Rex Jones II

Selenium 4 Intro
To Chrome DevTools Protocol

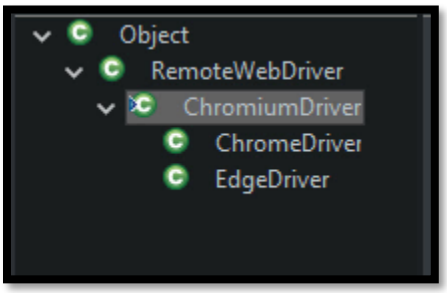
Hello and Welcome, in this session, I will talk about CDP which is an acronym for Chrome Debugging Protocol. It's a new Selenium 4 API feature that's designed for debuggers. All browsers built on the Chromium platform has an option for Developer Tools. Therefore, another name for CDP can be called Chrome DevTools Protocol.

If I go to Chrome then click this vertical ellipsis, navigate to More Tools, and at this point, we see Developer Tools. DevTools is short for Developer Tools. Click and we see a few tabs. Elements is the most popular tab for Automation but we also have Console, Sources, Network, Performance, Memory Application, and Security. It's the same panel if I right click a page and click Inspect.

Microsoft Edge also has these same tabs after accessing Developer Tools. We click this ellipsis, More Tools and here's Developer Tools.

Selenium 4 provides a way for us to take advantage of Chrome and Microsoft Edge's debugging protocol. Let's look behind the scenes at the classes and methods. Starting with selenium-chromium-driver, we see the ChromiumDriver class, open the Type Hierarchy. Notice, the ChromeDriver and

EdgeDriver classes are both under the ChromiumDriver class. That's because ChromiumDriver is the parent. However, ChromiumDriver is a child to the parent RemoteWebDriver.



In a nutshell, EdgeDriver extends ChromiumDriver. ChromeDriver extends ChromiumDriver but ChromiumDriver extends RemoteWebDriver.

```
public class EdgeDriver extends ChromiumDriver {

 public EdgeDriver() { this(new EdgeOptions()); }

 public EdgeDriver(EdgeOptions options) {
  this(new EdgeDriverService.Builder().build(), options);
 }
```

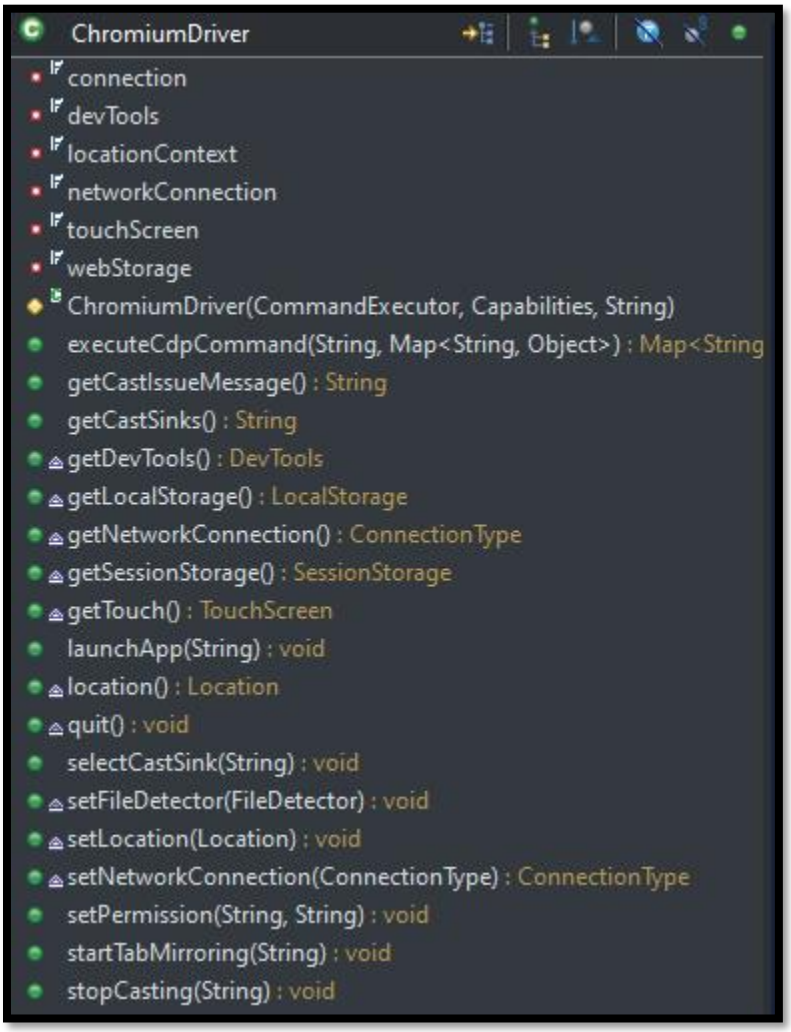
```
public class ChromeDriver extends ChromiumDriver {

 /**
  * Creates a new ChromeDriver using the {@link ChromeDri
  * server configuration.
  *
  * @see #ChromeDriver(ChromeDriverService, ChromeOptic
  */
```

```
public class ChromiumDriver extends RemoteWebDriver
    implements HasDevTools, HasTouchScreen, LocationContext, Networ

  private final RemoteLocationContext locationContext;
  private final RemoteWebStorage webStorage;
  private final TouchScreen touchScreen;
```

There's a lot of methods in the ChromiumDriver class but 2 methods allow us to control Developer Tools in Chrome and Microsoft Edge.

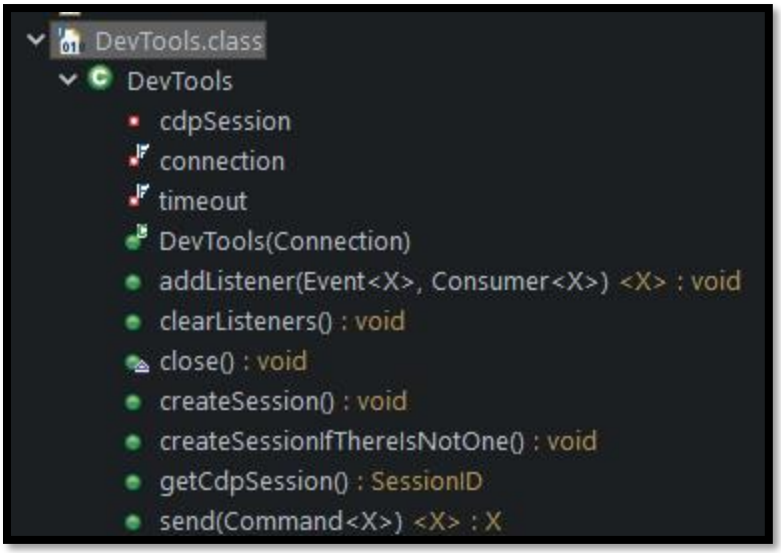


Those 2 methods are executeCdpCommand and getDevTools. The executeCdpCommand allows us to directly execute a Chrome DevTool Protocol command by passing in a parameter for that command. getDevTools is a method that returns DevTools.

```
public Map<String, Object> executeCdpCommand(String commandName, Map<String, Object> parameters) {
  Require.nonNull("Command name", commandName);
  Require.nonNull("Parameters", parameters);
```

```
@Override
public DevTools getDevTools() {
  return devTools.orElseThrow(() -> new WebDriverException("Unable to create DevTools connection"));
}
```

DevTools is a class that has methods to handle developer options. We see close, send, addListener. Know what, let me go back to the Project Explorer and look at the methods for DevTools. Here are the methods within DevTools: addListener, clearListener, close, createSession, createSessionIfThereIsNotOne, getCdpSession, and send.



We see a lot of methods for DevTools because we can do a lot of things with CDP. I'm going to show you how to view the Console Logs when it comes to CDP, Mock a GeoLocation, and Enable the Network Speed. Thanks for watching and I'll see you in the next session starting with View Console Logs. If you are interested in more videos, feel free to subscribe to my YouTube channel and click the bell icon. Also, follow me on Twitter, connect with me on LinkedIn and Facebook.

Contact

✔ YouTube https://www.youtube.com/c/RexJonesII/videos

- ✔ Facebook https://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos
- ✔ LinkedIn https://www.linkedin.com/in/rexjones34/