



# Test Project

Test Automation Framework

# Contents

- ▶ Overview
- ▶ Features
- ▶ Framework
  - ▶ Installation & Setup
  - ▶ Supported Environments
  - ▶ Hybrid Cloud & Offline Mode
  - ▶ Actions & Validations
  - ▶ Self Healing
- ▶ Demo
- ▶ Limitations
- ▶ Q & A

# Overview

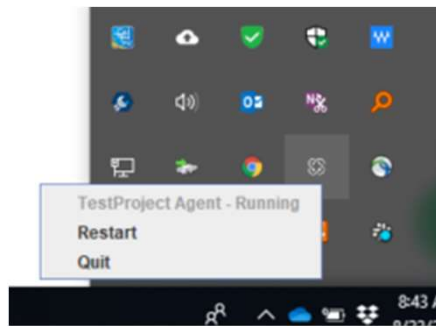
1. It provides users with powerful record and playback capabilities along with a developer SDK and the ability to use and create addons to extend the capabilities as needed.
2. TestProject is built on top of open source automation tools like Selenium and Appium
3. With the installation of single executable you have the ability to use all browsers and devices on a computer and you can instantly start recording, creating and executing tests.
4. TestProject is a community driven tool and has a free forever plan
5. For more details, browse through <https://docs.testproject.io/>

# Features

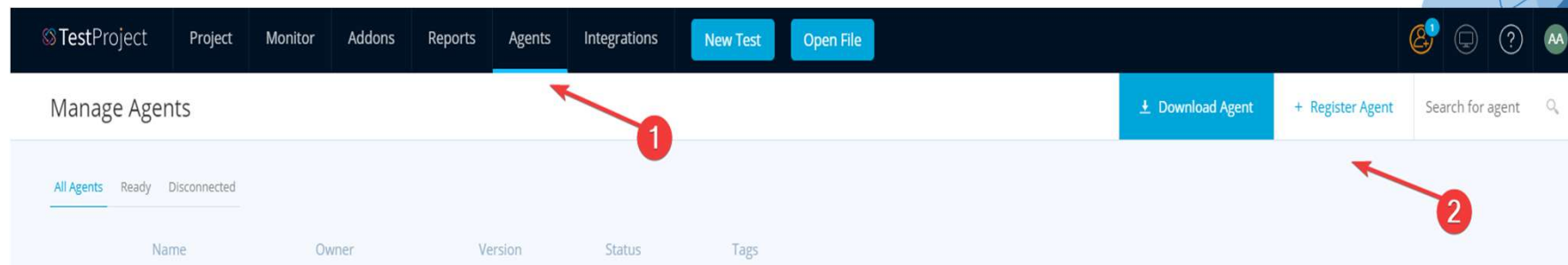
1. 100% free test automation development framework designed for agile teams
2. Advanced built-in recording capabilities
3. AI powered self healing technology
4. Supports iFrames, Popups and Dynamic Elements
5. With built in Adaptive-wait technology it automatically handles deviations in web application loading times
6. Hybrid cloud and offline modes
7. Execution through CLI Agent
8. Automation assistant

# Installation & Setup

1. Create a free account
2. Install and setup agent



3. Register the agent – Via UI



# Supported Environments

## Supported Browsers

- **Installed Browsers (local)**
- Chrome 77+ , Firefox 60+ , Safari 12+ , Edge 44+, Internet Explorer 11

## Supported Devices

- **Recording and Execution**
- Android: 6.0 ( Marshmallow) - 12.0 / API level 23+
- iOS: 10 - 15.0
- **Supported Emulators/Simulators Frameworks**
- Android Studio: Latest
- Genymotion: Latest
- XCode: 10-12.5

## Supported Agent Operating Systems (x64 only)

- Windows 7, 8, 8.1, 10
- macOS 10.14+
- Ubuntu 16.04, 18.04, 20.04

# Hybrid Cloud & Offline Mode

## **Hybrid Cloud Mode:**

1. With the hybrid cloud approach, test artifacts and reports on them will be stored in TestProject's safe and secure cloud environment.
2. Recording and execution of the tests themselves happen locally with the TestProject Agent
3. Access to Test Reports and dashboards along with the ability to do remote test execution

# Hybrid Cloud & Offline Mode

## Offline Mode:

1. More control of your tests using your own private environment
2. Strict compliance and security regulations and so you need to test in an environment that is completely disconnected from the internet
3. Create, edit and debug test with TestProject's in-browser recorder. However, instead of saving your test artifacts in the cloud they will be saved locally.
4. Debug and summarize test results using local HTML test reports that are automatically created for you.
5. TestProject Agent Command Line Interface (CLI) to execute tests



# Execution - Offline Mode

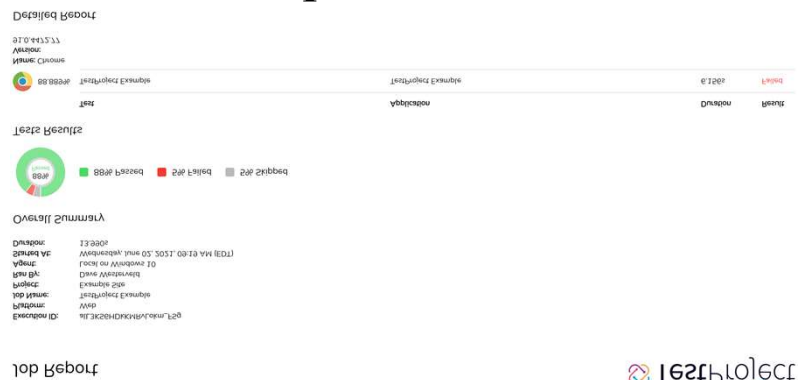
## 1. Executing the test file

*testproject-agent.exe run <name of downloaded file>*

## 2. View Test Results

```
Tests 100% [=====]
Steps 100% [=====]
Execution Report Path: C:\Users\ldwesterveld\AppData\Roaming\TestProject\Agent\reports\TestProject Example-20210602-091933.html
Execution Report URL: http://localhost:8585/api/report/TestProject Example-20210602-091933.html
Execution is complete.
```

## 3. Execution Report



# TestProject Agent - Help

```
C:\Users\nikhi>testproject-agent --help
Usage: testproject-agent [-hv] [-A=<agent>] [-C=<grpcTimeout>] [-D=<dataPath>]
                        [COMMAND]
  -A, --agent=<agent>    Target Agent (host:port)
  -C, --connect-timeout=<grpcTimeout>
                        Connection timeout in seconds
  -D, --data-path=<dataPath>
                        Path to Agent's data folder
  -h, --help             Print usage information.
  -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)
```

# TestProject Agent - Run

```
Runs an execution package file(s)
  <files>...      Path to execution package file(s)
-a, --alias=<alias>    Alias to use when naming the Agent
-A, --agent=<agent>    Target Agent (host:port)
-C, --connect-timeout=<grpcTimeout>
                    Connection timeout in seconds
-d, --driver-server=<servers>
                    Driver server - local within the Agent (default)
                    or cloud.
-D, --data-path=<dataPath> Path to Agent's data folder
  --grpc-address=<grpcAddress>
                    Agent's spawned gRPC server address (host:port)
-h, --help            Show this help message and exit.
-o, --output-path=<outputPath>
                    Output folder path for the local report file(s)
-p, --parameters=<parametersFile>
                    Parameters file (data-source CSV/YAML)
  --parallel-targets    Execute package(s) in parallel on all configured
                        browsers/devices
  --parallel-tests      Execute all tests in package(s) in parallel on all
                        configured browsers/devices
  --plain              Produce plain output (without progress bars).
-r, --report=<report>  Report type - local, cloud or both (default)
-rd, --restart-driver  Restart driver before each test
  --rest-address=<restAddress>
                    Agent's REST API address (host:port)
-s, --settings=<settingsFile>
                    Execution settings file (targets, etc.)
-t, --token=<token>    Token to use for registration
-V, --version          Print version information and exit.

Targets
  --browser=<browsers>  Target browsers (multi-value)
  --device=<devices>   Target devices (multi-value)
```

# TestProject Agent Details

TestProject

Project

Monitor

Addons

Reports

Agents

Integrations

New Test

Open File

Cheer

NJ

Manage Agents

Download Agent




+ Register Agent

Search for agent

All Agents

Ready

Disconnected

	Name	Owner	Version	Status	Tags	
	 DemoAgent	Nikhil Jain	3.1.0	Ready	 New Version Available	
General	General		Browsers	IP Addresses	Frameworks	
Devices	Agent Name	DemoAgent	Edge	94.0.992.38	Java	13.0.1
	Public Agent		Internet Explorer	11.0.19041.1202	ADB	1.0.41
Tags	Active / Max Workers	0 / 1	Chrome	94.0.4606.81		
Messages	Computer Name	LAPTOP-R8LP1L22				
	Environment	7.70GB of RAM				
Executions	Processor Cores	8				
	OS Type	Windows				

# Actions and Validations

Available Actions:

<https://docs.testproject.io/getting-started/available-actions>

Available Validations:

<https://docs.testproject.io/getting-started/available-validations>  
<https://docs.testproject.io/getting-started/step-output-validations>



# Self Healing

TestProject supports the self healing features to optimize the test case execution if any change is detected in the properties of the web element. Below is the process of self healing functionality in TestProject.

1. While recording, multiple locator strategies of the web element are saved in the test case. e.g. CSS-ID, CSS-Name, Xpath etc.
2. By default, web element is identified with the first locator strategy in the list
3. If there is any change in the properties and element is not identified with the first strategy, then element is identified with 2<sup>nd</sup> strategy and so on
4. Next time the working locator strategy is brought on top priority

*Note: Locator properties are not updated during execution*

# Limitations

1. 2 GB Free space (Need to connect their support team to get more storage and prices)
2. Agent needs to be installed manually to upgrade if there is any new agent version is released
3. Big Test Case yaml files (More than 50 MB) are not supported
4. Only CSV data source is supported
5. In case of unregistered agent and offline mode, version mismatch might occur, between the installed browser or the connected device and the bundled driver version.
6. Manual driver update is required in offline mode

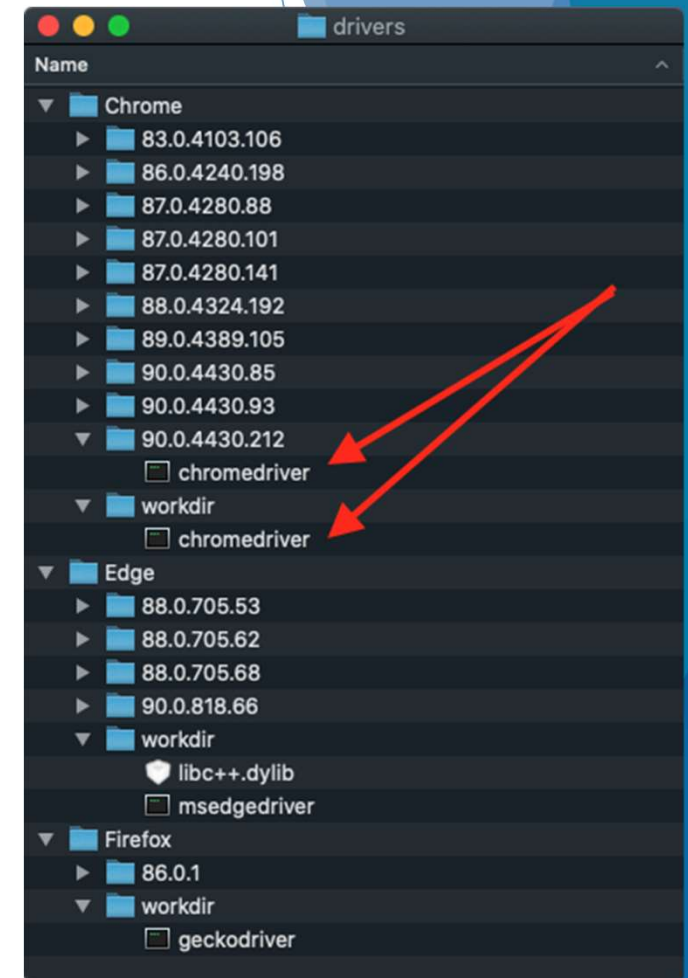
# Limitations Continued..

Agent stores the drivers in:

Windows -

%appdata%\TestProject\Agent\deps\drivers

1. Download the relevant driver and place it in the right location.
2. Determine browser version:
3. Create a "version" folder under the Browser folder with the exact version number
4. Extract the driver executable and place it into the version folder
5. Place the same executable in the workdir folder





Q & A



Thank You

