

Video Recorder

Table of Contents

JUnit Rule:.....	1
TestNG:	2
Configuration	4
Remote Video Recording:	5
TestNG + Remote Video recorder	5

[\[Build Status\]](#)

This library allows easily record video of your UI tests by just putting couple annotations.

Supports popular Java test frameworks:

- JUnit
- TestNg
- Spock
- Selenium Grid

JUnit Rule:

maven

```
<dependency>
  <groupId>com.automation-remarks</groupId>
  <artifactId>video-recorder-junit</artifactId>
  <version>LATEST</version>
</dependency>
```

gradle

```
compile group: 'com.automation-remarks', name: 'video-recorder-junit', version: '1.+'
```

```
import com.automation.remarks.video.annotations.Video;
import com.automation.remarks.video.junit.VideoRule;
import org.junit.Rule;
import org.junit.Test;

import static junit.framework.Assert.assertTrue;

public class JUnitVideoTest {

    @Rule
    public VideoRule videoRule = new VideoRule();

    @Test
    @Video
    public void shouldFailAndCreateRecordWithTestName() {
        Thread.sleep(5000);
        assert false;
    }

    @Test
    @Video(name = "second_test")
    public void videoShouldHaveNameSecondTest() {
        Thread.sleep(10000);
        assertTrue(false);
    }

    @Test
    @Video(name = "third_test", enabled = false)
    public void shouldFailWithoutRecording() {
        Thread.sleep(10000);
        assertTrue(false);
    }
}
```

TestNG:

maven

```
<dependency>
  <groupId>com.automation-remarks</groupId>
  <artifactId>video-recorder-testng</artifactId>
  <version>LATEST</version>
</dependency>
```

By default recorder will include testng dependency 6.9.10. If you want to place other version of TestNG use this declaration:

maven

```
<dependency>
  <groupId>com.automation-remarks</groupId>
  <artifactId>video-recorder-testng</artifactId>
  <version>1.0.2</version>
  <exclusions>
    <exclusion>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

gradle

```
compile group: 'com.automation-remarks', name: 'video-recorder-testng', version: '1.+'
```

```
import com.automation.remarks.video.annotations.Video;
import com.automation.remarks.video.testng.VideoListener;
import org.testng.annotations.Listeners;
import org.testng.annotations.Test;

import static junit.framework.Assert.assertTrue;

@Listeners(VideoListener.class)
public class TestNgVideoTest {

    @Test
    @Video
    public void shouldFailAndCreateRecordWithTestName() {
        Thread.sleep(1000);
        assert false;
    }

    @Test
    @Video(name = "second_test")
    public void videoShouldHaveNameSecondTest(){
        Thread.sleep(1000);
        assertTrue(false);
    }

    @Test
    @Video(name = "third_test", enabled = false)
    public void shouldFailWithoutRecording() {
        Thread.sleep(1000);
        assertTrue(false);
    }
}
```

Configuration

You are able to set some configuration parameters for Video Recorder:

```
// Default video folder is ${user.dir}/video. Could be changed by:
VideoRecorder.conf().withVideoFolder("custom_folder")
    // Video could be disabled globally. Set to "true"
    .videoEnabled(true)
    // There is two recording modes ANNOTATED AND ALL
    // Annotated is default and works only with methods annotated by @Video
    .withRecordMode(RecordingMode.ANNOTATED);
```

Remote Video Recording:

Build remote module:

```
./gradlew remote:jar
```

Run hub:

```
java -jar video-recorder-remote-1.0.jar -role hub -servlets
"com.automation.remarks.remote.hub.Video"
```

Run node:

```
java -jar video-recorder-remote-1.0.jar -servlets
"com.automation.remarks.remote.node.VideoServlet" -role node -port 5555 -hub
"http://localhost:4444/grid/register"
```

TestNG + Remote Video recorder

Change listener in your tests to **RemoteVideoListener**:

```
import com.automation.remarks.video.annotations.Video;
import com.automation.remarks.video.testng.VideoListener;
import org.testng.annotations.Listeners;
import org.testng.annotations.Test;

import static junit.framework.Assert.assertTrue;

@Listeners(RemoteVideoListener.class)
public class TestNgRemoteVideonTest {

    @Test
    @Video
    public void shouldFailAndCreateRecordWithTestName() {
        Thread.sleep(1000);
        assert false;
    }

    @Test
    @Video(name = "second_test")
    public void videoShouldHaveNameSecondTest(){
        Thread.sleep(1000);
        assertTrue(false);
    }

    @Test
    @Video(name = "third_test", enabled = false)
    public void shouldFailWithoutRecording() {
        Thread.sleep(1000);
        assertTrue(false);
    }
}
```

more [details](http://automation-remarks.com) by automation-remarks.com