Video Recorder

Table of Contents

[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: '1.0'
```

JUnitVideoTest.class

```
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>
```

gradle

```
compile group: 'com.automation-remarks', name: 'video-recorder-testng', version: '1.0.1'
```

TestNgVideoTest.class

```
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);
    }
}
```

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**:

TestNgRemoteVideonTest.class

```
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 by automation-remarks.com