

karate apache 1.4.1 not available dependency dans pom

Sources

Karate core
dependency...

stackov... 1

karate-demo
build fails to fin...

github 2

com.intuit.karate
core - Maven...

central... 3



Voir 2 de plus

Réponse

The issue you are facing is related to the dependency version of karate-core or karate-junit5 in your Maven project's pom.xml file. Based on the search results:

The latest stable version of Karate at the time of this query (June 10, 2024) is 1.4.1. ³ However, your pom.xml is trying to use version 0.9.6 of the karate-apache artifact, which is an older version. ⁴

To resolve this, you need to update the version in your pom.xml to the latest 1.4.1 version:

```
xml
<dependency>
  <groupId>com.intuit.karate</groupId>
  <artifactId>karate-junit5</artifactId>
  <version>1.4.1</version>
  <scope>test</scope>
</dependency>
```

Alternatively, you can use the karate-core artifact with the 'all'



Poser un suivi

Politique en matière de cookies

Nous utilisons des cookies pour améliorer votre expérience. En cliquant sur "Accepter tous les cookies" ou en sélectionnant "Cookies nécessaires", vous acceptez notre [politique de confidentialité](#).

Cookies nécessaires

Accepter tous les cookies

classifier to avoid dependency conflicts: 4

```
xml
<dependency>
  <groupId>com.intuit.karate</groupId>
  <artifactId>karate-core</artifactId>
  <version>1.4.1</version>
  <classifier>all</classifier>
  <scope>test</scope>
</dependency>
```

Make sure to also update the `<karate.version>` property in your pom.xml to 1.4.1. 1

This should resolve the dependency issue you are facing. 1 3

4

Partager



run compose with karate framework

Sources

Karate Framework fro...
youtube • 1

Karate | Test Automation...
karatela... • 2

com.intuit.karate apache 0.9.6 o...
libraries • 3

Voir 2 de plus

Réponse

Start Creating Karate Framework from Scratch, Add Feature File, Create Test Runner, Execute and Verify Reports

Learn Automated API Testing using Karate Framework from Scratch – Session 2



Karate



Watch

To run tests in parallel using the Karate fr leverage the `parallel()` method provid class. Here's how you can se up:

`ParallelRunner.java`:

Politique en matière de cookies

Nous utilisons des cookies pour améliorer votre expérience. En cliquant sur "Accepter tous les cookies" ou en sélectionnant "Cookies nécessaires", vous acceptez notre [politique de confidentialité](#).

Cookies nécessaires

Accepter tous les cookies

```

java
import com.intuit.karate.Results;
import com.intuit.karate.Runner;
import org.junit.Test;
import static org.junit.Assert.assertTrue;

public class ParallelRunner {
    @Test
    public void testParallel() {
        Results results = Runner.path("classpath:path/to/features")
            .tags("~@ignore") // ignore tagged tests
            .parallel(5); // run in 5 parallel threads
        assertTrue(results.getErrorMessages(), results.getErrors().isEmpty());
    }
}

```

In the above example:

- `Runner.path("classpath:path/to/features")` specifies the location of your feature files.
 - `tags("~@ignore")` runs all tests except those tagged with `@ignore`.
 - `parallel(5)` runs the tests in 5 parallel threads.
2. In your `pom.xml`, add the Karate dependencies and configure the `maven-surefire-plugin` to run the `ParallelRunner` class: ¹

```

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

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M5</version>
      <configuration>
        <includes>
          <include>src/test/java/**/*Test.java</include>
        </includes>
      </configuration>
    </plugin>
  </plugins>
</build>

```

Politique en matière de cookies

Nous utilisons des cookies pour améliorer votre expérience. En cliquant sur "Accepter tous les cookies" ou en sélectionnant "Cookies nécessaires", vous acceptez notre [politique de confidentialité](#).

[Cookies nécessaires](#)

[Accepter tous les cookies](#)

```
</configuration>
</plugin>
</plugins>
</build>
```

3. Run the tests in parallel using Maven:

```
mvn test
```

This will execute all your feature files in parallel using the specified number of threads. 1 4

You can customize the parallel execution by adjusting the ``path()``, ``tags()``, and ``parallel()`` methods in the ``ParallelRunner`` class. The ``Results`` object provides information about the test run, including any failures or errors.

4

Partager



run a compose file docker with karate before start test

Réponse

To run a Docker Compose file before starting your Karate tests, you can use a setup script or a Maven plugin. Here's an example of how you can achieve this using the ``docker-compose-maven-plugin``:

1. Add the ``docker-compose-maven-plugin`` to your ``pom.xml`` file:

```
xml
<build>
  <plugins>
    <!-- Other plugins -->
```

Politique en matière de cookies

Nous utilisons des cookies pour améliorer votre expérience. En cliquant sur "Accepter tous les cookies" ou en sélectionnant "Cookies nécessaires", vous acceptez notre [politique de confidentialité](#).

Cookies nécessaires

Accepter tous les cookies

```

        <execution>
            <id>up</id>
            <phase>pre-integration-test</phase>
            <goals>
                <goal>up</goal>
            </goals>
            <configuration>
                <composeFile>${project.basedir}
            </configuration>
        </execution>
        <execution>
            <id>down</id>
            <phase>post-integration-test</phas
            <goals>
                <goal>down</goal>
            </goals>
            <configuration>
                <composeFile>${project.basedir
            </configuration>
        </execution>
    </executions>
</plugin>
</plugins>
</build>

```

2. Replace ``${project.basedir}/path/to/docker-compose.yml`` with the actual path to your Docker Compose file.
3. In the ``<execution>`` sections, you can see that the ``up`` goal is bound to the ``pre-integration-test`` phase, which means it will run before your Karate tests. The ``down`` goal is bound to the ``post-integration-test`` phase, which means it will run after your Karate tests have completed.
4. Add the Karate dependencies and configure the ``maven-surefire-plugin`` to run your Karate tests:

```

xml
<dependencies>
    <dependency>
        <groupId>com.intuit.karate<
        <artifactId>karate-junit5</
    </dependency>
</dependencies>

```



Politique en matière de cookies

Nous utilisons des cookies pour améliorer votre expérience. En cliquant sur "Accepter tous les cookies" ou en sélectionnant "Cookies nécessaires", vous acceptez notre [politique de confidentialité](#).

Cookies nécessaires

Accepter tous les cookies

```
<build>
  <plugins>
    <!-- Other plugins -->
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>3.0.0-M9</version>
      <configuration>
        <includes>
          <include>/**/*.java</include>
        </includes>
      </configuration>
    </plugin>
  </plugins>
</build>
```

5. Run your Karate tests using Maven:

```
mvn clean test
```

This will first start the Docker Compose services (`docker-compose up`), then run your Karate tests, and finally stop the Docker Compose services (`docker-compose down`).

Make sure to replace the plugin versions and configurations with the appropriate values for your project. Additionally, ensure that your Docker Compose file is correctly configured and accessible from the specified path.

↗ Partager

🔗 Questions liées

how do I integrate Docker Compose with Karate for automated testing



what are the best practices for running Karate tests in a Docker container



how can I configure Docker Compose to



Politique en matière de cookies

Nous utilisons des cookies pour améliorer votre expérience. En cliquant sur "Accepter tous les cookies" ou en sélectionnant "Cookies nécessaires", vous acceptez notre [politique de confidentialité](#).

Cookies nécessaires

Accepter tous les cookies