Expert Course

The goal of the expert course

To be able to:

1. Setup and use the EyeServer
2. Configure Remote Connection
3. Modify report templates
4. Use the EyeAutomate Java API
5. Extend EyeAutomate with custom commands
6. Use the Selenium integration

Content

1. EyeServer
2. Running a Remote Script
3. Report from a Remote Script
4. Exercise 1
5. Remote Connection
6. Exercise 2
7. Image analysis
8. Customizable reports
9. Logs
10. EyeAutomate Java API
11. Custom Java commands
12. Exercise 3
13. Selenium integration
14. Exercise 4

EyeServer

The EyeServer can be used for:

- Running remote scripts
- View and access remote desktop
- Synchronizing files
- Viewing reports in HTML format
- Study manual test sessions
- View test coverage and product quality

Start the EyeServer

The EyeServer is included in the SetupEyeServer installation package

Start by double-clicking the **“EyeServer.jar”** file

or using the command:

java -jar “EyeServer.jar”

Check if up and running:

http://localhost:1234/hello

Start the EyeServer

Start the EyeServer on another port by providing the -p parameter:

java -jar “EyeServer.jar” -p 8080

Check if up and running:

http://localhost:8080/hello

or by opening the dashboard:

http://localhost:8080

Dashboard


Dashboard
Refresh
Run Script
Progress
0%
Test Summary
Test Step Duration
Schedule
Off
No script selected
States
Issues
Resolved Issues
Sessions
3
Coverage
100%
Quality
High
Issues
Coverage
Email Notifications
Settings

Stopping the EyeServer

Stop the EyeServer using the command:

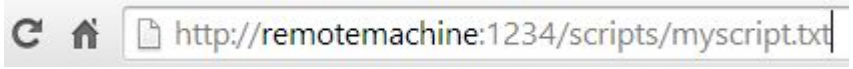
http://localhost:1234/kill

Running a Remote Script

Run a script on another machine using:

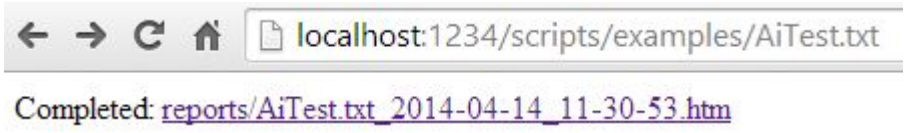
1. The **CallRemote** command from a script
2. A HTTP request in a browser or other software

```
// Call a script on another machine
CallRemote "http://remotemachine:1234/scripts/myscript.txt"
```



Report from a Remote Script

The EyeServer will respond with a Completed or Failed message to a HTTP request, for example:



Click the link to view the HTML report

Note that the reports are stored on the remote machine!

Web Service Reports

Reports can be viewed and generated from the EyeServer without using EyeAutomate Studio

Generate a Test Summary report using:

http://[server address]:[port]/TestSummary

Generate a Test Step Duration report using:

http://[server address]:[port]/TestStepDuration

Generate a Test Steps report using:

http://[server address]:[port]/TestSteps?script=scripts/myscript.txt

Exercise 1

Start the Server and Run a Script

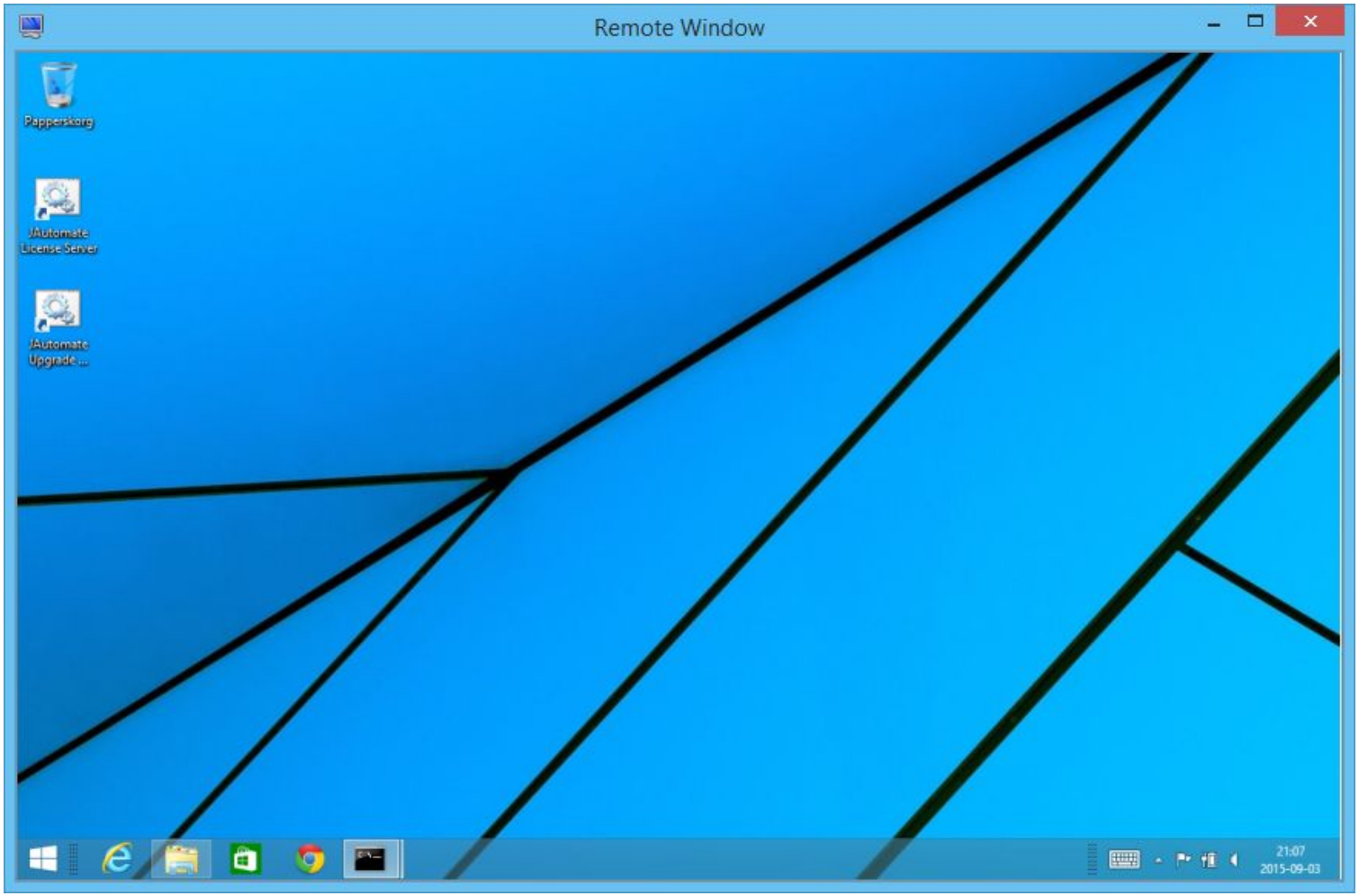
1. Download and install EyeServer
2. Start EyeServer by launching the “EyeServer.jar”
3. Open the Dashboard: http://localhost:1234
4. Run any of your scripts
5. View the report
6. Stop the service using the kill command

Remote Connection

Three options in the File menu in EyeAutomate Studio:

- Remote Dashboard
- Remote Window
- Synchronize

Remote Window



File synchronization

EyeAutomate Studio has built-in file synchronization:

- Enable and connect using the **Settings / File Synchronization** menu option
- Make some changes
- Synchronize by selecting **File / Synchronize**
- Performed automatically before a run when the Remote Window is visible

Filter settings

The “filter.properties” file contains the files to synchronize

Standard filter ("filter.properties"):

images/.*\.png
scripts/.*\.txt
data/.*\.csv
widgets/.*\.wid

Change repository

The “change_repository.csv” file in the “logs” folder contains all changes received from the clients when synchronizing files

Note that removing this file will reset all changes with the result that no clients will receive any updates!

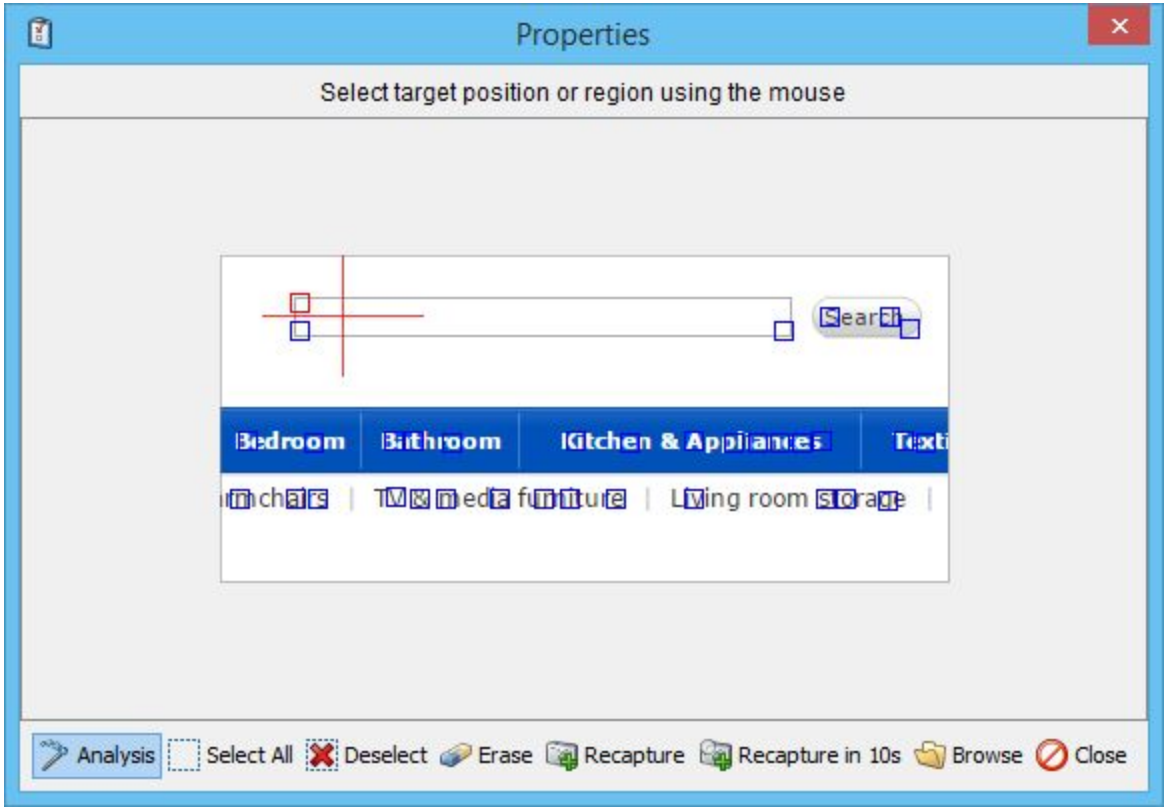
Exercise 2
Remote Connection

Follow the steps below:

1. Make sure that the EyeServer is running on some machine
2. Connect using the Settings / Remote Connection menu option. Connect to http://localhost:1234 or another machine
3. Open the Remote Window from the File menu
4. Create a simple script
5. Run the script on the remote machine

Image analysis

Reveal more information about how EyeAutomate analyses an image by pressing the Analysis button in the Properties dialog



Customizable reports

All report types have their own template (report_templates folder)

Example of things that can be customized:

- Report title
- Column titles
- Layout
- Fonts
- Colors
- Date and time format

Report template

```
<title>Test Summary</title>
<script type="text/javascript">
function toLocalTime(time)
{
     return time;
}
</script>
<style type="text/css">
td.green
{
     padding: 10px;
     background-color: #00ff00;
}
</style>
</head>
<body>
     <h2>Test Summary</h2>
        <table>
            <tr>
                <th>Script</th>
                <th>Runs</th>
                <th>Passed</th>
                <th>Failed</th>
                <th>Last Status</th>
            </tr>
               ID_REPORT_TABLE_ROWS
        </table>
</body>
```

Logs

- The result from the test runs are stored in the "test_history.csv" and "test_steps.csv" files in the “logs” folder
- The log can be cleared using the **Clear Statistics** option in the **Reports** menu. A backup is created before the log files are cleared

Script	Date	Time	Passed	Failed
scripts/siba	buy.txt	2012-10-10	15:30:20	8	0
scripts/siba	buy.txt	2012-10-10	15:30:37	8	0
scripts/siba	remove.txt	2012-10-10	15:30:47	3	0
scripts/siba	remove.txt	2012-10-10	15:30:52	3	0

EyeAutomate Java API

The Java API can be used from any Java program to run scripts or execute commands

- The Java project must include the “EyeAutomate.jar” file and import the “jautomate.ScriptRunner”
- The “EyeAutomate.jar” file does not have dependencies to any other non-standard libraries
- API documentation can be found in the “javadoc” folder

Custom Java commands

Extend the functionality in EyeAutomate using custom Java classes

- Custom classes are stored in the “custom” folder and should belong to the “custom” Java package
- Custom commands appears in the **Commands/Custom** menu

Example:

Replace “A text to replace” “text” “test”

Replaced = “A test to replace”

Custom command example 1

```
package custom;
import java.util.Properties;
public class Replace {
  public Boolean executeCommand(String[] commandParameters, Properties scriptParameters) {
    if(commandParameters.length<3) {
      scriptParameters.put("Error", "Missing parameter. Usage: Replace Text From To");
      return false;
    }
    String text=commandParameters[0];
    String from=commandParameters[1];
    String to=commandParameters[2];
    String replaced=text.replace(from, to);
    scriptParameters.put("Replaced", replaced);
    return true;
  }
}
```

Custom command example 2

```
public class ClickFirstVisible {
  public Boolean executeCommand(String[] commandParameters, Properties scriptParameters) {
    ScriptRunner scriptRunner=new ScriptRunner();
    for(String commandParameter:commandParameters) {
      BufferedImage image=scriptRunner.loadImage(commandParameter);
      if(image!=null) {
        if(scriptRunner.mouseMove(image)) {
          scriptRunner.mouseLeftClick();
          return true;
        }
      }
      else {
        scriptParameters.put("Error", "Failed to load image");
        return false;
      }
    }
    return false;
  }
}
```

Custom Methods

There are a few methods that can, optionally, be defined:

- getHelp() - Returns an URL to help documentation
- getParameters() - Returns a String array to parameter names used
- getCommand() - Returns an initial command String

Example:

```
public String getHelp() {
    return "http://jautomate.com/2014/03/03/add";
}
public String[] getParameters() {
    return new String[] {"Error", "Response"};
}
```

Exercise 3
Create a Custom Command

Follow the steps below to create a custom command using Java:

1. Download and install Eclipse Standard from: https://www.eclipse.org/downloads
2. Create a new project
3. Add a package named “custom”
4. Add a class to the package with the name of your command
5. Add the method:
   public Boolean executeCommand(String[] commandParameters, Properties scriptParameters)
6. Create your command and place the result in the scriptParameter Properties
7. Place the .class file in the “custom” folder of EyeAutomate
8. Restart EyeAutomate and try out your command

Selenium

Selenium automates browsers. That's it!

Two versions:

- Selenium IDE
- Selenium WebDriver

Selenium integration

- Possible to create cross functional scripts that are a mix of image recognition and structural techniques
- Can take advantage of the strengths and avoid the weaknesses of both technologies
- EyeAutomate can be used for navigating and verifying images while Selenium can be used for verifying text or selecting options from drop-down lists

Selenium integration

// Start Firefox from Selenium

OpenBrowser firefox

// Open Wikipedia using Selenium

GetUrl http://www.wikipedia.org

// Select the search input field

SelectId searchInput

// Write a text

WriteText "Tiger"

// Click the search button

Click 

// Verify the result

WaitVerify  34 25 30 26

// Close the browser using Selenium

CloseBrowser

Selenium Commands

Open the Firefox browser:

OpenBrowser firefox

Open the website (in this case www.wikipedia.org):

GetUrl http://www.wikipedia.org

Get the text or value from the text field, drop-down or list in focus:

GetValue

Get the text from the drop-down or list in focus:

GetSelectedText

Write a text to the text field in focus:

WriteText tiger

Selenium Commands

Select an item in the drop-down or list in focus using text, value or index (0=first):

SelectText English
SelectValue en
SelectIndex 2

Click on a widget:

ClickId searchInput
ClickText English
ClickName go
ClickXpath //option[@id='cat'

Close the web browser:

CloseBrowser

Custom Selenium Commands

A Selenium command may get and set the currently used web driver using the methods:

```
public void setWebDriver(WebDriver webDriver) {
   this.webDriver=webDriver;
}

public WebDriver getWebDriver() {
   return null;
}
```

Exercise 4
A Hybrid Selenium and EyeAutomate Script

Follow the steps below:

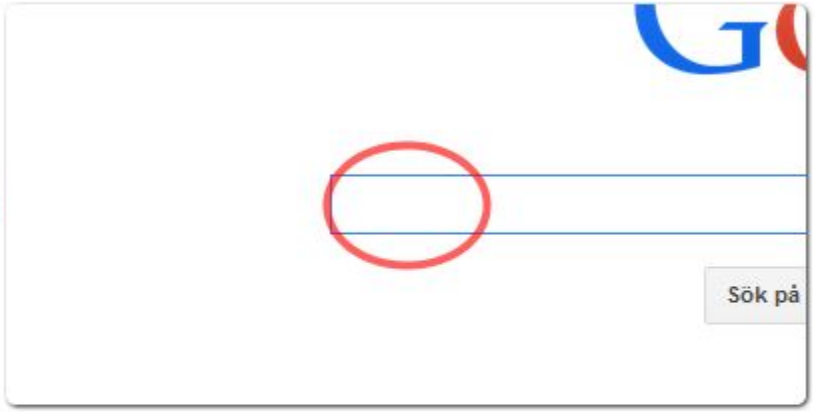
1. Create a script that searches for a page using Google containing both EyeAutomate and Selenium commands

Exercise 4
Solution Example

OpenBrowser firefox

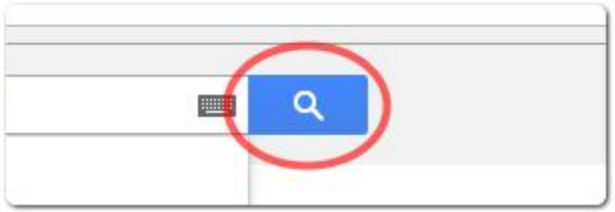
GetUrl http://www.google.com

Click



WriteText Hello

Click



Expert Course