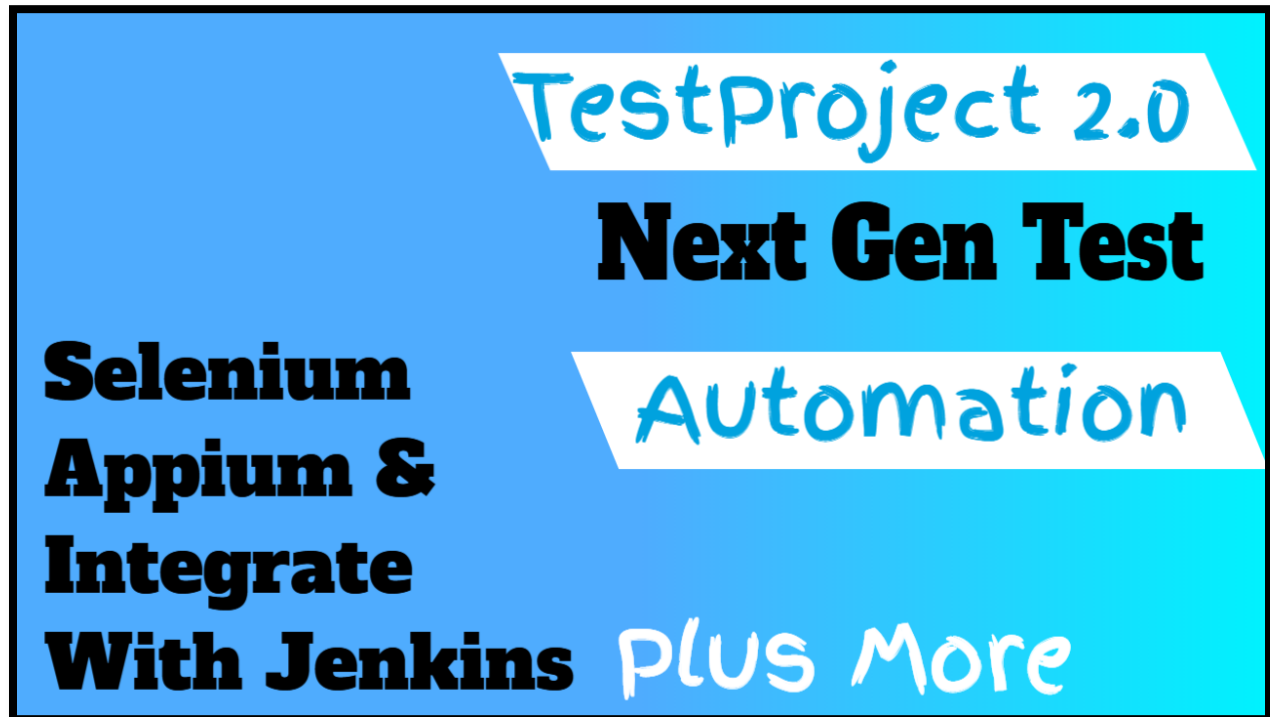


TestProject 2.0 Next Gen Release



Introduction

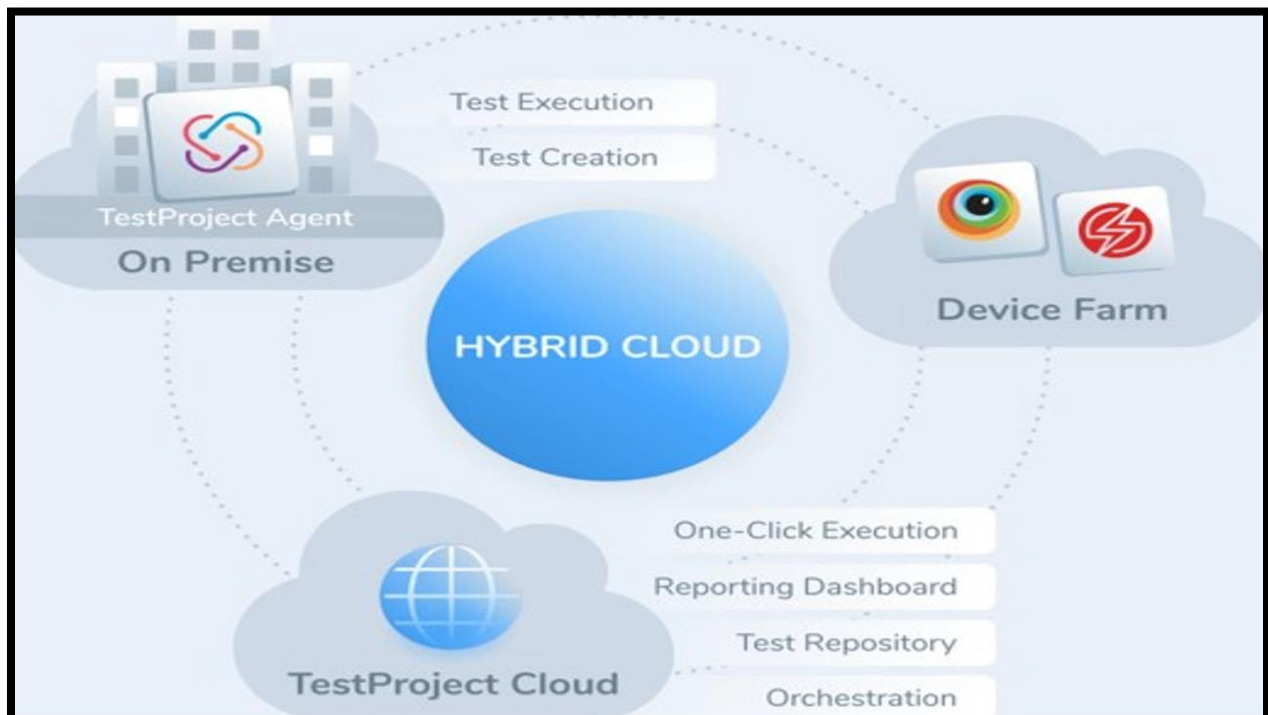
In this tutorial session, let's look at TestProject's 2.0 Next Gen Release. It has a lot of new features and capabilities. One of those features include working offline. In addition to this video, you can read more information from their blog site. Here is an [article](#) by Mark Announcing TestProject 2.0 Next Gen Release: Hybrid Cloud & Offline Mode. Also, you can go to their [2.0 version Documentation Release Notes](#). I will explain What Is Hybrid Cloud, the benefits of Hybrid Cloud, discuss then demo the Hybrid Cloud With Offline Mode, and the TestProject Agent CLI. CLI stands for Command Line Interface.

TestProject 2.0 Next Gen Release

- ▶ What Is Hybrid Cloud
- ▶ Benefits of Hybrid Cloud
- ▶ Hybrid Cloud With Offline Mode
- ▶ TestProject Agent (CLI) Command Line Interface

What Is Hybrid Cloud

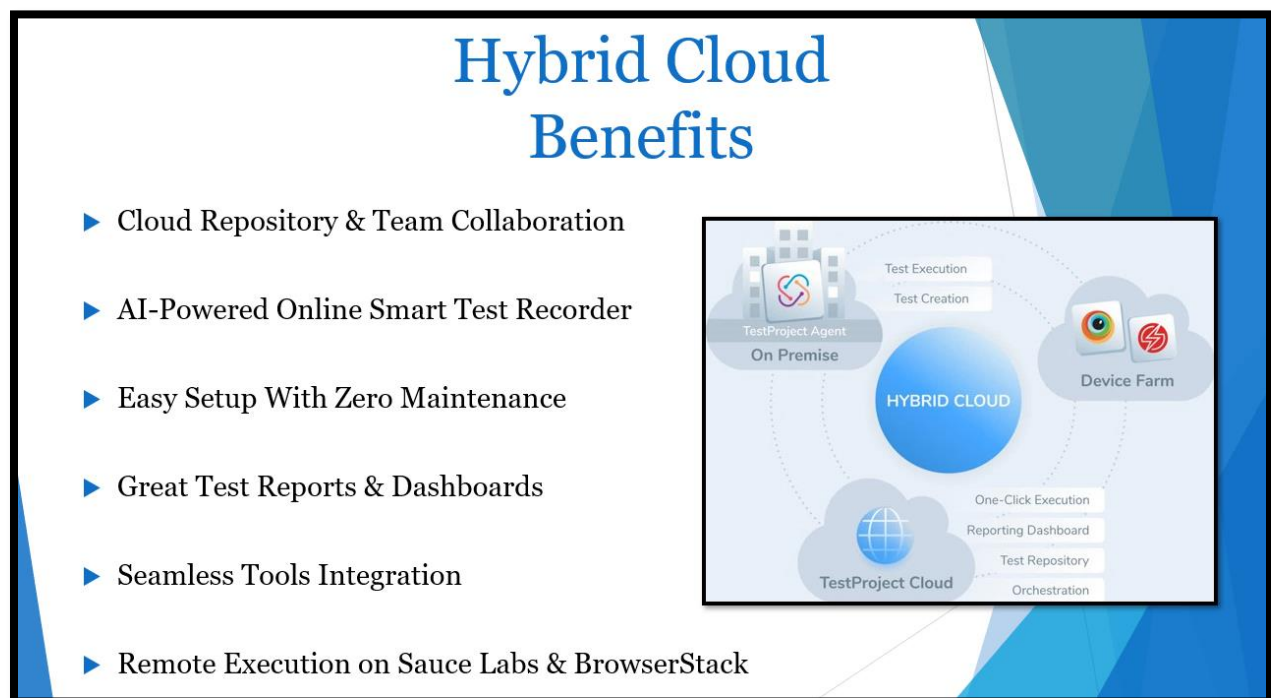
What is Hybrid Cloud? The Hybrid Cloud is a combination of 3 Components: TestProject's cloud repository, Agent, and Device Farm.



The cloud repository stores reports and test artifacts without including any data. We see in this screenshot, there is a One-Click Execution, Reporting Dashboard, Test Repository, and Orchestration between multiple platforms. The Agent is a local component with access to our local resources. It helps setup and maintain TestProject while assisting with creating our Test Scripts. The Test Scripts can be Web or Mobile. Selenium is for Web and Appium is for Mobile. A requested feature was for TestProject to operate offline. That means our local test can execute without connecting our agent to the cloud. The 3rd component is Device Farm which allow integration with cloud providers such as Sauce Labs, BrowserStack, AWS, and Azure.

Hybrid Cloud Benefits

Here's some of the Hybrid Cloud benefits: I mentioned how it's a cloud repository but it also promotes team collaboration. The Online Smart Test Recorder is powered by AI. Easy to setup with no maintenance. Great test reports and I think 5 or 6 Dashboards. The Dashboards are Velocity, Distribution, Platforms, Trends and Targets. So that's 5 Dashboards. The Hybrid Cloud also integrate with different tools like Jenkins, Slack, and qTest. Last but not least is Remote Execution on Sauce Labs and Browser Stack.



Hybrid Cloud With Offline Mode

The biggest change to TestProject 2.0 is Hybrid Cloud With Offline Model. We can create test, execute test, save test and more capabilities while offline. It's not a problem to continue with our test from our local source.

Create & Save An Offline Test

Next, let's look at how to create and save an offline test. Go to TestProject then select New Test. It's in 2 locations. Click Web and the Next button. Now, enter a name like Hybrid Offline Demo. Another new feature is Tags. We can add tags then search for the tags for our test and jobs. How about Hybrid, Test, Offline for the tag names?

Create test details

Provide your test a meaningful name and details, to help your colleagues understand the test purpose.

Name:

Hybrid Offline Demo

Description:

Test Description

Tags: i

Hybrid X

Test X

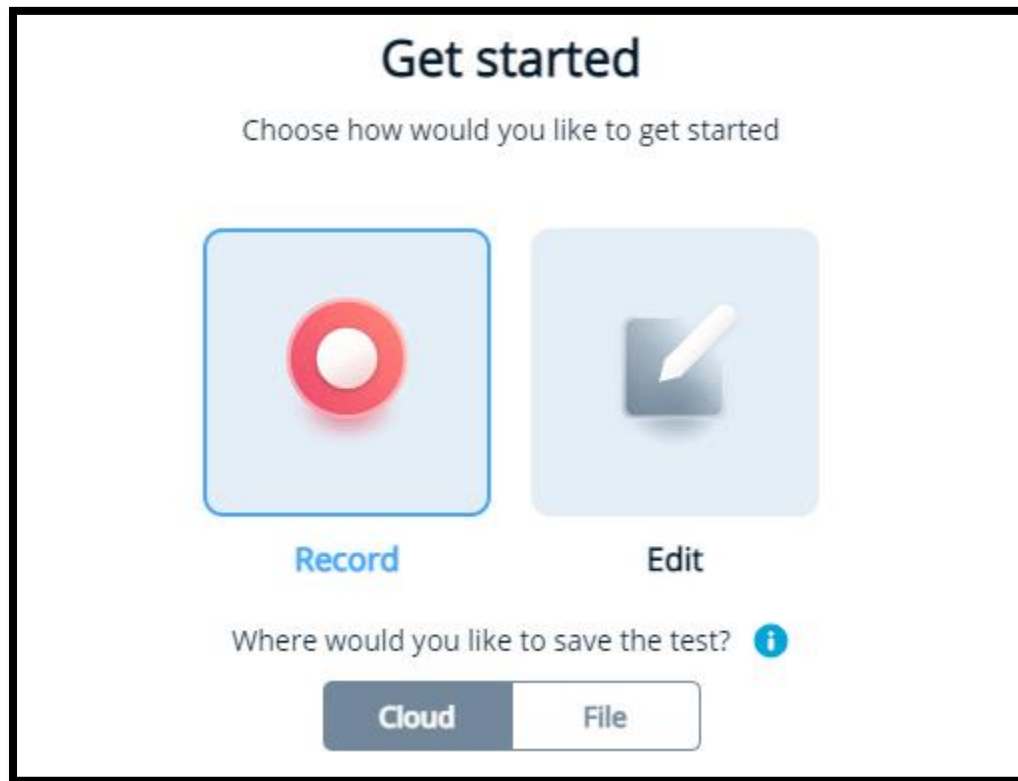
Offline X

Add tags

The Web Application will be TestProject Example Page. We have 2 options to save our test: Cloud or File. This icon explains both options. Cloud saves the test on TestProject's secure and SSL-encrypted cloud. File saves the test as a YAML file on our local machine. For this scenario, I will click File..

Cloud - Save tests on our secure and SSL-encrypted cloud to benefit from server maintenance, seamless collaboration and automatic deployment.

File - Save test as YAML file on your local machine with no footprint in the cloud to benefit from full local hosting control, versioning and offline experience.



Full Name is James File. Password is 12345 then click the Login button and Logout. Let's look at the step for typing our name. The Element section has CSSSELECTOR as a Locator with #name as the value. If we click the pencil icon to Edit the Element, we see CSS Selector - #name as the primary locator with 3 Backup Locators for XPATH. I'm going to cancel and for this test, Advance Options will let us always take a screenshot when typing the Full Name.

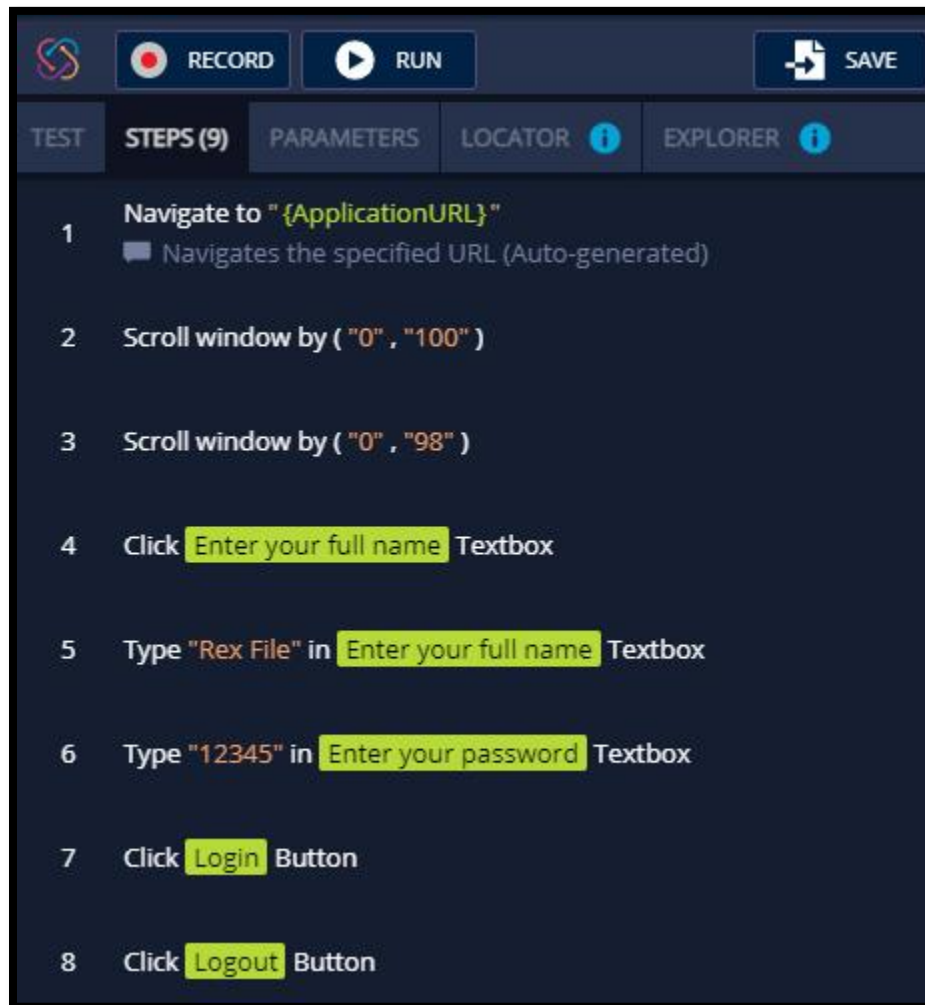
So for When To Take Screenshots? I will select Always, Save the Step. Notice this dialog has a Save option. It has a Save option because we chose File to save our test as a YAML file. I will save then go to my Downloads folder. We see Hybrid_Offline_Demo as the YAML filename. I'm going to open this file on my local machine to edit with Notepad ++. Save it again. For example, I will search for James then update James to Rex and save. Also, search for screenshot. One value says Failure. Another value says Always but most values say Inherit.

```

119     parameterMaps: []
120     conditions: []
121     validations: []
122     contexts: []
123     elementId: 3753d05450d19e837dcae6
124 - id: 5011cbf5ae8d8d95abb868
125   comments: ""
126   enabled: true
127   order: 5
128   repeat: 1
129   type: Action
130   settings:
131     sleepTime: -1
132     timeout: -1
133     sleepTiming: Inherit
134     failureBehaviorType: Inherit
135     takeScreenshotConditionType: Always
136   action:
137     source: System
138     id: 6f94fdaa-0041-4265-b73b-ecfcbe702d66
139     parameterMaps:
140     - direction: Input
141       name: keys
142       value: Rex File

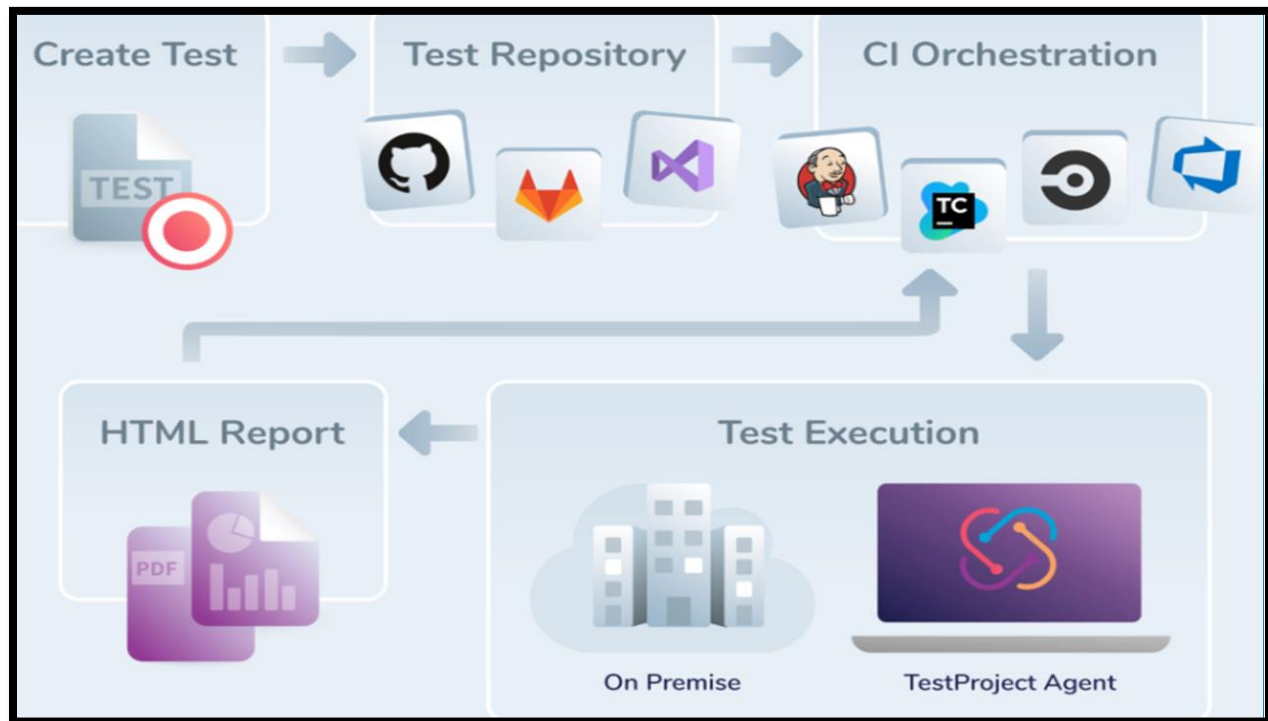
```

Let's also search for the #name locator value. We see the value and CSSSELECTOR locator and those 3 backup XPATH locators. Upload to TestProject then run the test and see if the name changed from James File to Rex File. We go back and select Open Test. Upload Hybrid_Offline_Demo then open to view the Test Steps. Full Name changed from James File to Rex File. That's it for creating and saving an offline test.



TestProject Agent CLI

Next, is the TestProject Agent CLI. The CLI is a cross operating system Command Line Interface Utility. Here's a screenshot of the process flow. The 1st 2 components were just covered in the previous session. We start by creating a test then saving the YAML file to my Downloads folder. The Downloads folder was my Test Repository. However, we can also copy the file to the GitHub repository or GitLab. Following the Test Repository is the CI Orchestration. The CI Orchestration allows us to integrate with tools like Jenkins to create a customized pipeline. I'm getting ready to show you how execute the test and view the HTML Report. We can execute when working offline or when there is no connection.



Going to my Downloads folder, I can type cmd to bring up my Command Prompt. To see the commands, we can type testproject-agent. This shows how to start the agent, connect, and other commands. Let's view the version by typing testproject-agent --version. It shows 2.1.0-RELEASE.

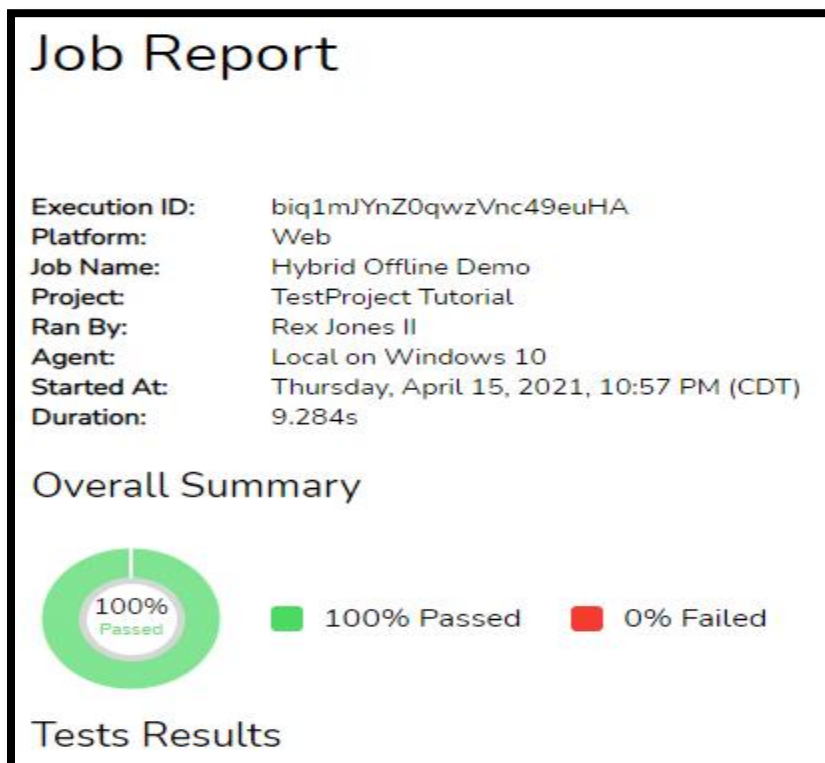
```
C:\Users\RexJo\Downloads>testproject-agent
Usage: testproject-agent [-v] [-a=<agent>] [COMMAND]
  -a, --agent=<agent>   Target Agent (host:port)
  -v, --version          Print version information and exit
Commands:
  start                Starts the Agent
  connect              Connects to an Agent and prints its version
  register              Registers an Agent using provided API or Development token
  browsers              List available browsers
  devices              List and query available devices
  validate              Inspects and validates an execution package file(s)
  run                  Runs an execution package file(s)
  help                 Displays help information about the specified command

C:\Users\RexJo\Downloads>testproject-agent --version
2.1.0-RELEASE
```


We can also view a list of our available browsers `testproject-agent -browsers`. I have 4 browsers: EDGE, CHROME, FIREFOX, and IE. Okay,

```
C:\Users\RexJo\Downloads>testproject-agent browsers
+=====+
| type           | version           |
|=====|=====|
| EDGE           | 89.0.774.76       |
| CHROME         | 89.0.4389.128     |
| FIREFOX        | 87.0.0.7747       |
| INTERNET_EXPLORER | 11.0.19041.1     |
+=====+
```

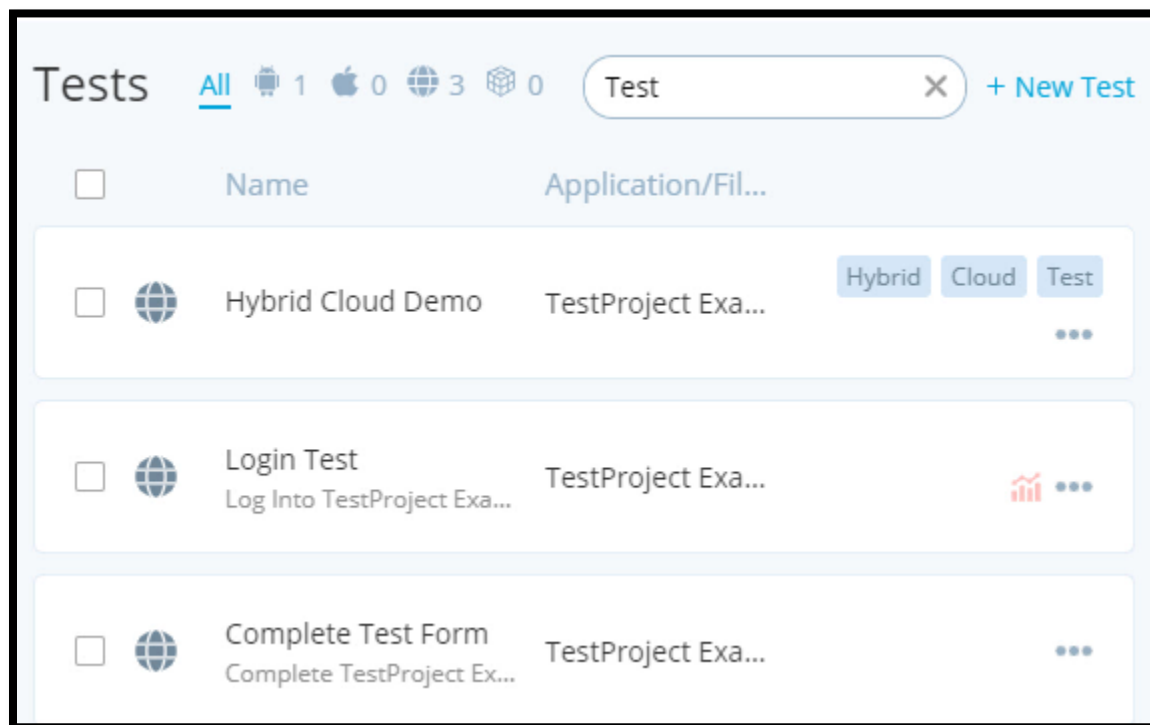
I'm going to run the save yaml test by writing `testproject-agent run` then drag the file `Hybrid_Offline_Demo`. Processing Execution. It's getting ready to run. Bingo, that's it. We can also view the report without being connected to TestProject Cloud Repository. Copy this url then paste it in the browser. Job Report shows the Overall Summary, Test Results, and Detailed Report with a screenshot. That's how we create a test, save the test, and execute the test offline.



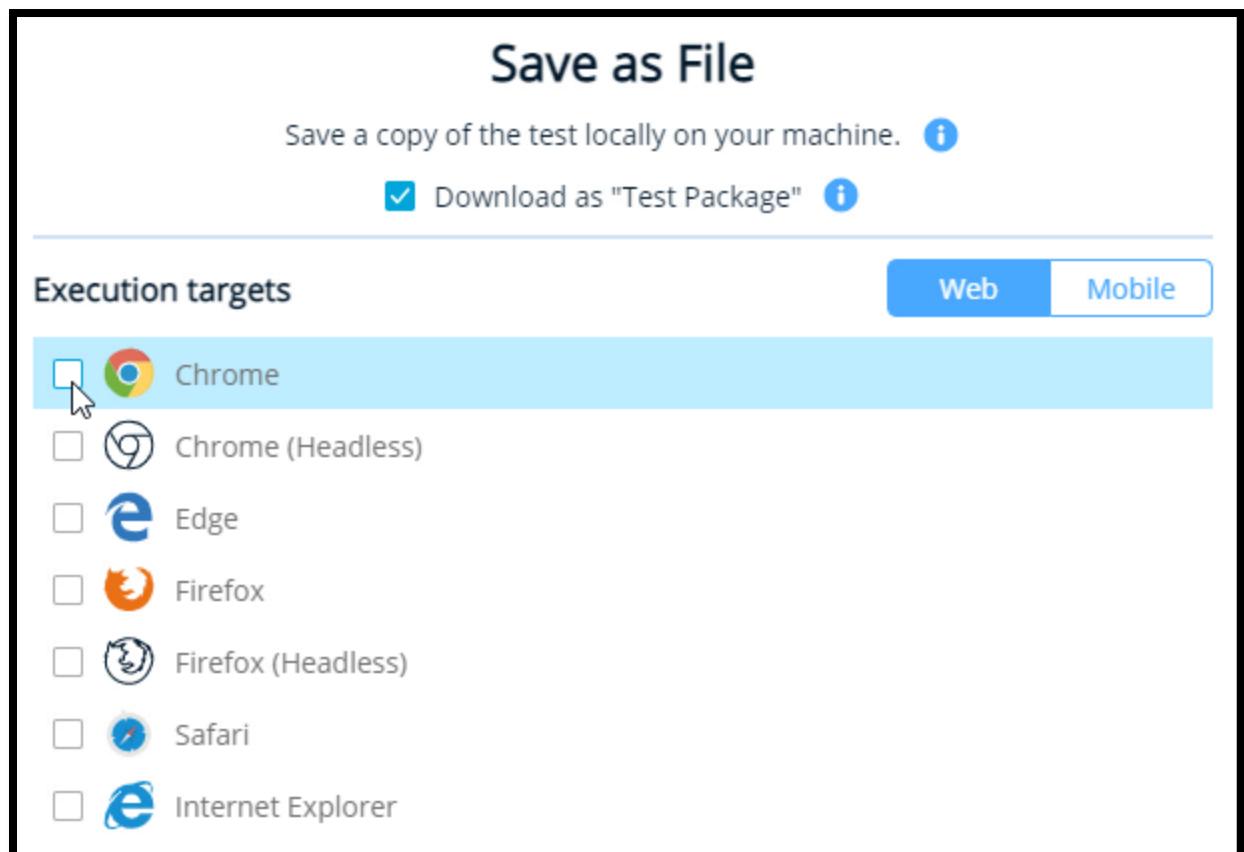
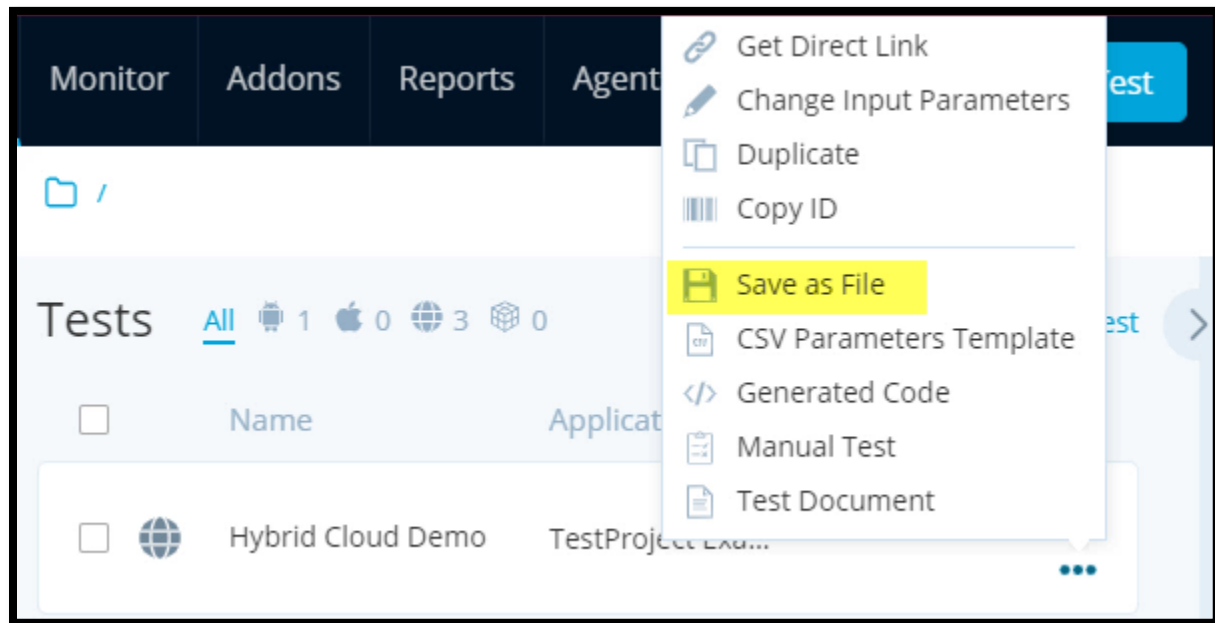
Create & Export A Cloud Test

Next, let's create and export a cloud test. Go back to our Project. New Test, Web, the name will be Hybrid Cloud Demo. Tags Hybrid, Cloud, Test then Next. The same Web Application TestProject Example. Record, Cloud, then Start Recording. Perform the same steps but this time let's add a Parameter for Full Name. Full Name will be John Doe. Password 12345. Login then Logout. Did you recognize how this dialog does not have the Save option? It's only because I clicked Cloud for saving this file. We can add a Parameter by clicking the step for typing John Doe then scrolling down to John Doe. Click the field then Use Parameter. On this dialog, click + to add a New Test Parameter. Name is Name with a value like Jane Doe then Save. Make sure to click the Name parameter and erase the hard coded John Doe value. The TEST tab will let us add a screenshot for all steps. When to take screenshots? Always then Save.

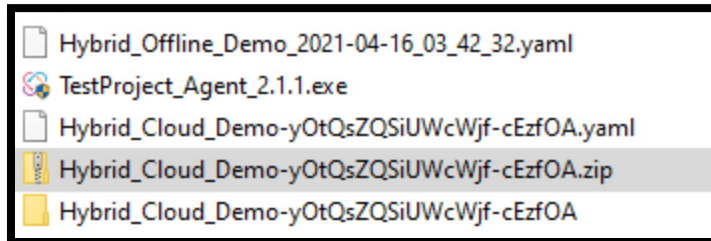
Here's 3 more new features, I wanted to show you. One feature is adding a tag then searching for a tag. Tags can be added to a test or job. Let's search for Test and 3 tests shows up. 1 of the test has Test as a Tag and the other 2 has Test in its name: Login Test and Complete Test Form.



The next features are for exporting the test. Click these 3 dots and we now have Save as File so we can have a copy locally on our machine. If I do not click the checkbox then only 1 file will be exported. Save and it goes to my Downloads folder. Click the ellipsis again then Save as File. The checkbox for Download as "Test Package" exports a zip file after selecting a browser like Chrome then Save.



Go to my Downloads folder. Do you see the single file Hybrid_Cloud_Demo? That's the file when I did not select the checkbox. If I open it then you will see the same data as the Hybrid_Offline_Demo YAML file. So, let's open the zip file package.



Extract All and we see 4 files. The test-parameters file let us add more values. Copy, Paste, then add Joe Doe as a value. We can make this a Data Driven Test by copying, pasting, then adding more information like Joe Doe. I'm not going to save but that's how it works for Data Driven Testing.

The settings.yaml file has Chrome browser because that's the only file I selected for this zip file. We can update it to include – Edge. The package.yaml looks just like the other yaml file that I executed using CLI. Let's get ready to execute this zip file using the CLI. Since I updated 2 files, I'm going to copy both files then add them to the zip file.

We use the same command to run the zip file. testproject-agent run then drag the zip file Hybrid Cloud Demo. Chrome is executing the steps and now Edge is executing the steps.

```
C:\Users\RexJo\Downloads>testproject-agent run C:\Users\RexJo\Downloads\Hybrid_Offline_Demo_2021-04-16_03_42_32.yaml
Processing execution...
Executing target 1/1: [Chrome 89.0.4389.128]
Tests 100% [=====] 1/1 (0:00:00 / 0:00:00)
Steps 100% [=====] 9/9 (0:00:09 / 0:00:00)
Execution Report: C:\Users\RexJo\AppData\Roaming\TestProject\Agent\reports\Hybrid Offline Demo-20210415-225736.html
Execution is complete.

C:\Users\RexJo\Downloads>testproject-agent run C:\Users\RexJo\Downloads\Hybrid_Cloud_Demo-yOtQsZQSiUWcWjf-cEzFOA.zip
Processing execution...
Executing target 1/2: [Chrome 89.0.4389.128]
Tests 100% [=====] 1/1 (0:00:00 / 0:00:00)
```

We take this url and view the report. This report has a screenshot for every step. That's it for TestProject's 2.0 Next Gen Release. Thanks for watching and I will see you in the next session.