

# TestRigor Behavior Driven Test Case Creation First Usage Test Report

Author: Wayne Roseberry,

<https://www.linkedin.com/in/wayneroseberry/>

<https://waynemroseberry.github.io/>

I performed a testing session on TestRigor's Behavior Driven Test Case Creation feature. My intended purpose was to explore the feature as a first-time user, basing my activities entirely upon the documentation information presented to me. My intent was to create some test cases for my own page based on traffic to the page. I used this user scenario as a guide to explore TestRigor and report whatever issues I may find while doing it.

## Summary

Completion of testing was blocked by TestRigor saying there was no recorded data. Prior to recording I noticed documentation issues. Actual instrumentation of a web application was simple to do, but I was unable to confirm if it was working.

Specific issues reported in Testing Activity section.

## Details

### Preliminary Data

I created a suite in TestRigor using the free offering, so the suite is publicly viewable. The suite address is <https://app.testrigor.com/test-suites/5murppu3pdiWqz3nY/test-cases>.

The page I tested is located here: <https://datamakerjs-f3b6b7d13de0.herokuapp.com/>

The version of the page of the page with the TestRigor instrumentation applied is located in the following github address: [datatoolsjs/pages/mainindex.html](https://github.com/ff9bd2131f0e3e8afe3541691163cfd7244213f3) at [ff9bd2131f0e3e8afe3541691163cfd7244213f3 · WayneMRoseberry/datatoolsjs \(github.com\)](https://github.com/ff9bd2131f0e3e8afe3541691163cfd7244213f3)

### Test Activity

#### Exploring Documentation

I discovered the Behavior Driven Test Case Creation prior to testing. This is how I learned of the feature – from this URL: [testRigor - AI-Based Test Automation Tool](#)

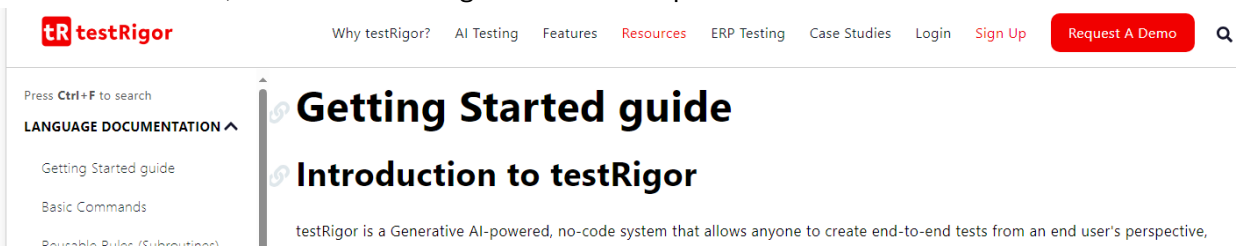
To mimic the behavior of a new user (which I am, coincidentally), I searched the page for a documentation link in context of the elements on the page. I did not find any.

**Issue 1.** The settings page for Behavior Driven Test Case creation offers no links to documentation about how to use the feature, something which presents other issues discovered later during testing.

I clicked on “Documentation” on the left menu navigation, this opened to

<https://testrigor.com/docs/language>

I scanned the left menu on the Documentation page – the table of contents – and did not see mention of the Behavior Driven Test Case Creation feature. I clicked on the search icon in the upper right corner (as per screen below) and attempted a search for “Behavior”, but this seemed to yield articles about BDD, and was searching outside the scope of the documentation.



I noticed the “Press CTRL+F to search” option in the upper left of the table of contents... and used that to search for “behavior”, which returned 3 hits inside the documentation, none of them pertaining to the Behavior Driven Test Case Creation feature.

I was able to find one mention of the feature on the “Top Features” page on the TestRigor website, but nothing instructional.

**Issue 2.** The TestRigor product documentation seems to offer no instructional material for how to use the Behavior Driven Test Case Creation feature.

I concluded that I should expect the settings page for the feature to be sufficient to guide me through correct and successful application of the feature.

## Instrumenting my website

I copied the code provided under “Add the following code in your page to start recording.” I left “Recording of user behavior enabled” to the default state of “on.” I changed “Allow recording to capture input values” to “on.” I changed “Maximum text length” to 1000. I set the Session cut-off to timeout to 24 hours. I set the “Paths to record from” to my website: <https://datamakerjs-f3b6b7d13de0.herokuapp.com/>

**Issue 3.** “Allow recording to capture input values” setting should provide privacy warnings. Input data is considered something sensitive to collect during instrumentation. Default to off is a good idea, but the user should be advised to be careful to turning this on. Depending on their application behavior, use of such a feature might fall under privacy regulation in certain jurisdictions, such as per GDPR. There was no text, instruction or warning provided.

**Issue 4.** “Maximum text length” and “Session cut-off timeout” do not have an explanation. What do these do? Maximum text length of what? What exactly is meant by a session? How should a user decide the right values for these settings? They are here, so they seem like they

should matter, but nothing tells the user what trade-offs they are making for what benefit. Would expect guidance and documentation.

**Issue 5.** The API key is hard-coded in the script example, something ill-advised for security concerns.

Most sites advise users do not put API keys in source code, particularly if it will be checked into a source repository, because if the repository is compromised then others can impersonate that application by use of their API key. Further, if the key is going into client side JavaScript (as opposed to server side) then the key is completely exposed. Unless this is the side effect of me using the free tier, or unless the API key for TestRigor is not considered a security risk to TestRigor itself, it seems it would be far better if the user were instructed to not put the API key in source code but use secrets and web settings instead and not to expose it in client-side script.

I added the code sample to my web application and tested local. I opened the developer console (Microsoft Edge, which runs on Chrome) and observed HTTP 403 errors in the console for <https://recorder.testrigor.com/v1/events/kOxmHzTpD2sdbRTivJ6i>.

**Issue 6.** Running a page modified with the TestRigor recording instrumentation code yields a 403 fetching on requests back to TestRigor.

I am not sure if I expect this to work or not. The setting on the test suite was for my website address, so I am wondering if this is correct behavior. Perhaps this is a security feature of TestRigor, rejecting the request because it is coming from a different URL than the API key is configured for? It does leave a question of how one is to test this behavior locally, but maybe that is just not a supported scenario. If so, then this might also be the mitigation for the risk of the compromised API key scenario above – but still, compromising API keys is half of the security package...

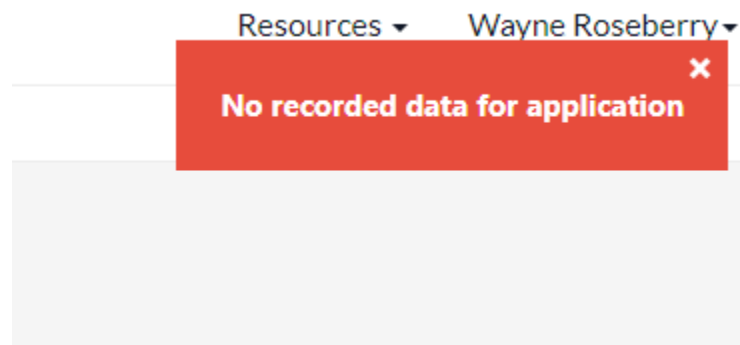
After confirming the page works and the script runs (despite the errors above – at least it is trying to execute), I updated my website at <https://datamakerjs-f3b6b7d13de0.herokuapp.com/> and confirmed the script is indeed on the page.

I loaded the page several times, selecting elements on the page that changed page content.

I went back to the following page on TestRigor: [testRigor - AI-Based Test Automation Tool](#)

I clicked “Trigger new test cases lookup”

I received the error “No recorded data for application”



**Issue 7.** Unable to create test cases from instrumented page - error “No recorded data for application.”

I looked in the developer console, and found no errors for the web page. I set a breakpoint, and the TestRigor code was running on page load. I captured a session in Fiddler and found the following requests indicating there was back and forth conversation happening between my web client and the TestRigor recorder services:

#	Result	Protocol	Host	URL	Body	Caching	Content-Type	Process	Comments	Custom
44	200	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/pages/mainindex.js						1,249
			msedge:20404	text/javascript						
45	200	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/pages/vizjswrapper.js						1,365,967
			msedge:20404	text/javascript						
46	304	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/api/namespaces					0	
			msedge:20404							
47	304	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/api/schemadefs?namespace=schemaexamples					0	
			msedge:20404							
48	200	HTTPS	functional.events.data.microsoft.com	/OneCollector/1.0/						9
			msedge:20404	application/json						
49	200	HTTPS	westeurope.livediagnostics.monitor.azure.com	/QuickPulseService.svc/ping?ikey=0386594d-de90-4d48-a7d0-0a69aaf16945					0	
				devenv:7724						
50	200	HTTP	Tunnel to	recorder.testrigor.com:443					0	
			msedge:20404							
51	200	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/api/schemadef?namespace=schemaexamples&schemaname=referenceschemaexample					201	
			msedge:20404	application/json; charset=utf-8						
52	200	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/api/schemadef/getrandomexample?namespace=schemaexamples&schemaname=referenceschemaexample&count=1					0	
			msedge:20404	627 application/json; charset=utf-8						
53	200	HTTPS	functional.events.data.microsoft.com	/OneCollector/1.0/						9
			msedge:20404	application/json						
54	200	HTTPS	datamakerjs-f3b6b7d13de0.herokuapp.com	/api/schemadef/dot?namespace=schemaexamples&schemaname=referenceschemaexample					302	
			msedge:20404	text/plain						
55	200	HTTPS	recorder.testrigor.com	/v1/events/gs8ztvqy1Bo5xP-gFQB9					0	no-cache, no-store, max-age=0, must-revalidate; Expires: 0
			msedge:20404							
56	200	HTTP	Tunnel to	recorder.testrigor.com:443					0	
			msedge:20404							
57	200	HTTPS	recorder.testrigor.com	/v1/events/gs8ztvqy1Bo5xP-gFQB9					0	no-cache, no-store, max-age=0, must-revalidate; Expires: 0
			msedge:20404							
58	200	HTTPS	recorder.testrigor.com	/v1/events/gs8ztvqy1Bo5xP-gFQB9					0	no-cache, no-store, max-age=0, must-revalidate; Expires: 0
			msedge:20404							
59	200	HTTP	Tunnel to	www.linkedin.com:443					1,362	msedge:20404
60	200	HTTPS	recorder.testrigor.com	/v1/events/gs8ztvqy1Bo5xP-gFQB9					0	no-cache, no-store, max-age=0, must-revalidate; Expires: 0
			msedge:20404							

At this point I concluded the test session, as all further planned activity depended on examination and exploration of the created test cases and their behavior during execution.