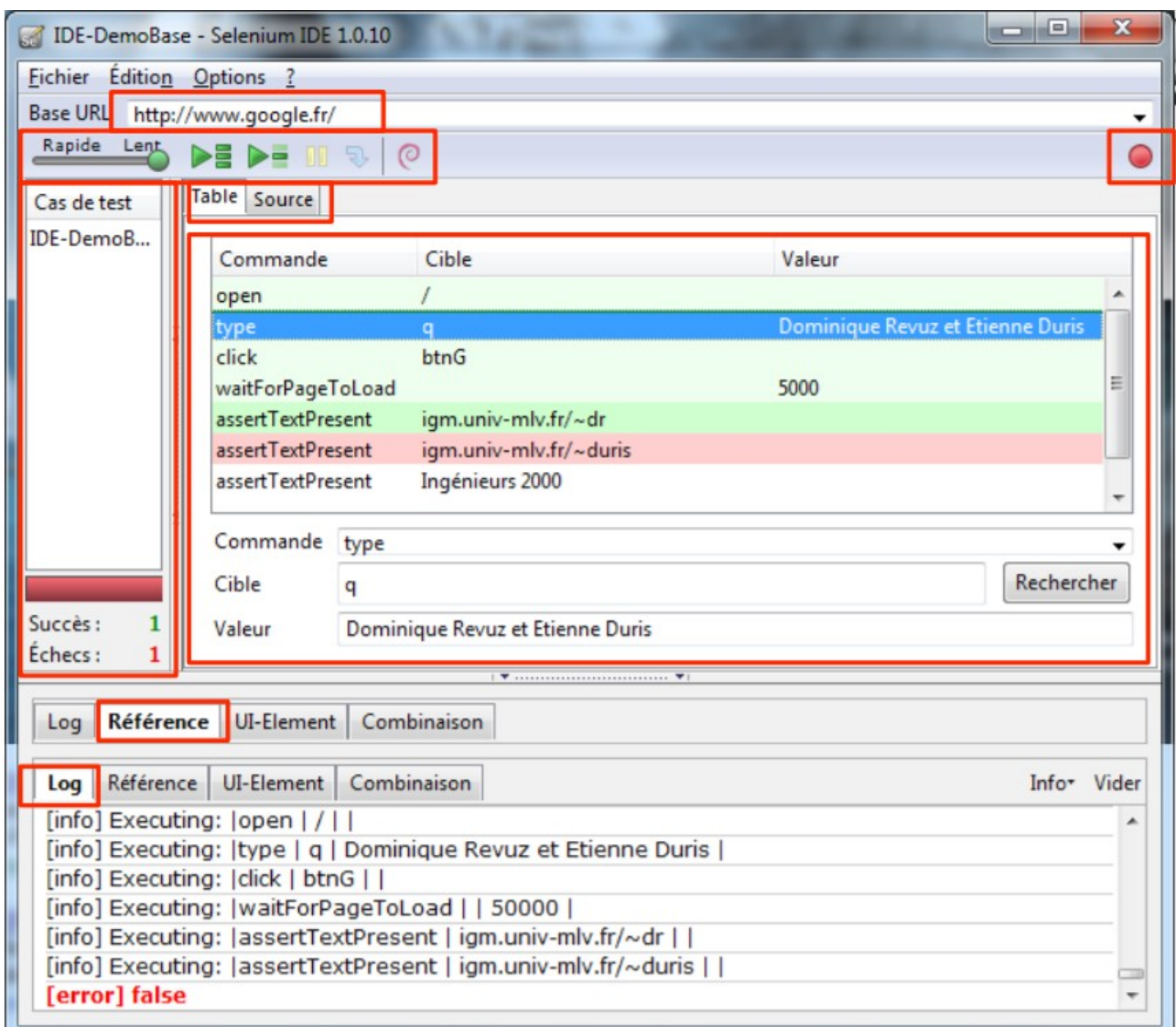
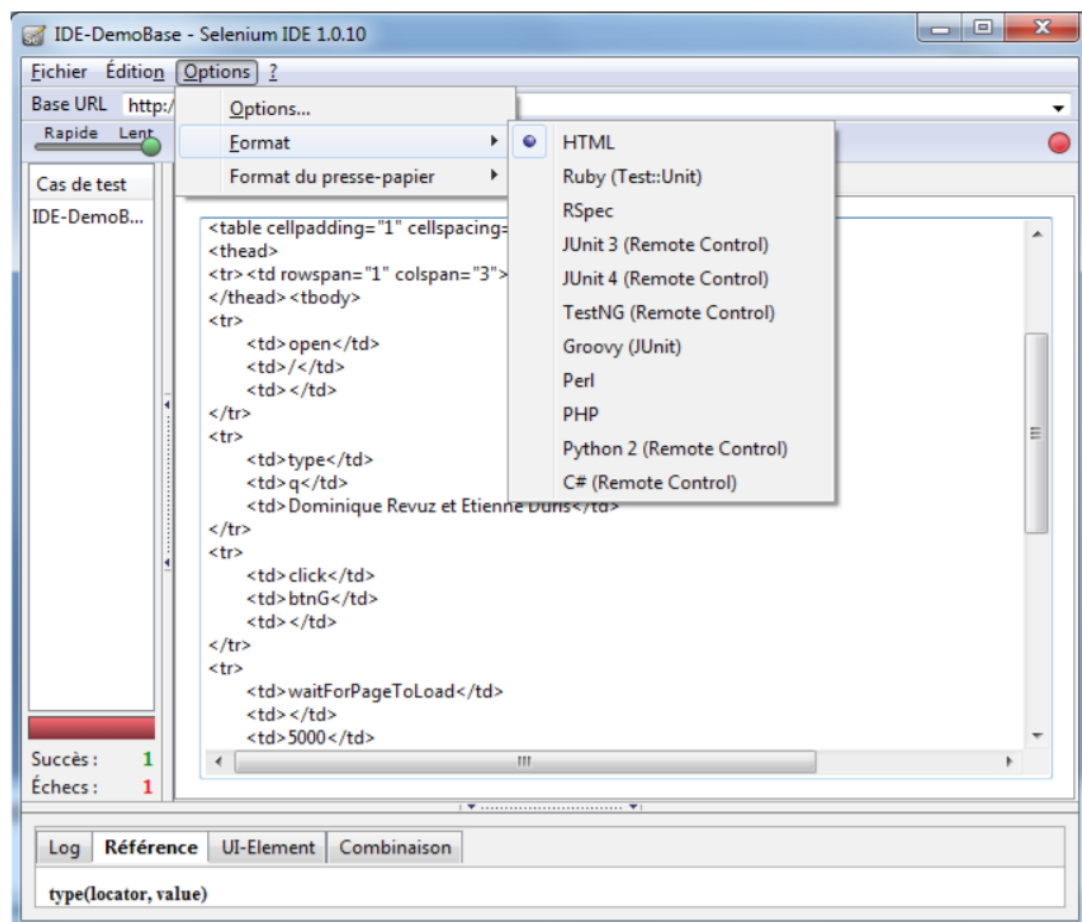
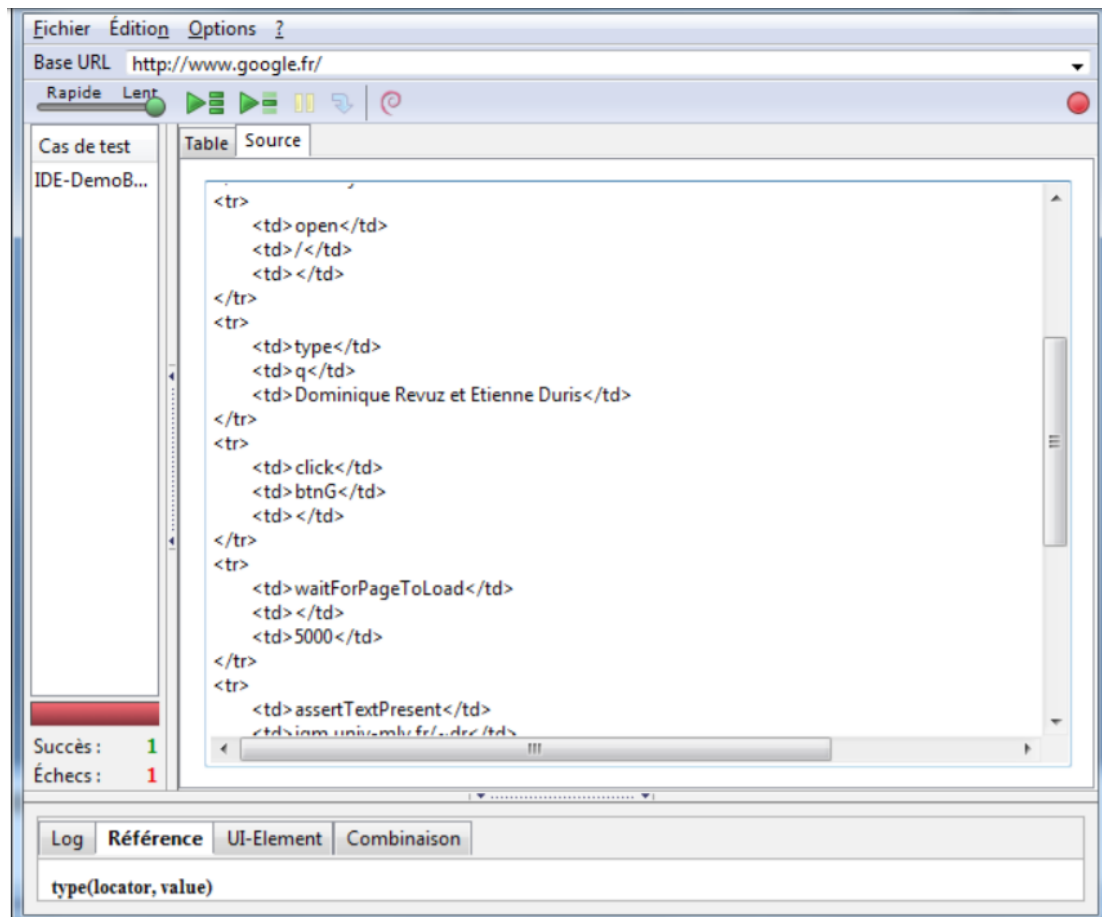


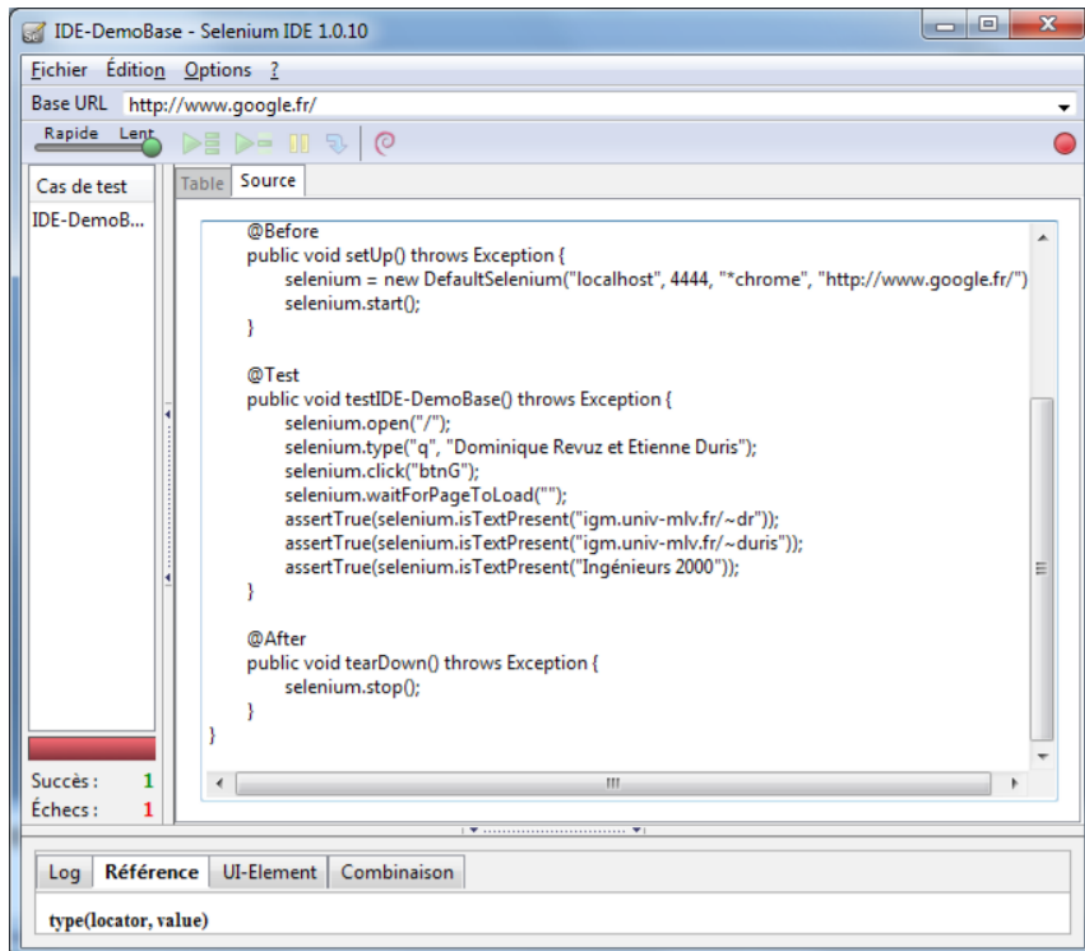
Documentation test IHM

Selenium:

- Extension Firefox
- Simuler l'interaction de l'utilisateur au travers de tests fonctionnels
- Permettre de faire des test IHM

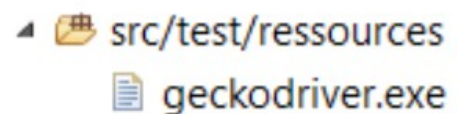
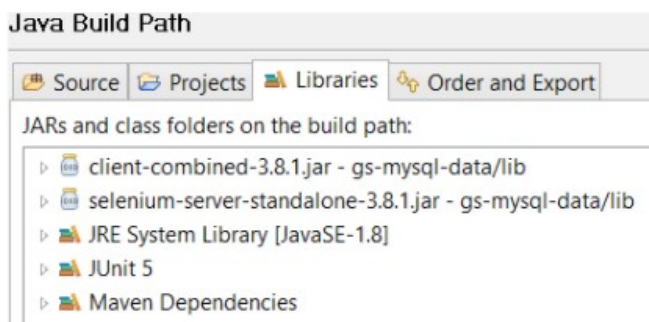






Integration dans Eclipse:

Il faut ajouter les jar et mettre le driver **geckodriver.exe** qui permet d'utiliser le web driver de Selenium



Quelques Indications sur le code sources:

Une fois que l'intégration est finie on commence à personnaliser le code afin de pouvoir effectuer les test IHM . Dans le code il faut indiquer le lien du geckodriver.exe et l'url de l'IHM à tester

Test IHM Login

```
public class test_ihm_login {  
  
    // @Ignore  
    @Test  
    public void test() {  
        write_file_test wr=new write_file_test();  
        String Form = null, footer = null, button = null, nav = null, titre=null;  
        System.setProperty("webdriver.gecko.driver", "src\\test\\ressources\\geckodriver.exe");  
  
        WebDriver driver=new FirefoxDriver();  
        driver.get("http://localhost:8080/login");  
    }  
}
```

Test IHM Inscription

```
public class test_ihm_Inscription {  
  
    // @Ignore  
    @Test  
    public void test() {  
        System.setProperty("webdriver.gecko.driver", "src\\test\\ressources\\geckodriver.exe");  
  
        WebDriver driver=new FirefoxDriver();  
        driver.get("http://localhost:8080/inscription");  
    }  
}
```

Exemple de vérification de l'existence d'un élément dans la page web :

```
try {  
    if(driver.findElement(By.className("form-control")) != null) {  
        Form="Formulaires Login/Mot de passe en position";  
    }  
} catch (Exception e) {  
    Form=" Formulaires introuvables";  
}
```

Une fois que toutes les vérifications sont faites on va générer les fichiers de sorties qui contiennent les résultats de ces vérifications et qui se trouve dans le dossier target.

