QR | Bar Codes in Selenium

Table of content

QR code

QR Code or Quick Response Code is a two-dimensional barcode that can be read by modern smartphones and special QR Code scanner devices.

> *A QR code consists of black squares arranged in a square grid on a white background.*

There are several variants of QR codes depending on their symbol size, layout, encoding, and structure.

QR Codes can be used to encode various types of data like - simple text, urls, phone numbers, sms, geolocation, email address, etc.

Xpath in Selenium

QR code Security

The text that is encoded using the QR Code can be read and interpreted by any smartphone and QR code scanner devices.

But, What if you don’t want everyone to read, what the QR Code is representing ?

QR Code itself does not provide any security, So follow below items to save your QR code from others / other devices

Before generating QR code with any text, encrypt the text using any cryptographic algorithm with a secret that only your application has access to.
Generate QR Code with the encrypted text.

Now, even if someone scans your QR Code, they will get the encrypted text which they can't decrypt unless they know the secret which was used to encrypt it.

But when you scan the QR Code in your application, you can decrypt the text using the secret and get the actual data stored in the QR Code.

> I am considering two apps (famous in India), PayTM and Tezz are providing QR codes, but we cannot read the QR code of a PayTM using Tezz, if we try it fails to parse the exact information, vice versa.

Note : I do not support online / mobile wallets, if everyone starts using the mobile wallet then Internet providing companies will increase the data charges.

Selenium Architecture

QR code Formats

Below are the few QR code formats, **PNG format will be more helpful** and they can be easily resized. This implies that you can scale the QR Code very easily depending on where you want it to be.

We would be reading QR code which is stored in Image format in this tutorial using selenium and java.

- HTML Code
- PNG File
- Tiff File
- SVG
- EPS
- PNG

Install QR code ZXing jars with Selenium :

In this part we would be discussing, how to read QR code using **java and selenium**, for reading QR code in Java, we need ZXing jars

For **Maven** Users:

```
<dependencies>
   <dependency>
      <groupId>com.google.zxing</groupId>
      <artifactId>core</artifactId>
      <version>3.3.0</version>
   </dependency>
   <dependency>
      <groupId>com.google.zxing</groupId>
      <artifactId>javase</artifactId>
      <version>3.3.0</version>
   </dependency>
</dependencies>
```

People who just want to download jars, they have to download two jar files:

*ZXing Core : **https://github.com/zxing/zxing/releases***

*ZXing Java SE Extensions : **https://mvnrepository.com/artifact/com.google.zxing/javase***

Try to download stable versions always rather than very latest versions, once download please add those jars into your Java Selenium Project in Eclipse

iFrames in Selenium

Reading QR Code in Selenium Java

We can read the QR code using Zxing Jar files, which we have integrated in the previous step.

Remember, You do not have to provide an exact image for the Reading QR code, but it should have all the Squares intact

While writing this tutorial, I had a thought, what if QR code image is bigger than the expected QR code size or what if the size is smaller

To get clarity on the above thing, I am presenting the Picasso in me, on below image on the steps

I have painted to show that how much disturbance can QR code can have without disturbing its content.

Steps to Read QR Code in Selenium :

1. Have the QR code image ready in your local machine, download below Image. I have stored **cherchertech** as the text in the bar code.

2. Create a File object for the image, so that the system understands the image as File.

```
File file = new File("C:\\PATH\\Desktop\\qrcode-selenium.png");
String decodedText = null;
```

3. So far we are calling a thing on the computer as an image, in above step we have converted that thing into File, Now we should make the system to understand that the File is Image. We use ImageIO classes to read the file into an image.

```
// store the file as an image
BufferedImage bufferedImage = ImageIO.read(file);
```

4. Now we have to clear all the other parts in the image and convert the image into a bitmap.

```
// process the image
```

```
LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));
```

5. Decode the details from the bitmap, and store it in the Result object. In the below code we are extracting only the text.

```
// store the details of the QR code
Result result = new MultiFormatReader().decode(bitmap);
decodedText =  result.getText();
```

6. Now
compare the text
retrieved from the image with the expected image.

```
// testng assertion
Assert.assertEquals(decodedText, "cherchertech");
```

Complete program for reading QR-Code in ***java selenium***

```
import com.google.zxing.*;
import com.google.zxing.client.j2se.BufferedImageLuminanceSource;
import com.google.zxing.common.HybridBinarizer;
import javax.imageio.ImageIO;
import org.testng.Assert;
import org.testng.annotations.Test;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
public class QRCodeReader {
    @Test
    public void QRTest() throws NotFoundException {
        try {
            File file = new File("C:\\PATH\\Desktop\\qrcode-selenium.png");
            String decodedText = null;

            // store the file as an image
            BufferedImage bufferedImage = ImageIO.read(file);

            // process the image
            LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
            BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));
            // store the details of the QR code
            Result result = new MultiFormatReader().decode(bitmap);
            decodedText =  result.getText();

            // print to console
            System.out.println("Decoded text = " + decodedText);
            // testng assertion
            Assert.assertEquals(decodedText, "cherchertech");

        } catch (IOException e) {
            System.out.println("Could not decode QR Code, IOException :: " + e.getMessage());
        }
    }
}
```

Custom Locators in Selenium

Reading QR Code image present on the website

We cannot expect that all the time we will have our QR code in our local system, Sometimes we may need to read the QR code from the image present on the website

Using Selenium Webdriver, we can read the images present in the website, or in remote url

For below example I have considered the QR Code present in the above example, you find the QR code using **id='picasso'**

Steps to Read QR code image present on the website :

1. **Navigate to the webpage** where the QR code image is present.

```
System.setProperty("webdriver.gecko.driver", "D:\\PATH\\geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
driver.get("https://chercher.tech/java/qrcode-barcode-selenium");
```

2. **Find the element** using its **locator** (used id locator for this example) 3. Every image will have an **src** attribute and this nothing but url on which image is present.

```
String src = driver.findElement(By.id("picasso")).getAttribute("src");
System.out.println("image url is : "+src);
```

4. Create an object for the URL class, using the src value.

```
URL urlOfImage = new URL(src);
```

5. Pass the URL class object to the ImageIO.read() method.

```
BufferedImage bufferedImage = ImageIO.read(urlOfImage);
```

The Complete code for reading a QR code present in the website using ***selenium webdriver***

```
@Test
public void QRCodeTestSelenium() throws NotFoundException, IOException {
   System.setProperty("webdriver.gecko.driver", "D:\\PATH\\geckodriver.exe");
   WebDriver driver = new FirefoxDriver();
   driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
   driver.get("https://chercher.tech/java/qrcode-barcode-selenium");

   String src = driver.findElement(By.id("picasso")).getAttribute("src");
   System.out.println("image url is : "+src);

   URL urlOfImage = new URL(src);
   BufferedImage bufferedImage = ImageIO.read(urlOfImage);

   LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
   BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));

   Result result = new MultiFormatReader().decode(bitmap);
   String textPresentInImage = result.getText();
   System.out.println("Text Present in Image : "+textPresentInImage);
   Assert.assertEquals(textPresentInImage, "cherchertech");
}
```

Parameterized Constructor in Selenium and Java

Read Screenshot QR Code in Selenium Webdriver

Sometimes you might have a website, where you don't have src attribute for the QR code image or the QR Code is not an image at all.

In such cases, we should take the **QR code element as Screenshot** and then we should create Image of it in Java using ImageIO class.

> **I strongly recommend you to read taking a screenshot of the element Tutorial**

Warning : You might encounter **"java.awt.image.RasterFormatException: (y + height) is outside of**

Raster" exception if the element is not inside the viewable area of the desktop, use scroll to bring the element to bring into view.

Read **JavascriptExecutor tutorial** to understand the scrollIntoView method.

Steps to Read QR code Screenshot :

1. **Navigate to the webpage** which has QR code

2. Find the image on the webpage and scroll to it, then get its location

3. **Take Webpage screenshot** and store it 4. Crop the image according to the QR code location

5. Pass the image to the ImageIO.read() method.

```
package listtest;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.concurrent.TimeUnit;
import javax.imageio.ImageIO;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.Point;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.Test;
import com.google.zxing.BinaryBitmap;
import com.google.zxing.LuminanceSource;
import com.google.zxing.MultiFormatReader;
import com.google.zxing.NotFoundException;
import com.google.zxing.Result;
import com.google.zxing.client.j2se.BufferedImageLuminanceSource;
import com.google.zxing.common.HybridBinarizer;
public class QRCodeWithSelenium {
   String pathOfImage = "D:\\screenshotOfElement.png";
```

```
    @Test
    public void QRCodeTestSelenium() throws NotFoundException, IOException, InterruptedExceptio
      System.setProperty("webdriver.chrome.driver", "D:\\PATH\\chromedriver.exe");
      WebDriver driver = new ChromeDriver();
      driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
      driver.get("https://chercher.tech/java/qrcode-barcode-selenium");

      JavascriptExecutor js = (JavascriptExecutor) driver;
      // scroll to particular element
      js.executeScript("document.getElementById('picasso').scrollIntoView(true)");
      Thread.sleep(10000);
      //Locate Image element to capture screenshot.
      WebElement element = driver.findElement(By.id("picasso"));
      //Capture entire page screenshot as File.
      //Used TakesScreenshot, OutputType Interface of selenium and File class of java to captur
      File screen = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
      //Used selenium Point class to get x y coordinates of Image element.
      //get location(x y coordinates) of the element.
      Point point = element.getLocation();
      int xcordinate = point.getX();
      int ycordinate = point.getY();
      //Used selenium getSize() method to get height and width of element.
      //Retrieve width of element.
      int imageWidth = element.getSize().getWidth();
      //Retrieve height of element.
      int imageHeight = element.getSize().getHeight();
      //Reading full image screenshot.
      BufferedImage img = ImageIO.read(screen);
      //cut Image using height, width and x y coordinates parameters.
      BufferedImage destination = img.getSubimage(xcordinate, ycordinate, imageWidth, imageHeig
      ImageIO.write(destination, "png", screen);
      //save Image screenshot In D: drive.
      FileUtils.copyFile(screen, new File(pathOfImage));

      BufferedImage bufferedImage = ImageIO.read(new File(pathOfImage));

      LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
      BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));

      Result result = new MultiFormatReader().decode(bitmap);
      String textPresentInImage = result.getText();
      System.out.println("Text Present in Image : "+textPresentInImage);
      Assert.assertEquals(textPresentInImage, "cherchertech");
    }
}
```

Take Web Page & Web Element Screenshot in selenium

Reading Complex QR-Code in Selenium

So far we have practiced with a simple text QR code, but in day to day life we will have little more **complex QR Code.**

The QR code will have Name, phone Number, Url, mail, Note may be some other fields as well, let's learn how to decode them

In the below image, I have Stored my details in meCard format, it is similar to vCard but meCard is more compact.

Steps to Read Complex QR-Code in Selenium Webdriver

1. Open Webpage , Find the QR code element and get src attribute.

2. Decode the QR code using ImageIO and other classes from ZXing jar

3. When getting the text, you will see some text stored in the QR code in Raw format.

4. I have removed the type from the Raw text using the replace method in Java

5. Parse the text based on the delimiter they have provided, I have used ':' (colon)

6. I have stored in the details in the map as keys and values for future purpose.

7. You should continue to proceed based on your requirement

```
package cherchertech;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.net.URL;
import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.TimeUnit;
import javax.imageio.ImageIO;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;
import com.google.zxing.BinaryBitmap;
import com.google.zxing.LuminanceSource;
import com.google.zxing.MultiFormatReader;
import com.google.zxing.NotFoundException;
import com.google.zxing.Result;
import com.google.zxing.client.j2se.BufferedImageLuminanceSource;
import com.google.zxing.common.HybridBinarizer;
public class QRCodeWithSelenium {

   @Test
    public void QRCodeTestSelenium() throws NotFoundException, IOException {
      System.setProperty("webdriver.gecko.driver", "D:\\PATH\\geckodriver.exe");
      WebDriver driver = new FirefoxDriver();
      driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
      driver.get("https://chercher.tech/java/qrcode-barcode-selenium.php");
      String src = driver.findElement(By.id("complex-qrcode")).getAttribute("src");
      System.out.println("image url is : "+src);
      URL urlOfImage = new URL(src);
      URL urlOfImage = new URL(src);
      BufferedImage bufferedImage = ImageIO.read(urlOfImage);
      LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
      BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));
      // store the details of the QR code
      Result result = new MultiFormatReader().decode(bitmap);
      String decodedText = result.getText();
      Map<String, String> contact = new HashMap<String, String>();
      // print to console
      System.out.println("Decoded content = " + decodedText);
      System.out.println("**************************************\n\n");
      String[] abc = decodedText.replace("MECARD:", "").split(";");
      for (String str : abc) {
         System.out.println(str);
         String[] content = str.split(":");
         contact.put(content[0], content[1]);
      }
      System.out.println("**************************************\n\n");
      //print the contact map
      System.out.println("Complete contact map :: ");
      System.out.println(contact);
    }
}
```

Up-Casting in Java and Selenium

Create QR Code

Similar to QR Reading, we can also create a QR Code images and supply it for other purposes; <**Creating QR code** is simpler than reading a QR code.

Steps to Create QR Code

1. Create an object of the QRCodeWriter class.

```
// create object to QR code write
QRCodeWriter qrCodeWriter = new QRCodeWriter();
```

2. Call encode method from the QRCodeWriter object, encode method accepts parameters

Text content to write into a QR code

QR code Type

QR code height

QR code width

```
// set the content, type, size
BitMatrix bitMatrix = qrCodeWriter.encode(QR_CODE_TEXT, BarcodeFormat.QR_CODE, 320, 320);
```

3. Write the QR code into the local system with the help of writeToPath method present in the **MatrixToImageWriter** class

```
// write the image content into the given file
MatrixToImageWriter.writeToPath(bitMatrix, "PNG", path);
```

Complete program to write the QR code in selenium java

```
package cherchertech;
import java.nio.file.FileSystems;
import java.nio.file.Path;
import org.testng.annotations.Test;
import com.google.zxing.BarcodeFormat;
import com.google.zxing.client.j2se.MatrixToImageWriter;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.QRCodeWriter;
public class QRCodeGeneratorSelenium {
    @Test
    public void generateQRCODE() throws Exception {
        String QR_CODE_TEXT = "QR Code generated by CherCher Tech";
        // create object to QR code write
        QRCodeWriter qrCodeWriter = new QRCodeWriter();
        // set the content, type, size
        BitMatrix bitMatrix = qrCodeWriter.encode(QR_CODE_TEXT, BarcodeFormat.QR_CODE, 320, 320);
        // set the file
        Path path = FileSystems.getDefault().getPath("C:\\PATH\\image_QR_Code.png");
        // write the image content into the given file
        MatrixToImageWriter.writeToPath(bitMatrix, "PNG", path);
    }
}
```

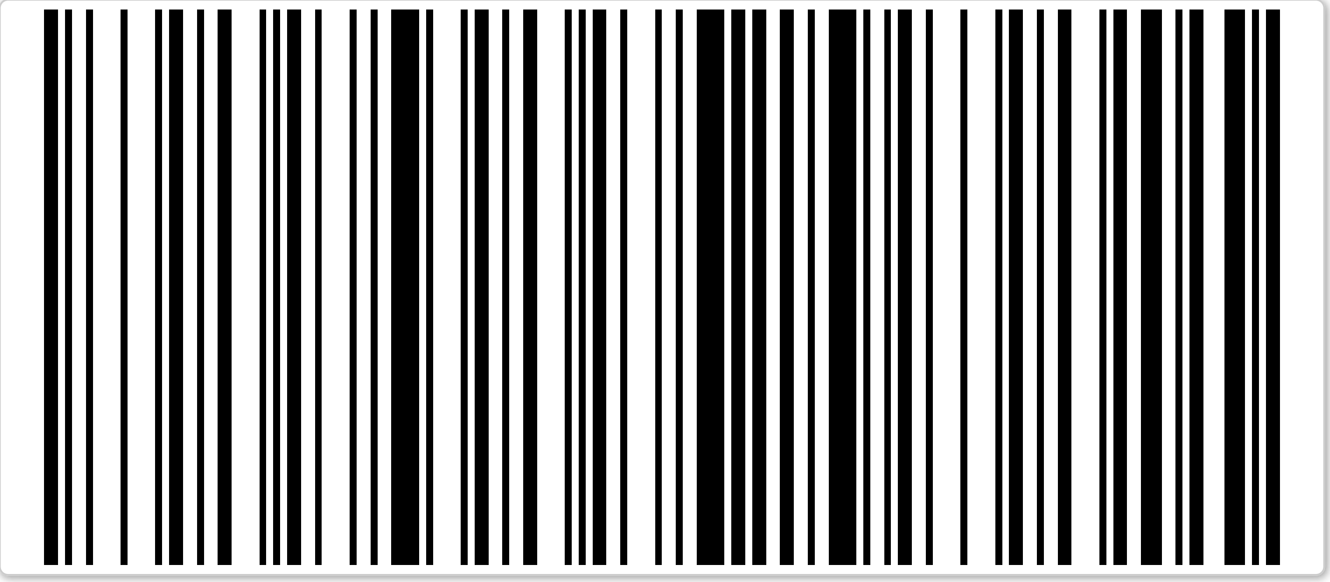
Compare Screenshots in selenium

Reading & Wring Barcodes in Selenium Java

A barcode is a square or rectangular image consisting of a series of parallel black lines and white spaces of varying widths that can be read by a scanner.

Barcodes are applied to products as a means of quick identification. They are used in Libraries, retail stores as part of the purchase process, in warehouses to track inventory, and on invoices to assist in accounting, among many other uses.

Barcode example:



Reading Barcode in Selenium :

There is no be much difference between QR code reading and **barcode reading**, the only difference is you need to provide the Barcode image path instead of QR code image.

```
@Test
public void QRTest() throws NotFoundException, IOException {
   BufferedImage bufferedImage = ImageIO.read(new File("C:\\PATH\\bar-code.png"));

   LuminanceSource source = new BufferedImageLuminanceSource(bufferedImage);
   BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));

   Result result = new MultiFormatReader().decode(bitmap);
   System.out.println(result.getText());
}
```

Writing Barcode in Selenium :

Similar to QR code generation, we can create Barcode using zxing jars, The only difference is, you need to create an object to Code128Writer class.

Call encode method from the Code128Writer class object and Content, Type of barcode, height, and width of the barcode.

```
@Test
   public void BAR_Code_Generator() throws Exception {
       Writer writer = new QRCodeWriter();

      //set Barcode format
       BitMatrix bitMatrix = new Code128Writer().encode("chercher tech", BarcodeFormat.CODE_128

       // write the barcode into file system
       MatrixToImageWriter.writeToStream(bitMatrix, "png", new FileOutputStream(new File("D://c

       System.out.println("Code128 Barcode Generated.");
   }
```

Note: Similar to QR code reading you can use all the different ways to read Bard like reading from the website and from url.

Dropdown Operations in selenium webdriver

Recommended Readings

- Selenium interview questions
- Selenium tricky interview questions
- selenium framework interview questions
- Core Java Interview Questions Set 1
- Selenium Relative Locators But I am hesitant to use
- Robot class in selenium
- Tags & Hooks in Cucumber with Selenium | BDD
- Parameterize Cucumber BDD with selenium