Video Recorder

Table of Contents

[Build Status]

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

Documentation

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.+'
```

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);
    }
}
```

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.+'
```

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);
    }
}
```

Configuration

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

```
  VideoRecorder.conf()
                .withVideoFolder("custom_folder")         // Default is
${user.dir}/video.
                .videoEnabled(true)                       // Disabled video globally
                .withVideoSaveMode(VideoSaveMode.ALL)     // Save videos for passed
and failed tests
                .withRecorderType(RecorderType.FFMPEG)    // Monte is Default recorder
                .withRecordMode(RecordingMode.ANNOTATED)  // Record video only for
tests with @Video
                .withScreenSize(1024,768);                // Set screen size
```

or with maven

```
mvn test -Dvideo.folder=custom_folder // default video
         -Dvideo.enabled=false        // default true
         -Dvideo.mode=ALL             // default ANNOTATED
         -Drecorder.type=FFMPEG       // default MONTE
         -Dvideo.save.mode=ALL        // default FAILED_ONLY
         -Dvideo.frame.rate=1         // default 24
         -Dvideo.screen.size=1024x768 // screen size
```

FFMPEG recorder configuration

In order to use ffmpeg type recorder first you need to perform such steps:

Linux or Mac

Need to install ffmpeg recorder

On ubuntu you can do it using such command:

```
sudo add-apt-repository ppa:mc3man/trusty-media
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install ffmpeg
```

For Mac just use brew:

```
brew install ffmpeg
```

Windows

In case of Windows platform you need to download ffmpeg

Just download it and unzip to some folder on you PC. Example **C:\ffmpeg**

Then set System variable path for ffmpeg. Example

Example: add to PATH variable **;C:\ffmpeg\bin**

Also you need to download SendSignalCtrlC.exe utility and put into the same folder as ffmpeg.

The final result should be folder with **ffmpeg**, **SendSignalCtrlC.exe** utilities and System variable that point to this folder.

To be sure that everything works properly, open CMD and perform first command:

ffmpeg

The output shoul lopk like this:

```
C:\Users\sepi>ffmpeg
ffmpeg version N-81234-ge1be80a Copyright (c) 2000-2016 the FFmpeg developers
  built with gcc 5.4.0 (GCC)
  configuration: --enable-gpl --enable-version3 --disable-w32threads --enable-dxva2
--enable-libmfx --enable-nvenc --enable-avisynth --enable-bzlib --enable-libebur128
--enable-fontconfig --enable-frei0r --enable-gnutls --enable-iconv --enable-libass
--enable-libbluray --enable-libbs2b --enable-libcaca --enable-libfreetype --enable
-libgme --enable-libgsm --enable-libilbc --enable-libmodplug --enable-libmp3lame
--enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable
-libopus --enable-librtmp --enable-libschroedinger --enable-libsnappy --enable-libsoxr
--enable-libspeex --enable-libtheora --enable-libtwolame --enable-libvidstab --enable
-libvo-amrwbenc --enable-libvorbis --enable-libvpx --enable-libwavpack --enable
-libwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-libxvid --enable
-libzimg --enable-lzma --enable-decklink --enable-zlib
  libavutil      55. 28.100 / 55. 28.100
  libavcodec     57. 51.100 / 57. 51.100
  libavformat    57. 44.100 / 57. 44.100
  libavdevice    57.  0.102 / 57.  0.102
  libavfilter     6. 49.100 /  6. 49.100
  libswscale      4.  1.100 /  4.  1.100
  libswresample   2.  1.100 /  2.  1.100
  libpostproc    54.  0.100 / 54.  0.100
Hyper fast Audio and Video encoder
usage: ffmpeg [options] [[infile options] -i infile]... {[outfile options] outfile}...

Use -h to get full help or, even better, run 'man ffmpeg'
```

Than execute in CMD another command:

SendSignalCtrlC

Output:

```
C:\Users\sepi>SendSignalCtrlC
SendSignalCtrlC <pid>
  <pid> - send ctrl-c to process <pid> (hex ok)
```

If no errors, that everything set properly. You can you FFMPEG recorder type in ypur tests

Remote Video Recording:

Build remote module:

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

Run hub:

```
java -jar video-recorder-remote-x.x.jar -role hub
```

Run node:

```
java -jar video-recorder-remote-x.x.jar -servlets
"com.automation.remarks.remote.node.Video" -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.testng.GridInfoExtractor;
import com.automation.remarks.testng.RemoteVideoListener;
import com.automation.remarks.video.annotations.Video;
import com.automation.remarks.video.recorder.VideoRecorder;
import com.codeborne.selenide.Configuration;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Listeners;
import org.testng.annotations.Test;

import java.net.URL;

import static com.automation.remarks.testng.test.TestNGRemoteListenerTest.startGrid;
import static org.testng.Assert.fail;


@Listeners(RemoteVideoListener.class)
public class IntegrationTest {

    RemoteWebDriver driver;

    @BeforeClass
    public void setUp() throws Exception {
        URL hubUrl = new URL("http://localhost:4444/wd/hub");
        driver = new RemoteWebDriver(hubUrl, DesiredCapabilities.firefox());
        String nodeIp = GridInfoExtractor.getNodeIp(hubUrl,
driver.getSessionId().toString());
        VideoRecorder.conf()
                .withRemoteUrl(nodeIp);
    }

    @Test
    @Video
    public void test() throws InterruptedException {
        driver.get("http://automation-remarks.com");
    }
}
```

Recorder logging

You can log recorder events using log4j.

Just need to set DEBUG level fot **package com.automation.remarks.video.recorder**

Log4j settings example: with Console and File appenders.

```
log4j.rootLogger=INFO, CA, FA

#Console Appender
log4j.appender.CA=org.apache.log4j.ConsoleAppender
log4j.appender.CA.layout=org.apache.log4j.PatternLayout
log4j.appender.CA.layout.ConversionPattern=%-4r [%t] %-5p %c %x - %m%n

#File Appender
log4j.appender.FA=org.apache.log4j.FileAppender
log4j.appender.FA.File=ffmpeg.log
log4j.appender.FA.layout=org.apache.log4j.PatternLayout
log4j.appender.FA.layout.ConversionPattern=[%-5p] %d %c - %m%n

# Set the logger level of File Appender to DEBUG
log4j.logger.com.automation.remarks.video.recorder=DEBUG, FA
log4j.additivity.com.automation.remarks.video.recorder=false
```

License

See LICENSE.

Java Video Recorder Changelog

Java Video Recorder allows easily record video of your UI tests by just putting couple annotations.

This document provides a high-level view of the changes introduced in Asciidoctor by release. For a detailed view of what has changed, refer to the commit history on GitHub.

1.2 (2016-12-12) - @SergeyPirogov

Improvements

- Remote recorder updated to selenium version 3
- Video screen size option added
- JUnit test rule replaced by Junit test watcher

By automation-remarks.com