An orange and white text on a yellow background

Description automatically generatedsearch for this dependency.

A screenshot of a web page

Description automatically generated

Go to the mvn website.

A screenshot of a computer

Description automatically generated

Added the dependency in pom file-

A computer code with text

Description automatically generated

Extent.properties is the standard file we need to create-

Do not change anything in this. At the most we can change the “name or path of the file”.

A computer screen shot of a computer code

Description automatically generated

Config file which is again standard-

Here also you can change the values like change from dark to standard, name of the report and so on.

A screenshot of a computer

Description automatically generated

Test output folder will be generated after first run-

Currently it is not present because test has not run even once.

A screenshot of a computer

Description automatically generated

These are the notes for the extent.properties-

#first line should be true as we want to generate spark report

#location of the html report - second line

#config file has to be added - new file which we need to add

#report folder generated here - report location

#screenshot location

#Screenshot location

#pdf is true for pdf reports

#pdf report location

#what all should be seen in report

#machine details

#user details

#what is the build number

#app name

This is the note for extent-config.xml-

<!—what theme, encoding, protocol being used etc 🡪

Run the runner-

|  |
| --- |
| package com.myTestRunner;  import org.junit.runner.RunWith;  import io.cucumber.junit.Cucumber;  import io.cucumber.junit.CucumberOptions;  @RunWith(Cucumber.class)  @CucumberOptions(  features = { "src/test/resources/appFeatures/" },  glue = { "stepDefinitions", "appHooks" },  plugin = { "pretty",  "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"}//this adapter needed for the new extent reports.  )  public class MyTestRunner {  } |

Report folder generated-

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Right click on index and click properties.

Copy the path and paste in browser and nice report opens.

A screenshot of a computer

Description automatically generated

Go to the bug section on left side and we can see failures-

A screenshot of a computer

Description automatically generated

Clicking on the link in test name opens error-

A screenshot of a computer

Description automatically generated

Try toggling between the feature names-

A screenshot of a phone

Description automatically generated

We can see the other details also-

A screenshot of a computer

Description automatically generated

You can explore the html reports.

Amazing pdf report generated which you can explore-

A screenshot of a computer

Description automatically generated

In pdf report there will be no screenshot of failures. Screenshot present in spark html report.

Screenshot failure is also present in the other folder which we mentioned in properties-

A screenshot of a computer

Description automatically generated

# Project structure –

A screenshot of a computer program

Description automatically generated

# Issues related to version number of extent reports cucumber adapter-

When we change the version number to 2.6.0 it has vulnerabilities and gives issues.

A computer code with text

Description automatically generated

See in mvn repo it shows that there are vulnerabilities.

A screenshot of a computer

Description automatically generated

See pom also it throws this error.

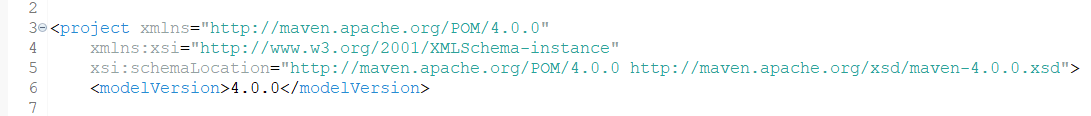
A screenshot of a computer

Description automatically generated

Change to 2.8.0 and it solves the issue-

A computer code with text

Description automatically generated



# Codes from this chapter-

A close up of text

Description automatically generated

Extent.properties-

|  |
| --- |
| extent.reporter.spark.start=true  extent.reporter.spark.out=test-output/SparkReport/Spark.html  extent.reporter.spark.config=src/test/resources/extent-config.xml  extent.reporter.spark.out=test-output/SparkReport/  screenshot.dir=test-output/  screenshot.rel.path=../  extent.reporter.pdf.start=true  extent.reporter.pdf.out=test output/PdfReport/ExtentPdf.pdf  #basefolder.name=reports  #basefolder.datetimepattern=d-MMM-YY HH-mm-ss  extent.reporter.spark.vieworder=dashboard,test,category,exception,author,device,log  systeminfo.os=Mac  systeminfo.user=Naveen  systeminfo.build=1.1  systeminfo.AppName=AutomationPractice |

Extent config.xml-

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <extentreports>  <configuration>  <!-- report theme -->  <!-- standard, dark -->  <theme>dark</theme>  <!-- document encoding -->  <!-- defaults to UTF-8 -->  <encoding>UTF-8</encoding>  <!-- protocol for script and stylesheets -->  <!-- defaults to https -->  <protocol>http</protocol>  <!-- title of the document -->  <documentTitle>Extent</documentTitle>  <!-- report name - displayed at top-nav -->  <reportName>Grasshopper Report</reportName>  <!-- location of charts in the test view -->  <!-- top, bottom -->  <testViewChartLocation>bottom</testViewChartLocation>  <!-- custom javascript -->  <scripts>  <![CDATA[  $(document).ready(function() {  });  ]]>  </scripts>  <!-- custom styles -->  <styles>  <![CDATA[  ]]>  </styles>  </configuration>  </extentreports> |

Test runner-

|  |
| --- |
| package com.myTestRunner;  import org.junit.runner.RunWith;  import io.cucumber.junit.Cucumber;  import io.cucumber.junit.CucumberOptions;  @RunWith(Cucumber.class)  @CucumberOptions(  features = { "src/test/resources/appFeatures/" },  glue = { "stepDefinitions", "appHooks" },  plugin = { "pretty",  "com.aventstack.extentreports.cucumber.adapter.ExtentCucumberAdapter:"}//this adapter needed for the new extent reports.  )  public class MyTestRunner {  } |