https://diascom.dev.truist.com/';  
 } else if (env === 'test') {  
 config.baseUrl = 'https://test.example.com/api';  
 } else if (env === 'prod') {  
 config.baseUrl = 'https://prod.example.com/api';  
 }  
  
 return config;  
}

Usage in .feature file:

Given url baseUrl + '/users'

# 5. Running Tests for Specific Environments

From terminal: mvn test -Dkarate.env=test

From STS:  
1. Right-click the project > Run As > Maven Build…  
2. Set Goals: test  
3. Click the Arguments tab  
4. In the VM arguments, enter: -Dkarate.env=test  
5. Click Run

# 6. Grouping and Filtering Tests Using Tags

Tag your scenarios or features:

@basic  
Scenario: Basic user retrieval  
 Given url baseUrl + '/users/1'  
 When method get  
 Then status 200

Update runner class to include tags:

@Karate.Test  
Karate runTagged() {  
 return Karate.run().tags("@basic").relativeTo(getClass());  
}

From terminal: mvn test -Dkarate.options="--tags @basic"

From STS:  
1. Right-click the project > Run As > Maven Build…  
2. Set Goals: test  
3. Click the Arguments tab  
4. In the VM arguments, enter: -Dkarate.options="--tags @basic"  
5. Click Run