# 1. Installation

## 1.1. Introduction

Selenium Python bindings provides a simple API to write functional/acceptance tests using Selenium WebDriver. Through Selenium Python API you can access all functionalities of Selenium WebDriver in an intuitive way.

Selenium Python bindings provide a convenient API to access Selenium WebDrivers like Firefox, Ie, Chrome, Remote etc. The current supported Python versions are 3.5 and above.

This documentation explains Selenium 2 WebDriver API. Selenium 1 / Selenium RC API is not covered here.

## 1.2. Installing Python bindings for Selenium

Use pip to install the selenium package. Python 3 has pip available in the standard library. Using *pip*, you can install selenium like this:

```
pip install selenium
```

You may consider using virtualenv to create isolated Python environments. Python 3 has venv which is almost the same as virtualenv.

You can also download Python bindings for Selenium from the PyPI page for selenium package. and install manually.

## 1.3. Instructions for Windows users

1. Install Python 3 using the MSI available in python.org download page.
2. Start a command prompt using the `cmd.exe` program and run the `pip` command as given below to install *selenium*.

   ```
   C:\Python39\Scripts\pip.exe install selenium
   ```

Now you can run your test scripts using Python. For example, if you have created a Selenium based script and saved it inside `C:\my_selenium_script.py`, you can run it like this:

```
C:\Python39\python.exe C:\my_selenium_script.py
```

## 1.4. Installing from Git sources

To build Selenium Python from the source code, clone the official repository. It contains the source code for all official Selenium flavors, like Python, Java, Ruby and others. The Python code resides in the `/py` directory. To build, you will also need the Bazel build system.

> **Note:**
>
> Currently, as Selenium gets near to the 4.0.0 release, it requires Bazel 3.2.0 (Install instructions), even though 3.3.0 is already available.

v: latest

To build a Wheel from the sources, run the following command from the repository root:

```
bazel //py:selenium-wheel
```

This command will prepare the source code with some preprocessed JS files needed by some webdriver modules and build the `.whl` package inside the `./bazel-bin/py/` directory. Afterwards, you can use `pip` to install it.

## 1.5. Drivers

Selenium requires a driver to interface with the chosen browser. Firefox, for example, requires geckodriver, which needs to be installed before the below examples can be run. Make sure it's in your *PATH*, e. g., place it in */usr/bin* or */usr/local/bin*.

Failure to observe this step will give you an error *selenium.common.exceptions.WebDriverException: Message: 'geckodriver' executable needs to be in PATH.*

Other supported browsers will have their own drivers available. Links to some of the more popular browser drivers follow.

| | |
|---|---|
| **Chrome**: | https://sites.google.com/chromium.org/driver/ |
| **Edge**: | https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/ |
| **Firefox**: | https://github.com/mozilla/geckodriver/releases |
| **Safari**: | https://webkit.org/blog/6900/webdriver-support-in-safari-10/ |

For more information about driver installation, please refer the official documentation.

## 1.6. Downloading Selenium server

> Note:
>
> **The Selenium server is only required if you want to use the remote WebDriver**. See the Using Selenium with remote WebDriver section for more details. If you are a beginner learning Selenium, you can skip this section and proceed with next chapter.

Selenium server is a Java program. Java Runtime Environment (JRE) 1.6 or newer version is recommended to run Selenium server.

You can download Selenium server 2.x from the download page of selenium website. The file name should be something like this: `selenium-server-standalone-2.x.x.jar`. You can always download the latest 2.x version of Selenium server.

If Java Runtime Environment (JRE) is not installed in your system, you can download the JRE from the Oracle website. If you are using a GNU/Linux system and have root access in your system, you can also use your operating system instructions to install JRE.

If *java* command is available in the PATH (environment variable), you can start the Selenium server using this command:

```
java -jar selenium-server-standalone-2.x.x.jar
```

Replace *2.x.x* with the actual version of Selenium server you downloaded from the site.

If JRE is installed as a non-root user and/or if it is not available in the PATH (environment variable), you can type the relative or absolute path to the *java* command. Similarly, you can provide a relative or absolute path to Selenium server jar file. Then, the command will look something like this:

```
/path/to/java -jar /path/to/selenium-server-standalone-2.x.x.jar
```

v: latest