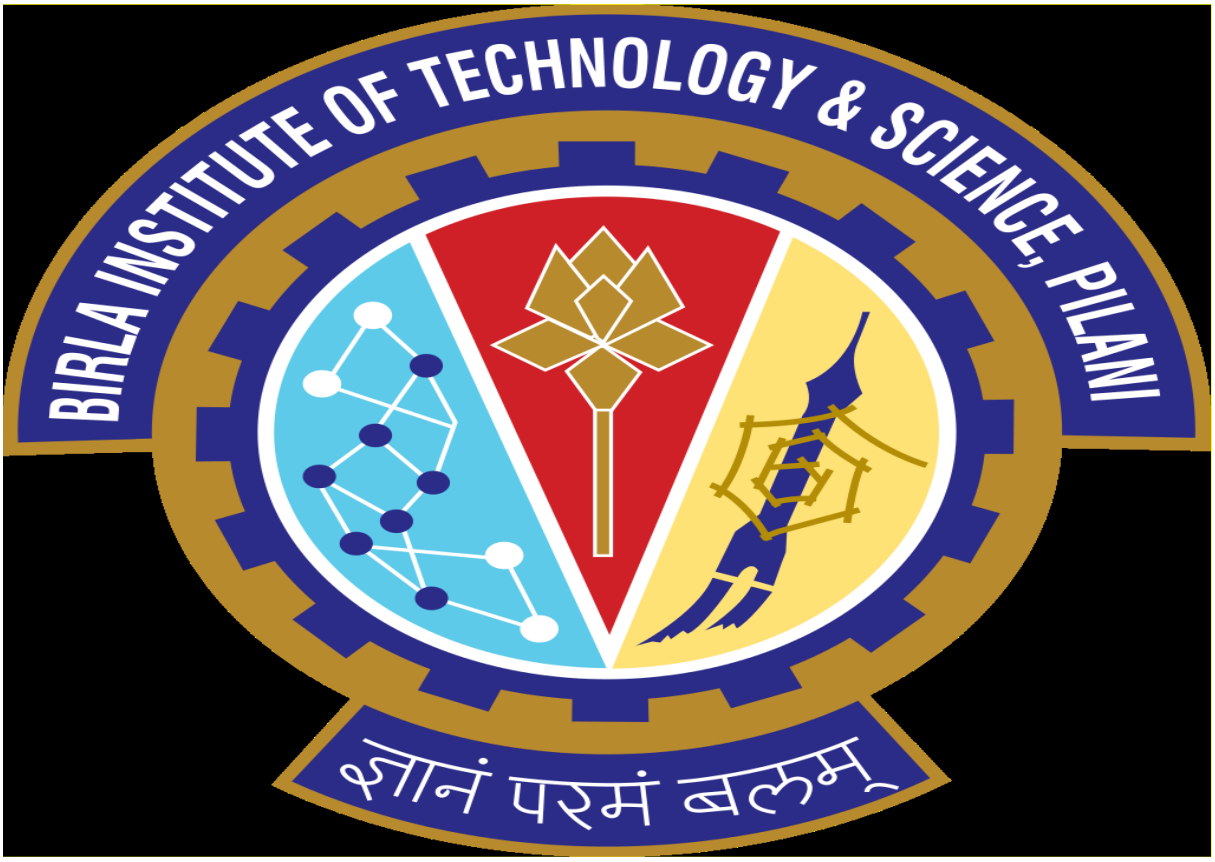
Mid-Semester Report

on

Co-browse Performance Test Using Selenium



Mid-Semester Report

on

Co-browse Performance Test Using Selenium

Written By

Pranjal Deshmukh 2013A3PS276P

In Partial Fulfilment of the
Practice School - II Course

At

AVAYA India Pvt.Ltd,Pune

A Practice School - I Station of
Birla Institute of Technology and Science, Pilani
September, 2016

Birla Institute of Technology and Science Pilani (Rajasthan)

Station: AVAYA

Centre: Pune

Duration: 5 months

Date of start: 18-07-2016

Date of submission: 23-09-2016

Title of the project: Co-browse Performance Test Using Selenium

Name and ID no. of student: Pranjal Deshmukh (2013A3PS276P)

Name and designation of the expert: Mr.Arjun Rawat(Project Manager)

Name of the PS faculty: A. Vijayalakshmi

Keywords: Cobrowse,Performance testing, PhantomJS

Project areas: JAVA, Selenium Webdriver and Co-browsing

Abstract: This report probes into the area of performace testing of Co-browse which is a part of Avaya's upcoming project Oceana™ 3.2. Co-browsing is a software enabled technique helping a customer by connecting it to an agent. Avaya Co-Browsing Snap-in provides a set of consolidated services for sharing a webpage session. Using Avaya Co-Browsing Snap-in, two users can browse the same webpages simultaneously to collaborate and accomplish certain tasks. The agent can assist the customer to navigate through the webpages and, if required, in filling out forms.

Acknowledgements

I am highly indebted to The Management of Avaya,Pune for their guidance and constant supervision as well as for providing necessary information regarding my project .

I wish to express my deep sense of gratitude to Mr. Arjun Rawat.(Manager, CCSV Team) and for his able guidance and useful suggestions throughout my project.

Also I would like to acknowledge the help and support given by our PS Instructor, A. Vijayalakshmi ma'am.

Words are inadequate in offering my thanks to Mr. Rohit Galchar & Ms.Anujaa Shenvi for their encouragement and cooperation in carrying out the project work.

I would like to acknowledge BITS Pilani, to have given us the opportunity of Practice School-II.

Contents

INTRODUCTION:

About Avaya:

Overview & History:

Avaya's history can be traced back over a century to the acquisition of Western Electric by American Bell Telephone Company in 1881, the predecessor to AT&T. In 1984, after the US government forced the breakup of AT&T, the equipment business was renamed AT&T Technologies, and in 1996, spun off again as Lucent Technologies. In 2000, Lucent spun off to its shareholders its Business Communications unit, naming the new company Avaya. On October 1, 2000, Avaya began trading on the New York Stock Exchange as an independent, public company. In 2007, Avaya was taken private by two private equity companies, TPG Capital and Silver Lake Partners, in a $8.3 billion leveraged buyout (LBO) at the height of an economic boom fueled by a bubble in residential housing. In 2009, Avaya purchased Nortel's Enterprise Solutions business (NES) for nearly $1 billion, bolting on approximately $2 billion in incremental revenue and giving it access to Nortel's data hardware business, patents, and a trove of new technology.

Today Avaya, with over $5 billion in worldwide revenue is a behemoth in the business telecommunications world. It is the leading global provider of next-generation business collaboration and communication solutions according to an advertisement placed at the beginning of a recent filing with the SEC. It claims to do business with 85% of the Fortune 500 companies and has operations around the globe with over 400,000 customers at 1 million locations. Along with its direct sales force, Avaya has a large and sophisticated global supply chain that is run by 10,200 indirect channel partners allowing it to deliver technology solutions anywhere there's a business need.

Avaya Inc is a privately held global provider of business communications and collaboration systems. The company supplies contact centers, networking (routers, switches and other networking hardware), unified communications (UC), and video products (integrated hardware and software) services

Avaya focuses on these major businesses:

Unified Communications: Avaya's unified communications solutions help companies increase employee productivity, improve customer service and reduce costs by integrating multiple forms of communications, including telephony, e-mail, instant messaging and video all across multivendor networks. With Avaya unified communications, customers can communicate effectively regardless of location or device. Avaya's open, standards-based UC software and hardware are widely recognized as the most reliable, secure and comprehensive offerings in the industry.

Contact Centers :Avaya is the global leader in the contact center market. Avaya offers highly reliable, scalable communications solutions that improve customer service and help companies compete more effectively. Avaya's contact center solutions include intelligent routing, selfservice and proactive contact applications that drive effective communications and transactions with customers. In addition, Avaya's analytics and reporting platform, Avaya IQ, provides companies with detailed customer information that improves profitability and customer retention. :

Avaya Global Services : evaluates, designs, implements and manages enterprise communications networks for superior business results. Avaya's consulting and implementation services are backed by approximately 8,700 employees worldwide; 32 global delivery support centers; and unique, patented design and management tools.

Small & Medium Enterprise Communications :Avaya's Small and Medium Enterprise Communications unit is focused on enterprises with up to 250 employees. The company offers complete solutions that bring together telephony, messaging, networking, conferencing, and customer management designed for the requirements of small and medium enterprises. The products and services are sold primarily through Avaya's global channel partners.

Data Networking: Avaya's Data Networking portfolio offers the reliable, secure, endto-end solutions needed to fully use real-time communications, delivering world-class performance and superior return on investment. Avaya's diverse data portfolio is committed to delivering innovative technology, energy-efficient hardware, and improving the effectiveness of business by creating a simpler and more effective network. Organizations should never be constrained by their

networks, and Avaya's proven data solutions ensure the network is accepted and valued as a genuine real-time communications enabler.

About the Project:

My project comprisises of performance testing of Co-browse.I was asked to write the scripts for the performance testing of co-browse. Avaya Co-Browsing Snap-in provides a set of consolidated services for sharing a webpage session. Using Avaya Co-Browsing Snap-in, two users can browse the same webpages simultaneously to collaborate and accomplish certain tasks. The agent can assist the customer to navigate through the webpages and, if required, in filling out forms. Avaya Co-Browsing Snap-in leverages the Document Object Model (DOM), which is an application programming interface (API) for valid HTML documents.

Avaya Co-Browsing Snap-in provides the following functionality:

• A standard REST Web Service API to provide access to the Avaya Co-Browsing Snap-in services

• A developer SDK, including a sample reference client, that provides co-browsing capabilities.

• Out-of-the box summary reports about agents, sessions, and customers.

This co-browse application provides full-fledged collaboration between customer and agent. The application uses DOM synchronization to achieve this purpose. Customer is the person who needs help and agent is the person who is helping.The task given to me was to erite scripts in JAVA & selenium WebDriver, such that they will execute the following stepts:

1. Open agent's webpage.

2.Type in the name &then click on Create Session button(Agent),

3.Agent joins the session & session key is displayed on the webpage,

4.Open customer's webpage,

5. Customer joins the session by entering his name & session key(which is generated when agent joins the session),

6.Now customer clicks on Open Account button & starts filling the form.

7. Agent verifies whether the details filled by the customer are correct or not.

The next task given to me was to write scripts such that they will execute the following steps:

1. Open Customer's webpage.

2.Customer clicks on the request button in order to request for a co-browse session.

3.Customer enters his/her nameand then clicks on the request button.

4.When a customer requests for a session,a session key will be generated.That session key will be copied by the agent afterwards in order to join the co-browse session.

4.Open agent's webpage.

5. Agent joins the session by entering his name & session key(which is generated when Customer requests for a session).

6.Now customer clicks on Open Account button & starts filling the form.

7. Agent verifies whether the details filled by the customer are correct or not.

Project Details:

Technologies Used:

1.Selenium Webdriver:

Selenium Web driver is a web automation tool which enables you to run your tests against different browsers. These browsers can be Internet Explorer, Firefox or Chrome. To use a particular browser with Selenium you need corresponding driver.

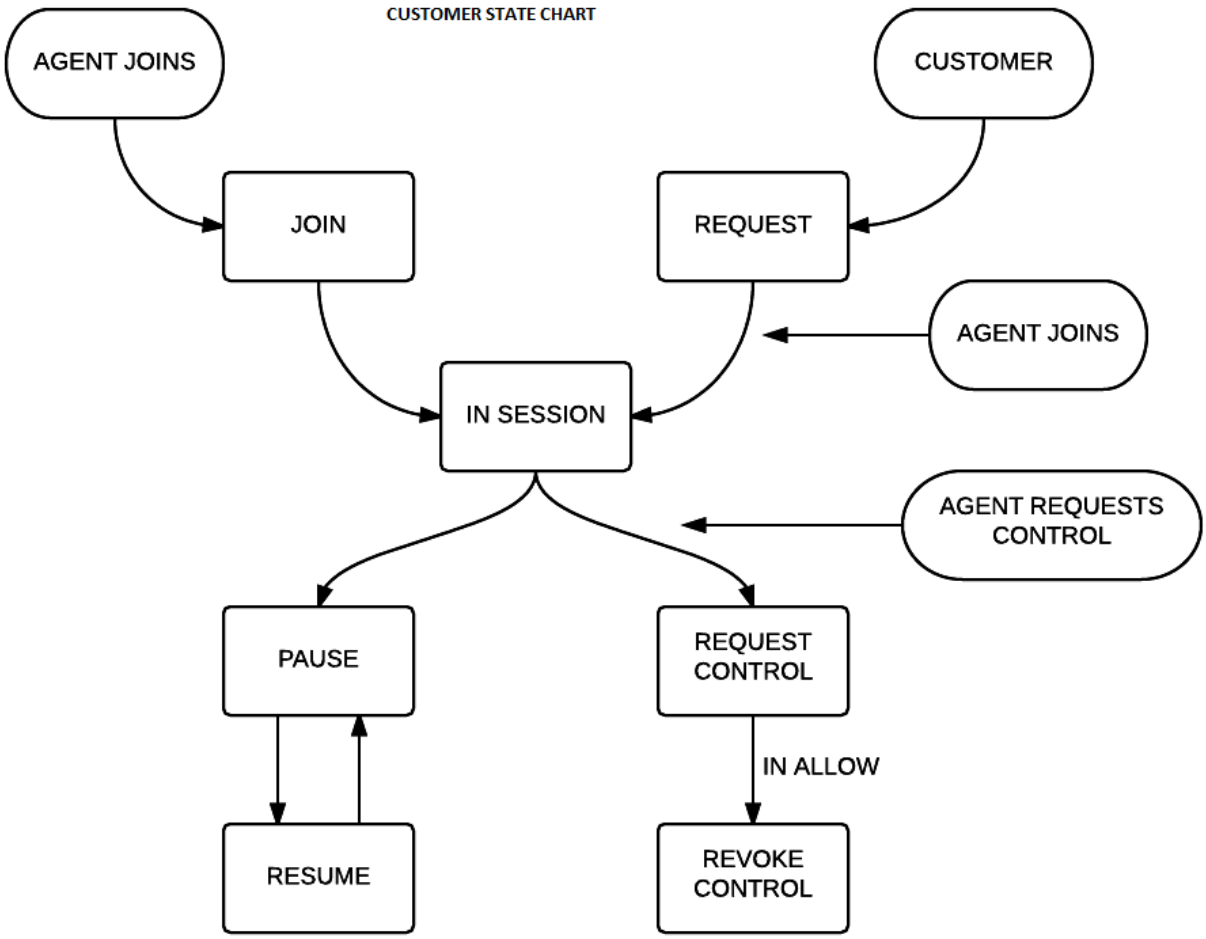
At test run, Selenium launches the corresponding browser called in script and executes test steps. You can see the browser and the test execution in action.

2.PhantomJS

PhantomJS is a headless browser with JavaScript API. It is an optimal solution for Headless Website Testing, access and manipulate webpages & comes with the standard DOM API.

In order to use PhantomJS with Seleniun, one has to use GhostDriver. **GhostDriver** is a implementation of Webdriver Wire protocol in simple JS for PhantomJS.

The latest release of PhatomJS has **integrated** GhostDriver and **there is no need to separately install it.**



As shown above,either agent or a customer can initiate a co-browse session.

Agent-initiated co-browsing:

An agent can start a co-browsing session to assist the customer. The agent must share the system generated session key with the customer so that the customer can join the session. The session key is an eight-digit number and must have a space between four digits. For example, “1234 5678”. The user name you specify cannot be more than 30 characters long and has to be separated with a space after the first 8 characters. For example, “Abcdefgh ijklmn”.

Two-way co-browsing:

The customer can give control of the co-browsed webpage to an agent for assistance. By default, an agent has view-only permission to a co-browsed webpage. The agent can request permission to control or the customer can promote the agent to control the webpage. During the period in which the agent has control, the agent gains restricted access to the webpage. The customer can revoke control from the agent at any point of time or the agent can voluntarily release control.

Data masking:

The customer can ensure data privacy and secure co-browsing by using security measures such as hiding sensitive information, preventing certain actions, and hiding certain elements. Depending on the legal and location-specific requirements, the customer can apply data masking to certain fields such as Social Security Number or credit card number. The customer can also block certain actions so that the agent does not submit any information on behalf of the customer.

.

Agent initiated session :

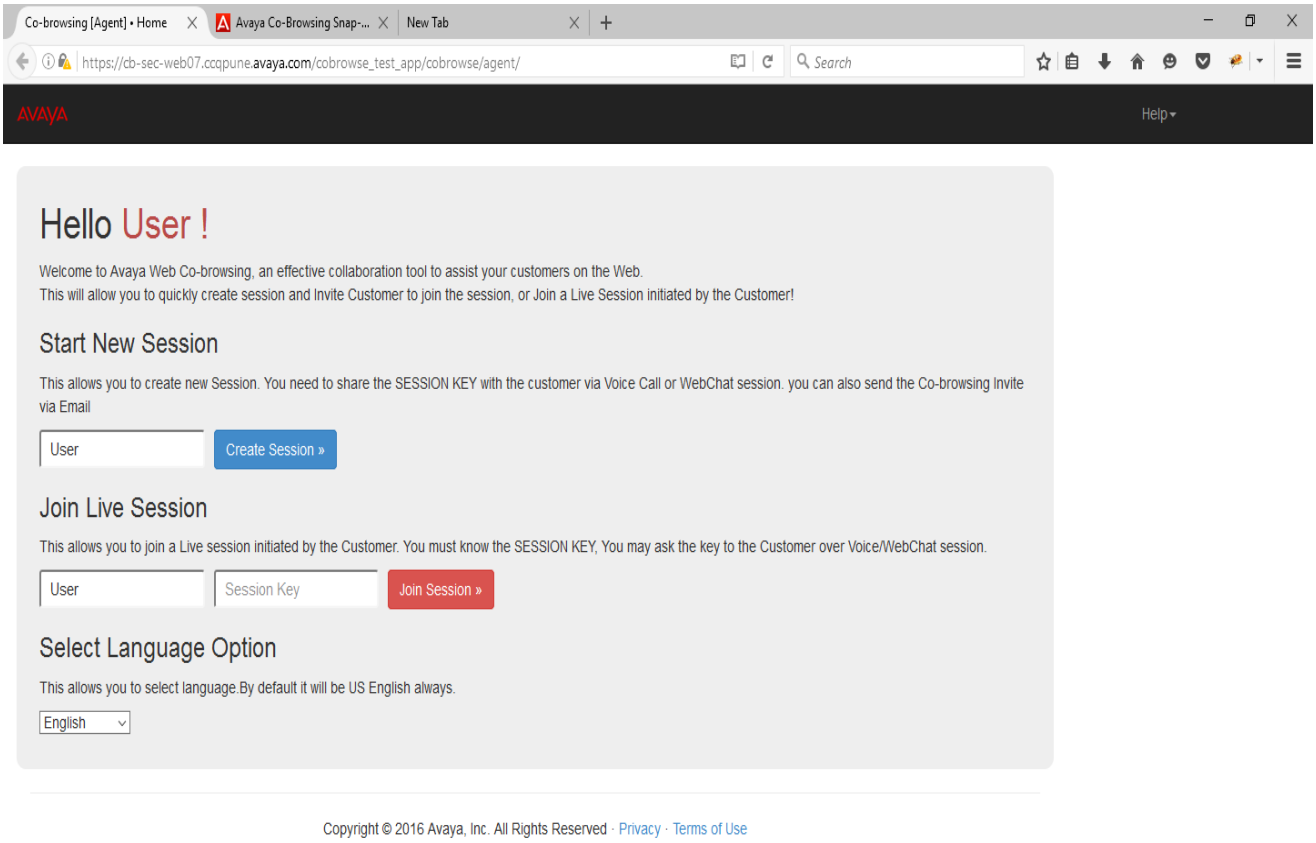
An agent can initiate a session or join a live session. For an agent initiated session, the system generates a session key and displays the session key on screen. The customer must use the key to join the session. An agent can initiate a session and cancel the session even after the session key is generated. The system cancels the session key and the agent can start a new session. An agent can request control from the customer. When an agent requests control, the customer can allow or deny sharing the control. The agent can logout from the session. If the agent, that is, the session owner is idle for some time, then the system automatically closes the session. You can configure the inactivity time out. The default value is 2 minutes.

Customer initiated session

A customer can initiate a session or join a live session. For customer initiated session, the system generates a session key and displays the session key on screen. The agent must use the key to join the session. The customer can pause and resume the current session. If the customer is in control of the session, the customer can pause a session. Only when the customer resumes the session, the changes made are synchronized and visible to the agent. The customer can stop the session. If the customer, that is, the session owner is idle for some time, then the system automatically closes the session. You can configure the inactivity time out. The default value is 10 minutes. While the agent is controlling the session, the customer can revoke the access at any point of time.

Screenshots of the Co-browse session:

Agent's Webpage:

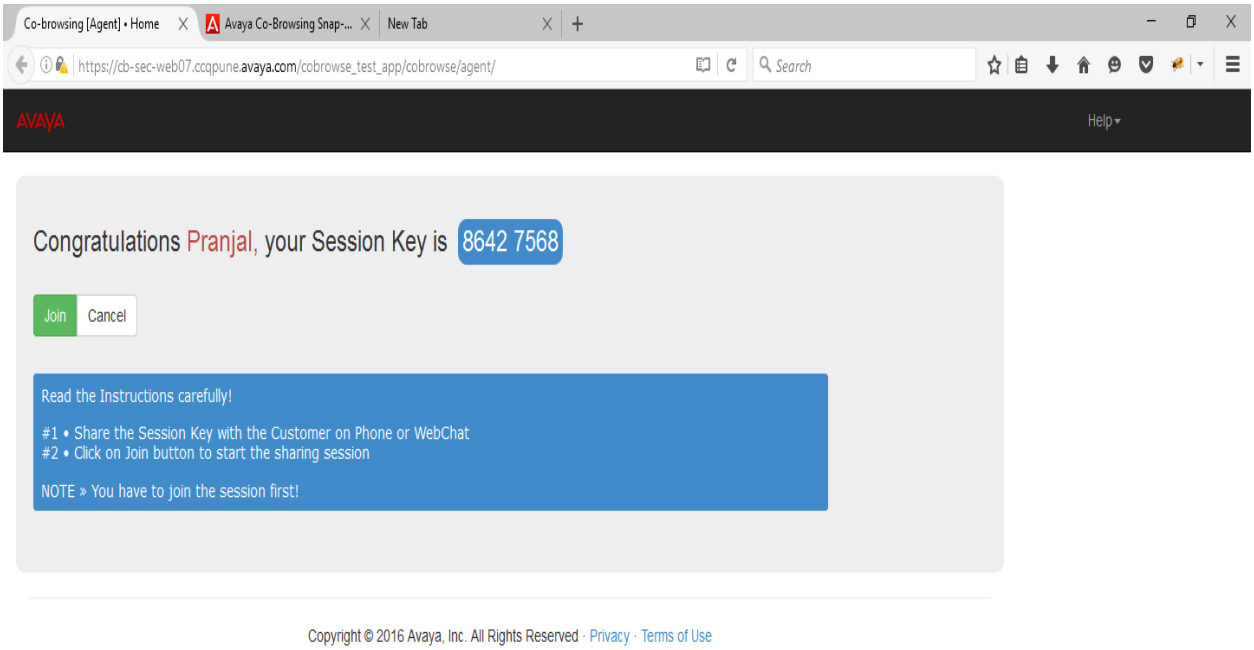


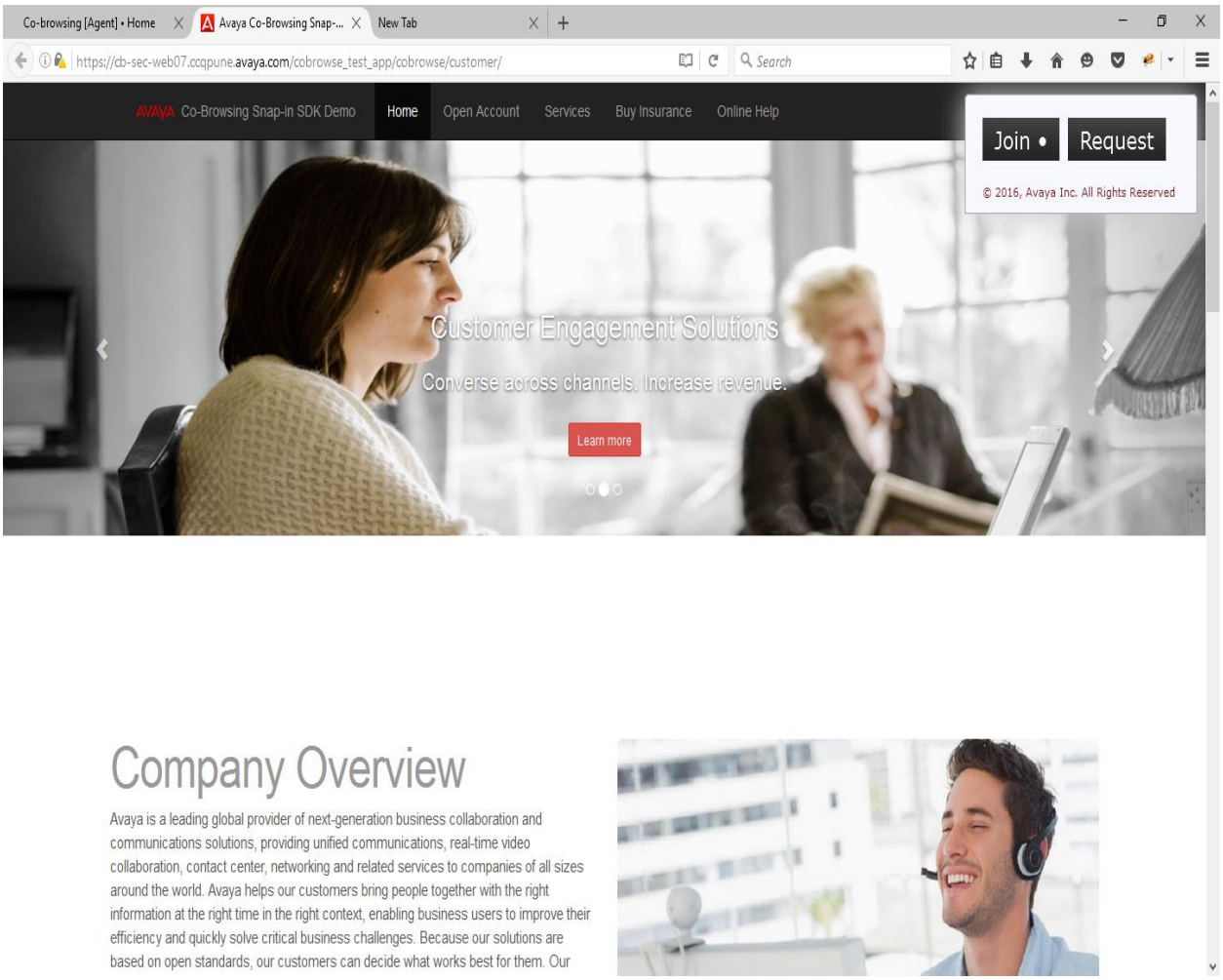
Agent Enters username & clicks on the create session button:

After completion of the above steps, a session key will be generated(8 digit session key)as shown below:

:

Now the agent should click on the join button in order to join co-browse session

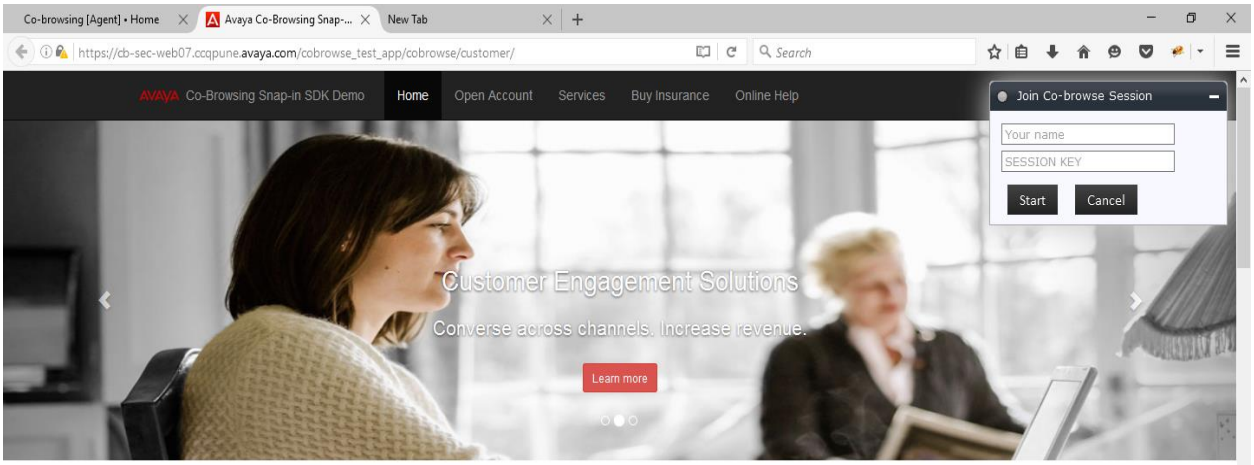


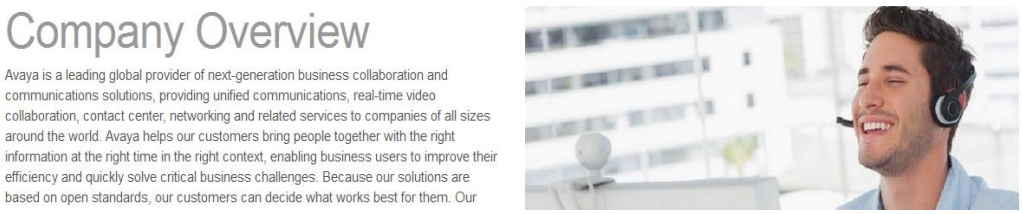


Customer's Webpage:

After opening the customer's webpage,customer will click on the join button in order to join the co-browse session initiated by the agent.

After Customer clicks on the join button:



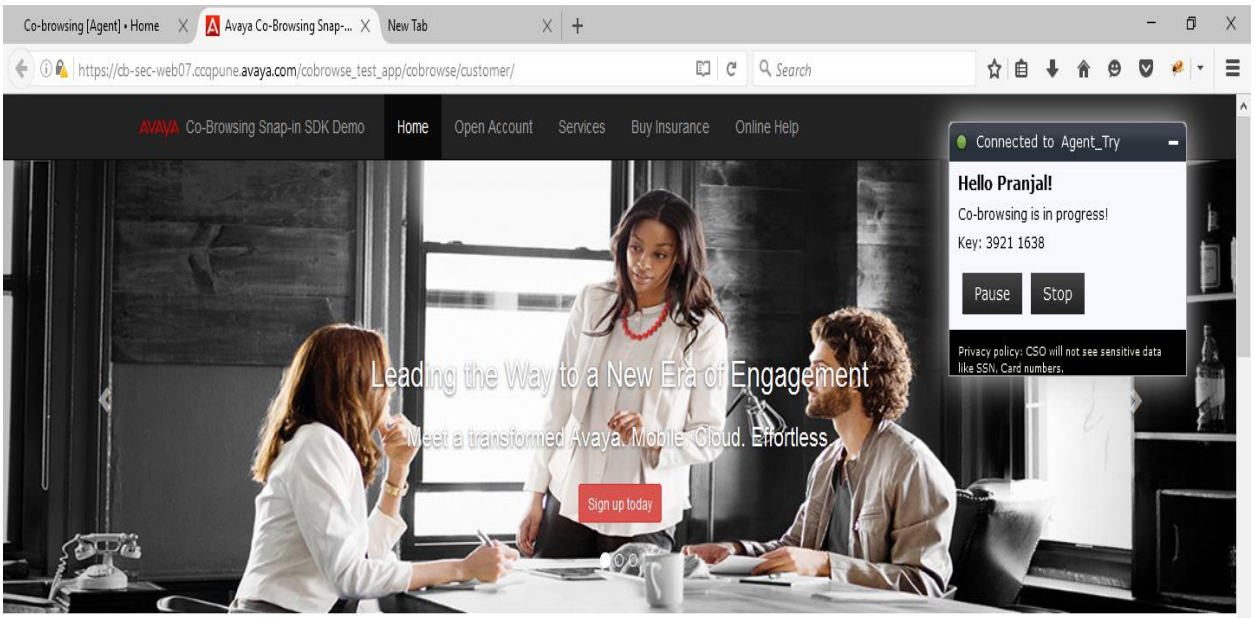


Customer fills the details(His name & session key):

When agent joins the session,a session key is generated.That session key will be copied by agent and then the customer when fills that session key in the field,he will be able to join the session with that particular agent.

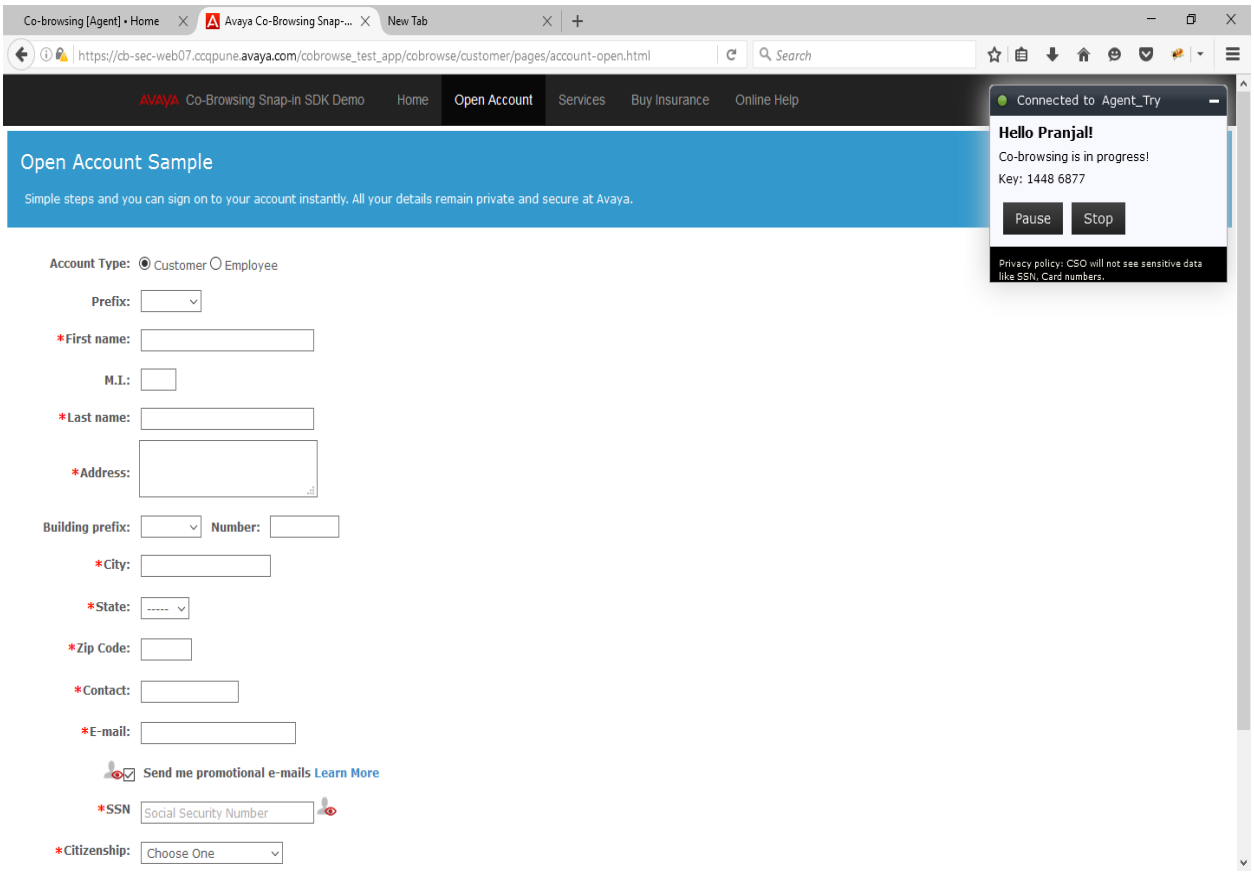
After entering customer's name & session key,customer will click on the start button to start the CB session.

As can be seen from the screenshot given below,The customer is now connected to Agent_Try.Agent is now able to share the same screen as customer.Basically agent can now see whatever the customer is doing.



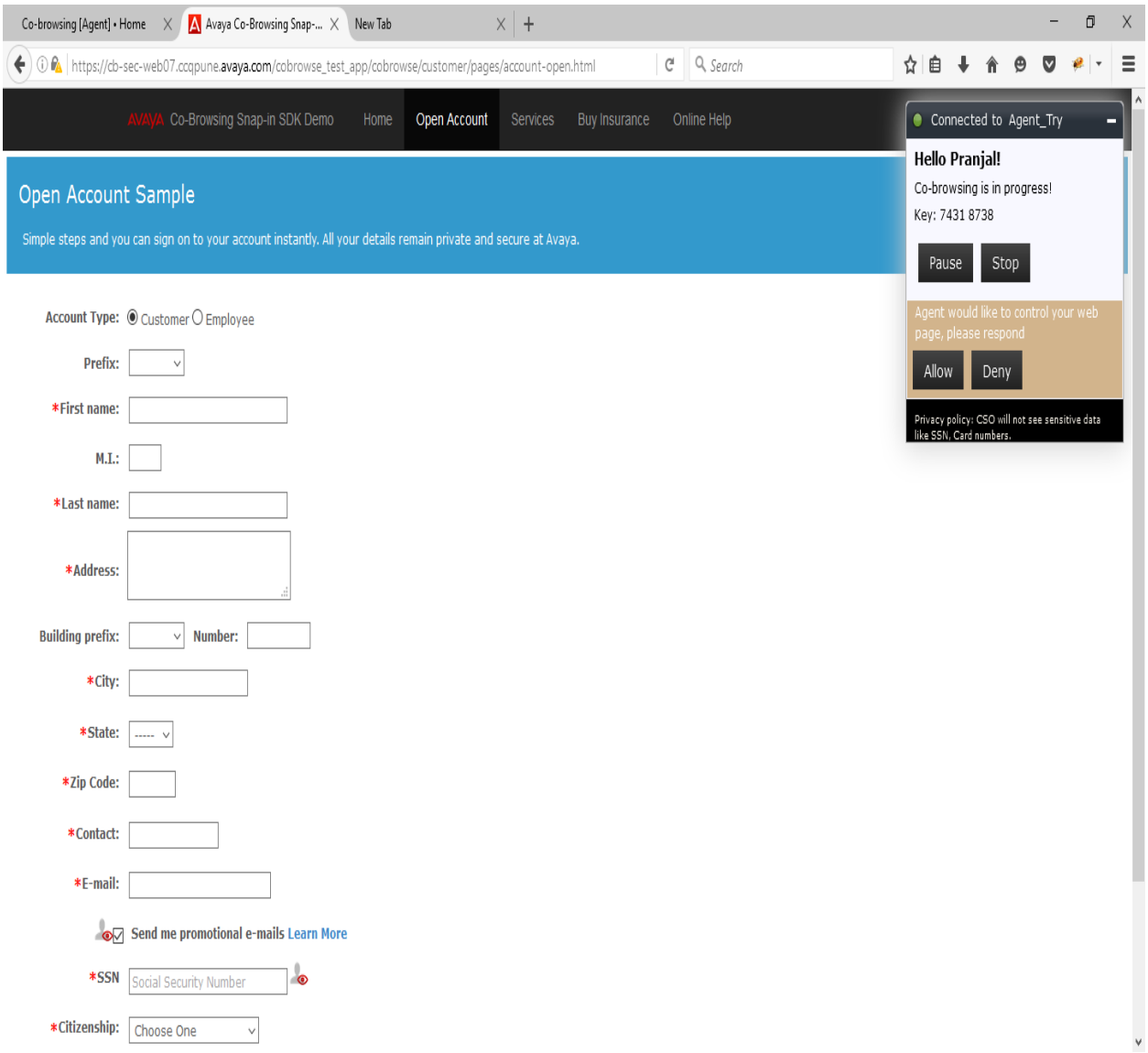
Customer clicks on the Open Account button:

After the customer clicks on the open account button,a form is displayed as shown below.



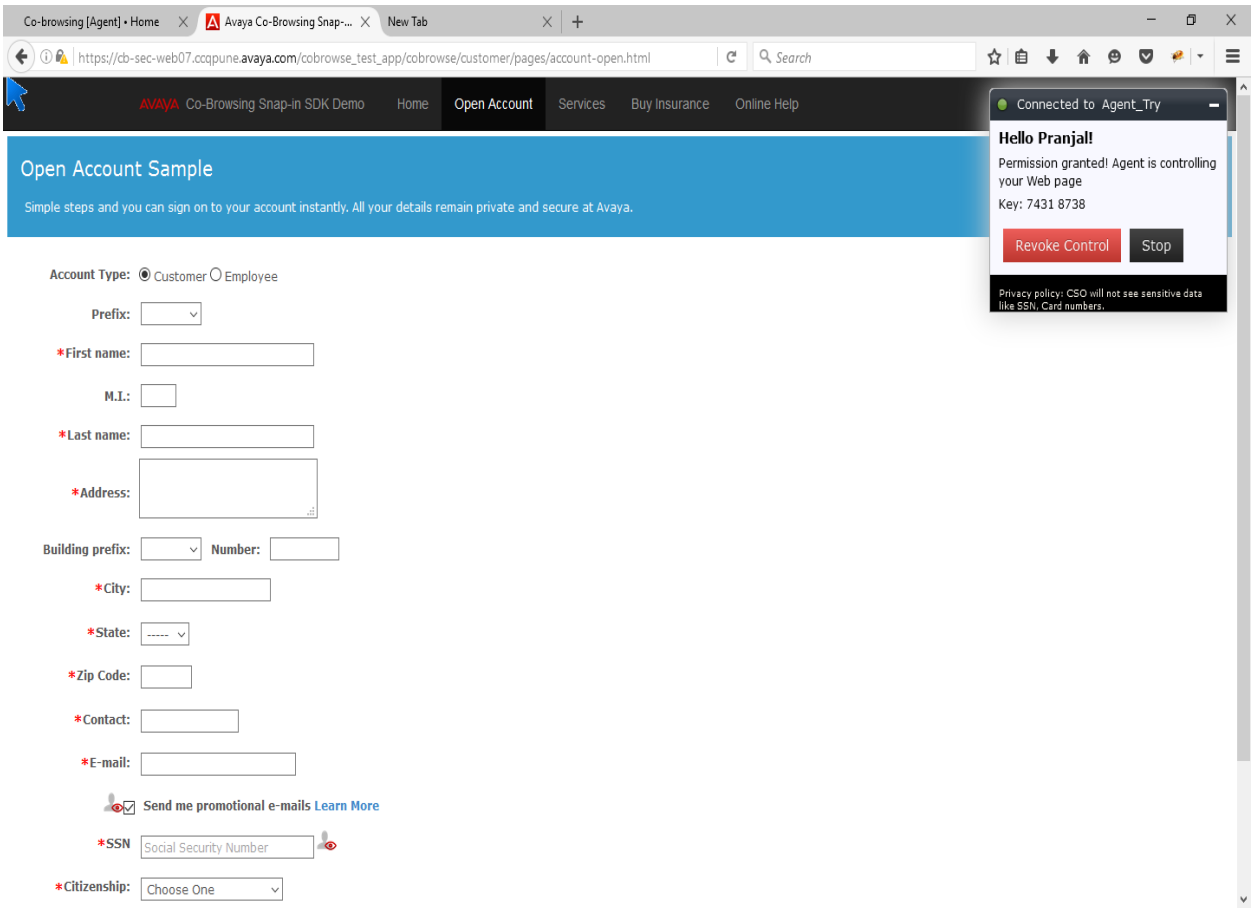
Customer fills the form:

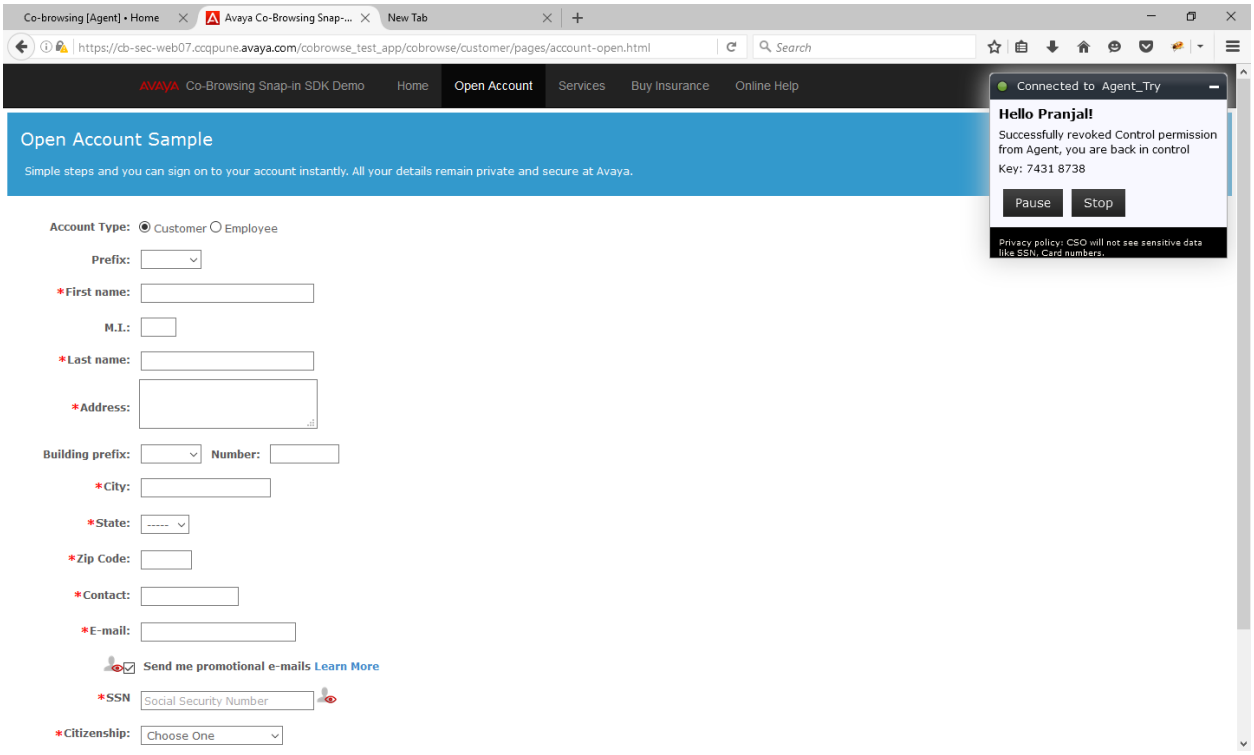
Customer starts filling the text fields one by one.Agent will be verifying whatever the customer is entering.If at any point,agent thinks that the details filled by the customer are wrong,then he can request control to the customer.If customer allows,then agent will get the control of the webpage & he can then edit the form accordingly.



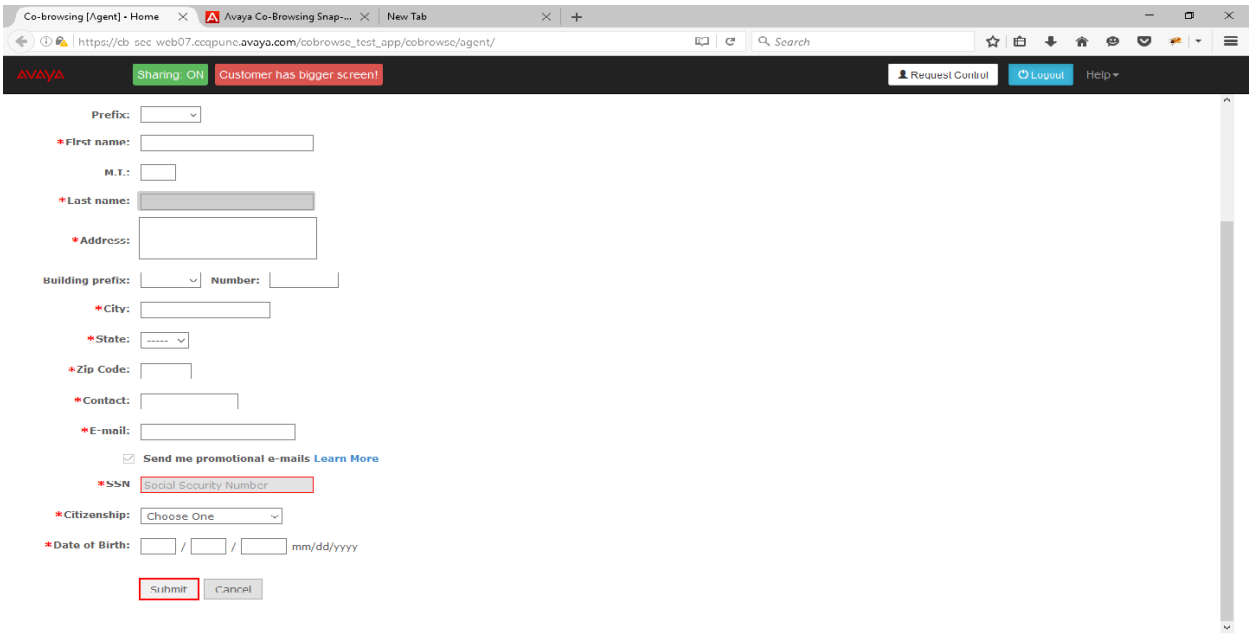
After agent requests control & the customer clicks on the allow button:

As can be seen from the screenshot,if at any point the customer wants to revoke the control,he can do so by clicking on the Revoke Control button.In this way,the control of the webpage will again get tranferred to the customer.After doing so,this message will be displayed:”Successfully revoked control permission from agent,You are back in control!”


Co-browsing [Agent] • Home
Avaya Co-Browsing Snap-...
New Tab
https://cb-sec-web07.ccqpune.avaya.com/cobrowse_test_app/cobrowse/customer/pages/account-open.html
Search
AVAYA Co-Browsing Snap-in SDK Demo
Home
Open Account
Services
Buy Insurance
Online Help
Connected to Agent_Try
Hello Pranjal!
Permission granted! Agent is controlling your Web page
Key: 7431 8738
Revoke Control
Stop
Privacy policy: CSO will not see sensitive data like SSN, Card numbers.
Open Account Sample
Simple steps and you can sign on to your account instantly. All your details remain private and secure at Avaya.
Account Type: Customer Employee
Prefix:
*First name:
M.I.:
*Last name:
*Address:
Building prefix:
Number:
*City:
*State:
*Zip Code:
*Contact:
*E-mail:
Send me promotional e-mails Learn More
*SSN
Social Security Number
*Citizenship:
Choose One

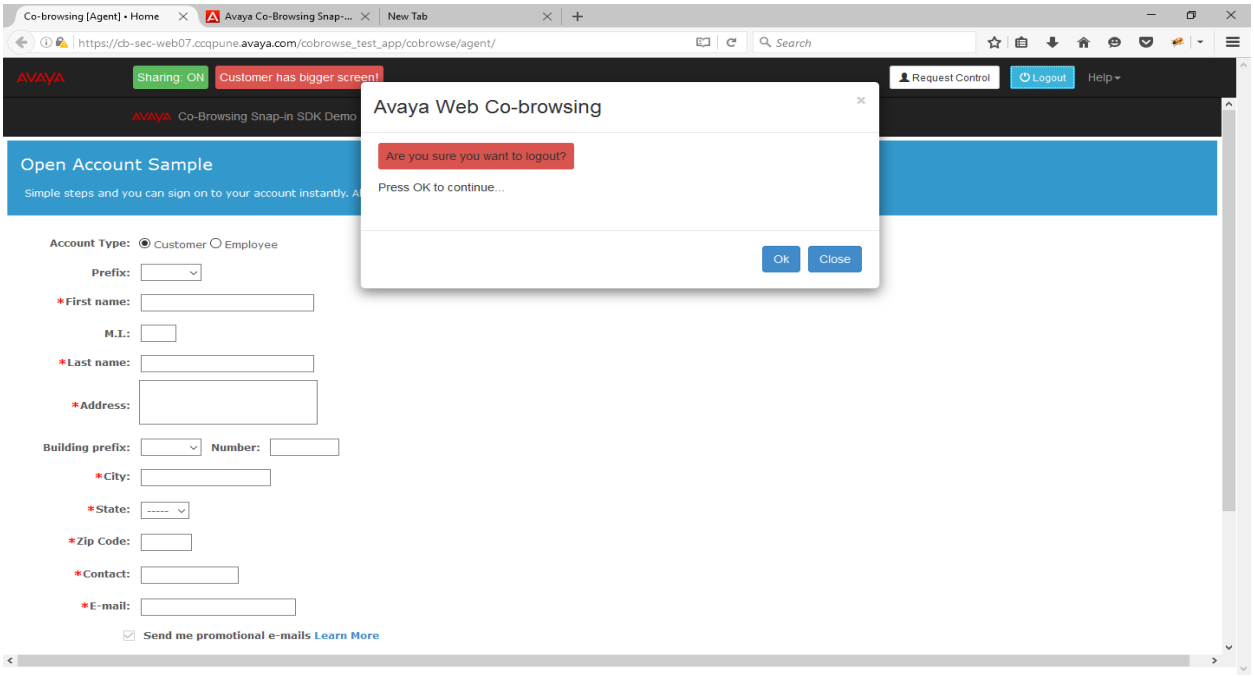


After filling all the details,the customer will now submit the form.(By clicking on the submit button)

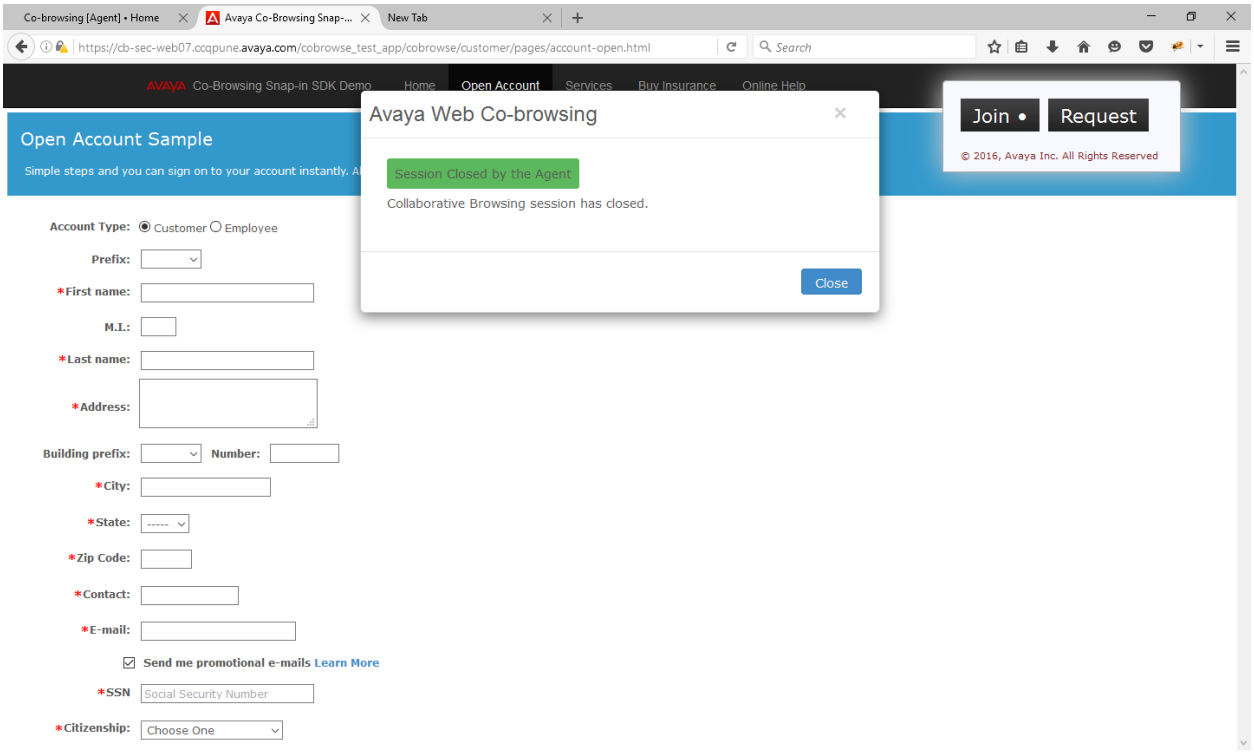


Agent Logs out:

After agent clicks on the logout button,a popup window will be displayed to confirm the logout(With a message:”Are you sure you want to logout?”)



At the same time,at the customer webpage,another popup will be opened telling him that the session is closed by the agent.



In this way,both agent and the customer are now on their respective initial pages.

Here,one session is completed.Likewise,I was asked to write code such that multiple sessions can be run in parallel instead of running only one session.Per session,two browser windows will get opened(one for agent and one for the customer).e.g,In order to run 10 sessions at a time,20 browser windows should get opened(10 for agents & 10 for customers).

The next task given to me was to run the sessions in loop.That is after one session is completed,(driver.quit() command is executed),another session should get started.Likewise,should continue for the specified number of times.

Another task given to me was to calculate the time it takes for a session to complete.Also,if number of sessions are run in parallel,then how much time it will take to complete all the sessions.Also,if the sessions are run in loop then per thread time calculation was done.To calculate time,

static long *startTime* = System.*nanoTime*();

And:

long endTime = System.*nanoTime*();

System.***out***.println("Took "+(endTime - *startTime*)/(1000000000) + " s");

Were written in the main driver script.The difference was divided by 10^9,in order to convert nanoseconds into seconds.

At the start of the main driver script, these were imported:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
```

Also,to open a new chrome-driver,

```
WebDriver driver1=new ChromeDriver();
WebDriver driver2=new ChromeDriver();
```

```
System.setProperty("webdriver.chrome.driver","C:\\Users\\deshmukh6\\Sel
enium\\chromedriver.exe");

//The path of Chromedriver was specified using above functionality.

//In order to copy the session key:


AgentDemo.Agent_CreateSession(driver1, driver2,name);

String sessionkey= AgentDemo.CopySessionKey(driver1, driver2,name);
sessionkey=sessionkey.substring(5, 12);

AgentDemo.Agent_JoinSession(driver1, driver2,name);


TryCustomerDemo.CustomerJoinsSession(driver1, driver2,sessionkey,name);
```

There are some browser tools that you can use in order to identify web elements in the webpage easier. These are:

- Firebug for Firefox
- Google Developer Tools for Chrome
- Web Inspector for Safari

Locating Elements with Selenium WebDriver:

findElement() method returns and WebElement and findElements() returns a list of WebElements.

1. By ID:

1 in Java: driver.findElement(By.id("element id"))

2. By CLASS:

1 in Java: driver.findElement(By.className("element class"))

3. By NAME:

1 in Java: driver.findElement(By.name("element name"))

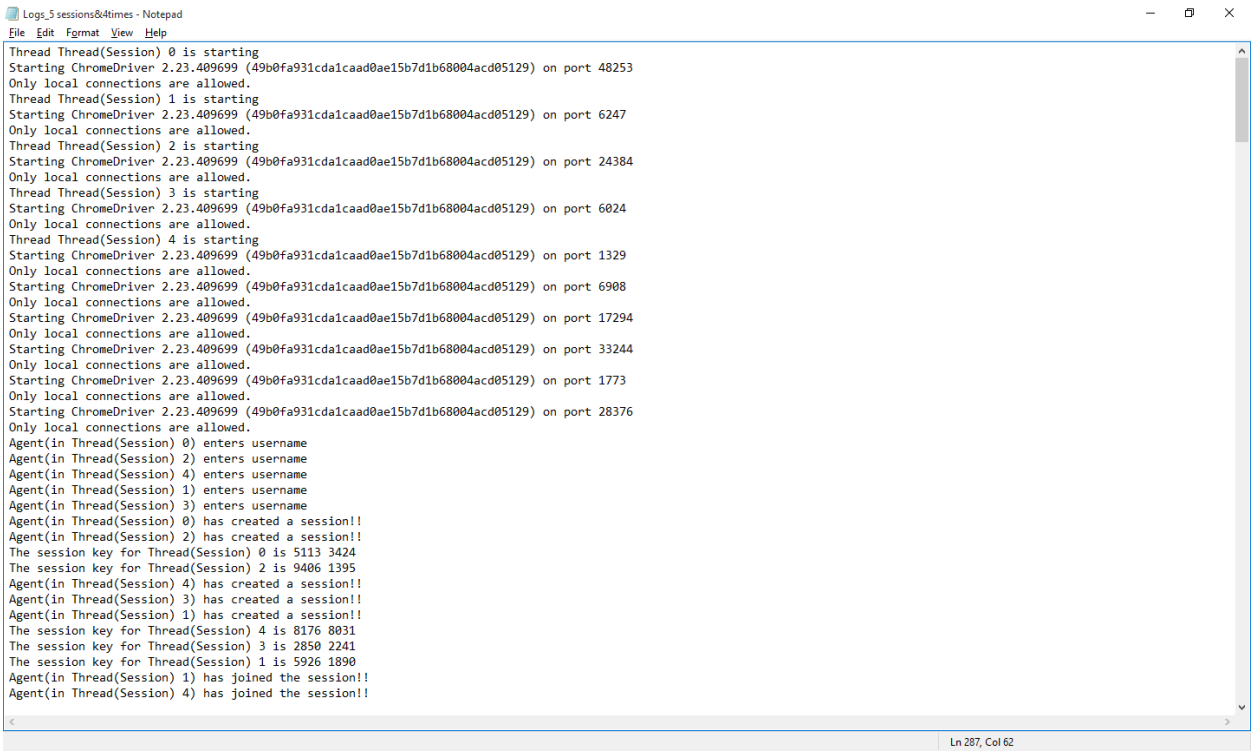
4. By XPath:

in Java: driver.findElement(By.xpath("xpath expression"))

Here,in the code which I have written,I have used element's id and xpath only.All these scripts written by me,will later be used for performance testing of co-browse

Conclusion(Logs Captured):

When I run the code by setting number of sessions to be run in parallel as 5 & the number of times they should be run as 4,the logs captured are shown below:

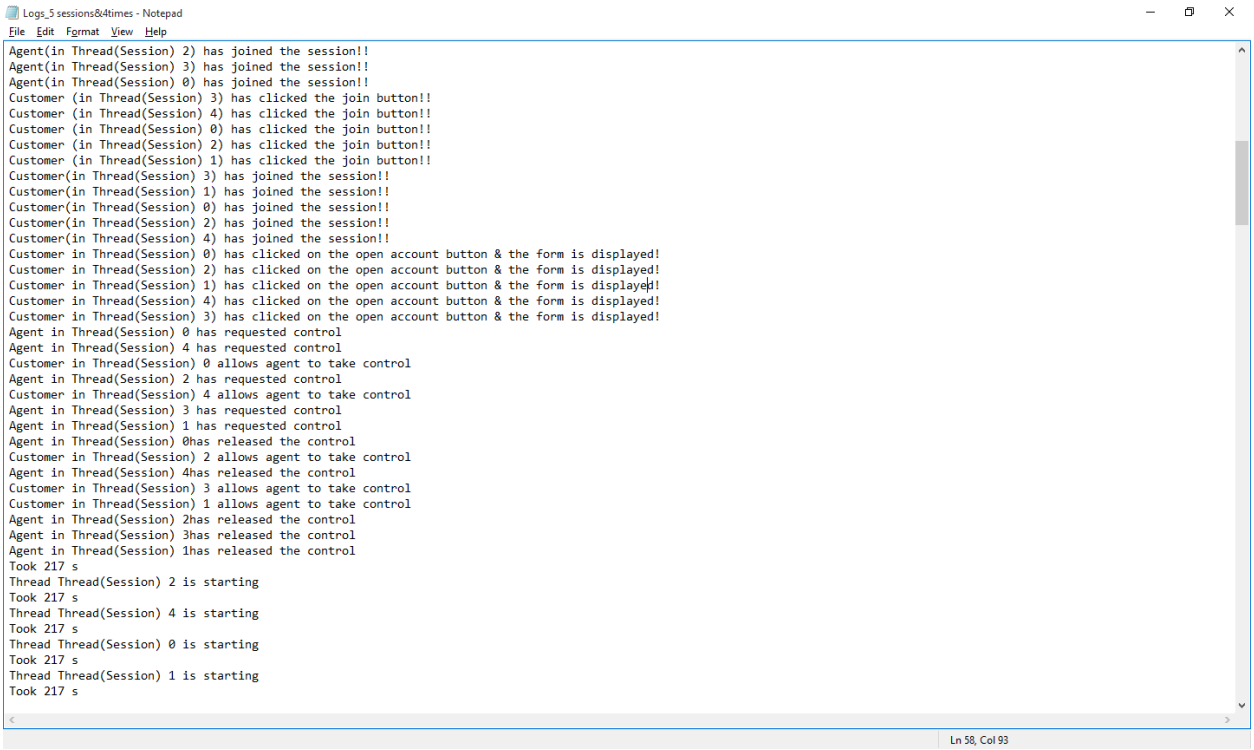


Logs_5 sessions&4times - Notepad

File Edit Format View Help

```
Thread Thread(Session) 0 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 48253
Only local connections are allowed.
Thread Thread(Session) 1 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 6247
Only local connections are allowed.
Thread Thread(Session) 2 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 24384
Only local connections are allowed.
Thread Thread(Session) 3 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 6024
Only local connections are allowed.
Thread Thread(Session) 4 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 1329
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 6908
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 17294
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 33244
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 1773
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 28376
Only local connections are allowed.
Agent(in Thread(Session) 0) enters username
Agent(in Thread(Session) 2) enters username
Agent(in Thread(Session) 4) enters username
Agent(in Thread(Session) 1) enters username
Agent(in Thread(Session) 3) enters username
Agent(in Thread(Session) 0) has created a session!!
Agent(in Thread(Session) 2) has created a session!!
The session key for Thread(Session) 0 is 5113 3424
The session key for Thread(Session) 2 is 9406 1395
Agent(in Thread(Session) 4) has created a session!!
Agent(in Thread(Session) 3) has created a session!!
Agent(in Thread(Session) 1) has created a session!!
The session key for Thread(Session) 4 is 8176 8031
The session key for Thread(Session) 3 is 2850 2241
The session key for Thread(Session) 1 is 5926 1890
Agent(in Thread(Session) 1) has joined the session!!
Agent(in Thread(Session) 4) has joined the session!!
```

Ln 287, Col 62



Logs_5 sessions&4times - Notepad

File Edit Format View Help

```
Agent(in Thread(Session) 2) has joined the session!!
Agent(in Thread(Session) 3) has joined the session!!
Agent(in Thread(Session) 0) has joined the session!!
Customer (in Thread(Session) 3) has clicked the join button!!
Customer (in Thread(Session) 4) has clicked the join button!!
Customer (in Thread(Session) 0) has clicked the join button!!
Customer (in Thread(Session) 2) has clicked the join button!!
Customer (in Thread(Session) 1) has clicked the join button!!
Customer(in Thread(Session) 3) has joined the session!!
Customer(in Thread(Session) 1) has joined the session!!
Customer(in Thread(Session) 0) has joined the session!!
Customer(in Thread(Session) 2) has joined the session!!
Customer(in Thread(Session) 4) has joined the session!!
Customer in Thread(Session) 0) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 2) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 1) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 4) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 3) has clicked on the open account button & the form is displayed!
Agent in Thread(Session) 0 has requested control
Agent in Thread(Session) 4 has requested control
Customer in Thread(Session) 0 allows agent to take control
Agent in Thread(Session) 2 has requested control
Customer in Thread(Session) 4 allows agent to take control
Agent in Thread(Session) 3 has requested control
Agent in Thread(Session) 1 has requested control
Agent in Thread(Session) 0has released the control
Customer in Thread(Session) 2 allows agent to take control
Agent in Thread(Session) 4has released the control
Customer in Thread(Session) 3 allows agent to take control
Customer in Thread(Session) 1 allows agent to take control
Agent in Thread(Session) 2has released the control
Agent in Thread(Session) 3has released the control
Agent in Thread(Session) 1has released the control
Took 217 s
Thread Thread(Session) 2 is starting
Took 217 s
Thread Thread(Session) 4 is starting
Took 217 s
Thread Thread(Session) 0 is starting
Took 217 s
Thread Thread(Session) 1 is starting
Took 217 s
```

Ln 58, Col 93

Logs_5 sessions&4times - Notepad

File Edit Format View Help

```
Took 217 s
Thread Thread(Session) 1 is starting
Took 217 s
Thread Thread(Session) 3 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 17708
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 17239
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 42310
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 15183
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 5986
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 12166
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 15988
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 20929
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 11745
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 3478
Only local connections are allowed.
Agent(in Thread(Session) 2) enters username
Agent(in Thread(Session) 4) enters username
Agent(in Thread(Session) 1) enters username
Agent(in Thread(Session) 3) enters username
Agent(in Thread(Session) 0) enters username
Agent(in Thread(Session) 4) has created a session!!
Agent(in Thread(Session) 2) has created a session!!
Agent(in Thread(Session) 1) has created a session!!
The session key for Thread(Session) 2 is 4499 3345
The session key for Thread(Session) 4 is 5029 9315
The session key for Thread(Session) 1 is 9171 4359
Agent(in Thread(Session) 3) has created a session!!
The session key for Thread(Session) 3 is 9263 6693
Agent(in Thread(Session) 0) has created a session!!
The session key for Thread(Session) 0 is 5234 5341
Agent(in Thread(Session) 2) has joined the session!!
Agent(in Thread(Session) 3) has joined the session!!
Agent(in Thread(Session) 4) has joined the session!!
```

Ln 73, Col 42

Logs_5 sessions&4times - Notepad

File Edit Format View Help

```
Agent(in Thread(Session) 0) has joined the session!!
Agent(in Thread(Session) 1) has joined the session!!
Customer (in Thread(Session) 2) has clicked the join button!!
Customer (in Thread(Session) 4) has clicked the join button!!
Customer (in Thread(Session) 3) has clicked the join button!!
Customer (in Thread(Session) 1) has clicked the join button!!
Customer (in Thread(Session) 0) has clicked the join button!!
Customer(in Thread(Session) 2) has joined the session!!
Customer(in Thread(Session) 1) has joined the session!!
Customer(in Thread(Session) 4) has joined the session!!
Customer(in Thread(Session) 0) has joined the session!!
Customer(in Thread(Session) 3) has joined the session!!
Customer in Thread(Session) 0) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 2) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 3) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 4) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 1) has clicked on the open account button & the form is displayed!
Agent in Thread(Session) 4 has requested control
Agent in Thread(Session) 3 has requested control
Agent in Thread(Session) 2 has requested control
Agent in Thread(Session) 1 has requested control
Agent in Thread(Session) 0 has requested control
Customer in Thread(Session) 4 allows agent to take control
Customer in Thread(Session) 3 allows agent to take control
Customer in Thread(Session) 2 allows agent to take control
Customer in Thread(Session) 1 allows agent to take control
Customer in Thread(Session) 0 allows agent to take control
Agent in Thread(Session) 4has released the control
Agent in Thread(Session) 3has released the control
Agent in Thread(Session) 2has released the control
Agent in Thread(Session) 1has released the control
Agent in Thread(Session) 0has released the control
Took 418 s
Thread Thread(Session) 4 is starting
Took 418 s
Thread Thread(Session) 0 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 5979
Only local connections are allowed.
Took 419 s
Thread Thread(Session) 3 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 5512
Only local connections are allowed.
```

Ln 148, Col 59

Logs_5 sessions&4times - Notepad

File Edit Format View Help

```
Only local connections are allowed.
Took 419 s
Thread Thread(Session) 2 is starting
Took 419 s
Thread Thread(Session) 1 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 44324
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 39329
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 24100
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 33090
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 37751
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 12743
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 46609
Only local connections are allowed.
Agent(in Thread(Session) 1) enters username
Agent(in Thread(Session) 4) enters username
Agent(in Thread(Session) 3) enters username
Agent(in Thread(Session) 1) has created a session!!
The session key for Thread(Session) 1 is 4784 5993
Agent(in Thread(Session) 4) has created a session!!
The session key for Thread(Session) 4 is 9318 1568
Agent(in Thread(Session) 3) has created a session!!
Agent(in Thread(Session) 0) enters username
Agent(in Thread(Session) 2) enters username
The session key for Thread(Session) 3 is 5425 1620
Agent(in Thread(Session) 0) has created a session!!
Agent(in Thread(Session) 2) has created a session!!
The session key for Thread(Session) 0 is 9302 0608
The session key for Thread(Session) 2 is 9380 8052
Agent(in Thread(Session) 1) has joined the session!!
Agent(in Thread(Session) 4) has joined the session!!
Agent(in Thread(Session) 3) has joined the session!!
Customer (in Thread(Session) 1) has clicked the join button!!
Customer (in Thread(Session) 4) has clicked the join button!!
Agent(in Thread(Session) 0) has joined the session!!
Customer (in Thread(Session) 3) has clicked the join button!!
Agent(in Thread(Session) 2) has joined the session!!
```

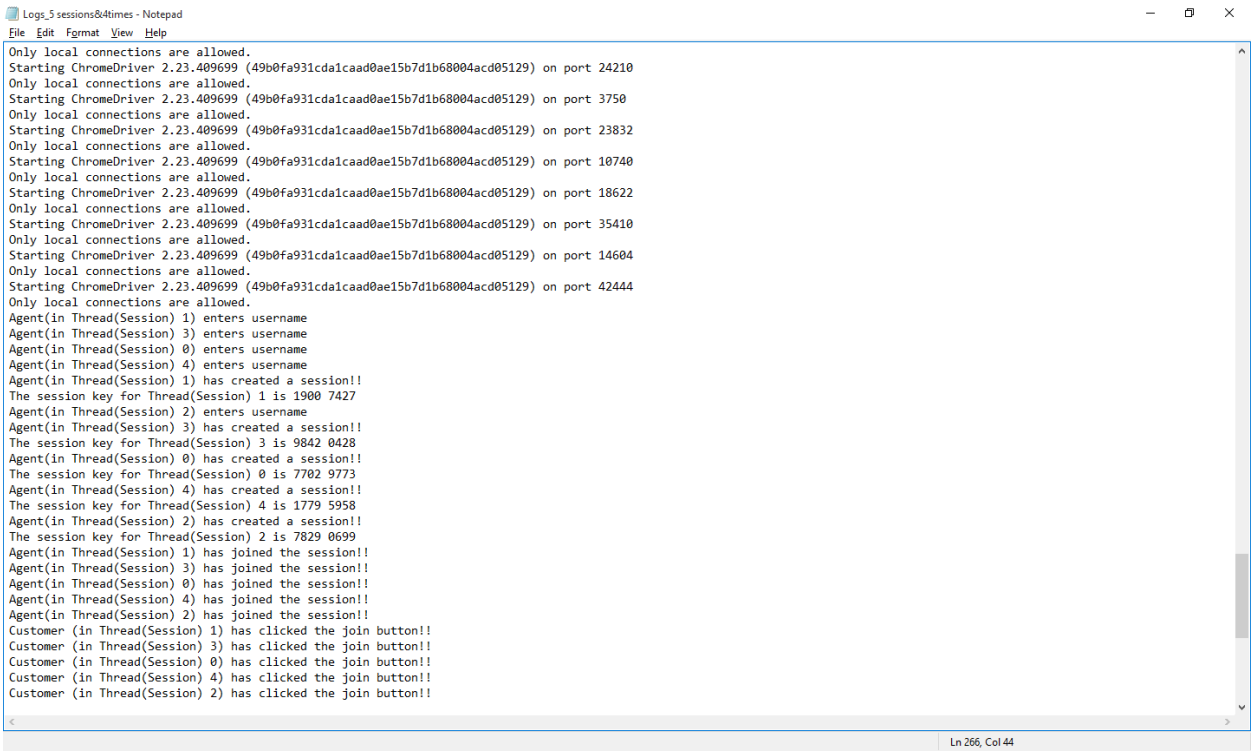
Ln 195, Col 44

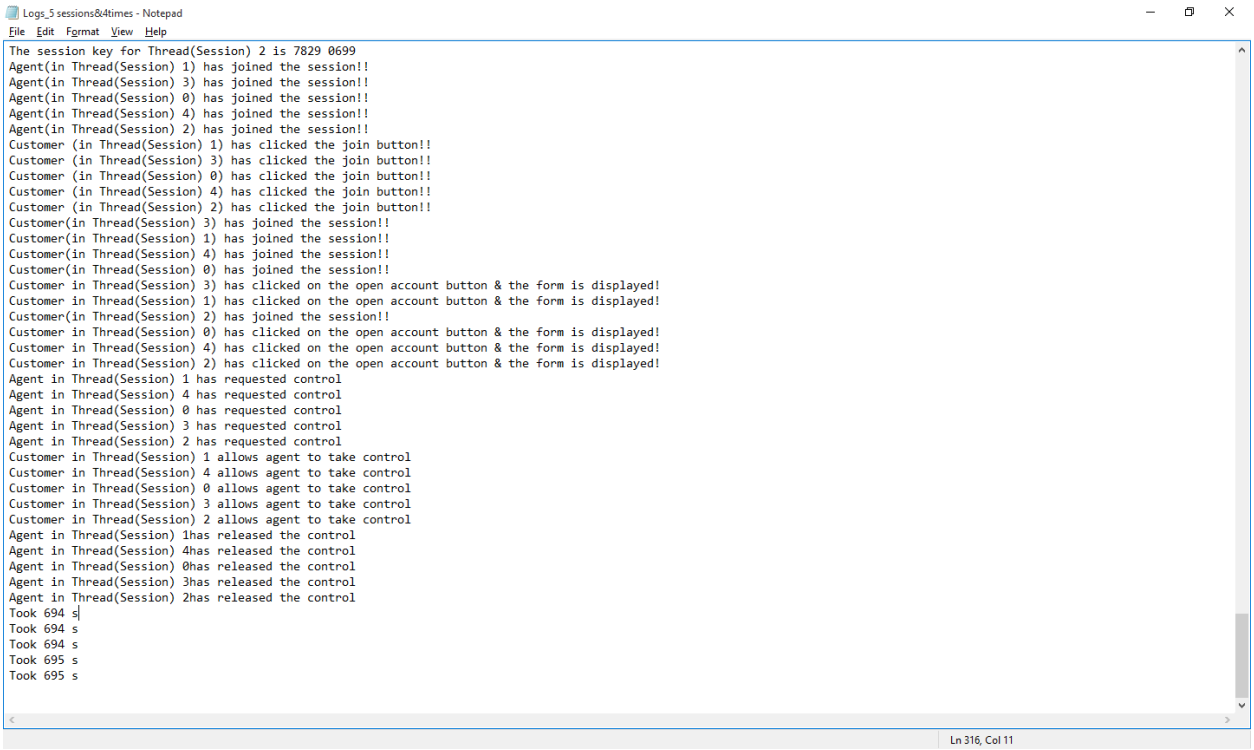
Logs_5 sessions&4times - Notepad

File Edit Format View Help

```
Agent(in Thread(Session) 2) has joined the session!!
Customer (in Thread(Session) 0) has clicked the join button!!
Customer (in Thread(Session) 2) has clicked the join button!!
Customer(in Thread(Session) 1) has joined the session!!
Customer(in Thread(Session) 4) has joined the session!!
Customer(in Thread(Session) 3) has joined the session!!
Customer in Thread(Session) 1) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 4) has clicked on the open account button & the form is displayed!
Customer(in Thread(Session) 0) has joined the session!!
Customer in Thread(Session) 3) has clicked on the open account button & the form is displayed!
Customer(in Thread(Session) 2) has joined the session!!
Customer in Thread(Session) 0) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 2) has clicked on the open account button & the form is displayed!
Agent in Thread(Session) 1 has requested control
Agent in Thread(Session) 4 has requested control
Agent in Thread(Session) 3 has requested control
Customer in Thread(Session) 1 allows agent to take control
Customer in Thread(Session) 4 allows agent to take control
Customer in Thread(Session) 3 allows agent to take control
Agent in Thread(Session) 0 has requested control
Agent in Thread(Session) 1has released the control
Agent in Thread(Session) 4has released the control
Agent in Thread(Session) 2 has requested control
Agent in Thread(Session) 3has released the control
Customer in Thread(Session) 0 allows agent to take control
Customer in Thread(Session) 2 allows agent to take control
Agent in Thread(Session) 0has released the control
Agent in Thread(Session) 2has released the control
Took 558 s
Thread Thread(Session) 4 is starting
Took 558 s
Thread Thread(Session) 1 is starting
Took 558 s
Thread Thread(Session) 3 is starting
Took 558 s
Thread Thread(Session) 2 is starting
Took 559 s
Thread Thread(Session) 0 is starting
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 30931
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 27807
Only local connections are allowed.
```

Ln 230, Col 49


Logs_5 sessions&4times - Notepad
File Edit Format View Help
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 24210
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 3750
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 23832
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 10740
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 18622
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 35410
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 14604
Only local connections are allowed.
Starting ChromeDriver 2.23.409699 (49b0fa931cda1caad0ae15b7d1b68004acd05129) on port 42444
Only local connections are allowed.
Agent(in Thread(Session) 1) enters username
Agent(in Thread(Session) 3) enters username
Agent(in Thread(Session) 0) enters username
Agent(in Thread(Session) 4) enters username
Agent(in Thread(Session) 1) has created a session!!
The session key for Thread(Session) 1 is 1900 7427
Agent(in Thread(Session) 2) enters username
Agent(in Thread(Session) 3) has created a session!!
The session key for Thread(Session) 3 is 9842 0428
Agent(in Thread(Session) 0) has created a session!!
The session key for Thread(Session) 0 is 7702 9773
Agent(in Thread(Session) 4) has created a session!!
The session key for Thread(Session) 4 is 1779 5958
Agent(in Thread(Session) 2) has created a session!!
The session key for Thread(Session) 2 is 7829 0699
Agent(in Thread(Session) 1) has joined the session!!
Agent(in Thread(Session) 3) has joined the session!!
Agent(in Thread(Session) 0) has joined the session!!
Agent(in Thread(Session) 4) has joined the session!!
Agent(in Thread(Session) 2) has joined the session!!
Customer (in Thread(Session) 1) has clicked the join button!!
Customer (in Thread(Session) 3) has clicked the join button!!
Customer (in Thread(Session) 0) has clicked the join button!!
Customer (in Thread(Session) 4) has clicked the join button!!
Customer (in Thread(Session) 2) has clicked the join button!!
Ln 266, Col 44


Logs_5 sessions&4times - Notepad
File Edit Format View Help
The session key for Thread(Session) 2 is 7829 0699
Agent(in Thread(Session) 1) has joined the session!!
Agent(in Thread(Session) 3) has joined the session!!
Agent(in Thread(Session) 0) has joined the session!!
Agent(in Thread(Session) 4) has joined the session!!
Agent(in Thread(Session) 2) has joined the session!!
Customer (in Thread(Session) 1) has clicked the join button!!
Customer (in Thread(Session) 3) has clicked the join button!!
Customer (in Thread(Session) 0) has clicked the join button!!
Customer (in Thread(Session) 4) has clicked the join button!!
Customer (in Thread(Session) 2) has clicked the join button!!
Customer(in Thread(Session) 3) has joined the session!!
Customer(in Thread(Session) 1) has joined the session!!
Customer(in Thread(Session) 4) has joined the session!!
Customer(in Thread(Session) 0) has joined the session!!
Customer in Thread(Session) 3) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 1) has clicked on the open account button & the form is displayed!
Customer(in Thread(Session) 2) has joined the session!!
Customer in Thread(Session) 0) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 4) has clicked on the open account button & the form is displayed!
Customer in Thread(Session) 2) has clicked on the open account button & the form is displayed!
Agent in Thread(Session) 1 has requested control
Agent in Thread(Session) 4 has requested control
Agent in Thread(Session) 0 has requested control
Agent in Thread(Session) 3 has requested control
Agent in Thread(Session) 2 has requested control
Customer in Thread(Session) 1 allows agent to take control
Customer in Thread(Session) 4 allows agent to take control
Customer in Thread(Session) 0 allows agent to take control
Customer in Thread(Session) 3 allows agent to take control
Customer in Thread(Session) 2 allows agent to take control
Agent in Thread(Session) 1has released the control
Agent in Thread(Session) 4has released the control
Agent in Thread(Session) 0has released the control
Agent in Thread(Session) 3has released the control
Agent in Thread(Session) 2has released the control
Took 694 s
Took 694 s
Took 694 s
Took 695 s
Took 695 s
Ln 316, Col 11

References:

1. https://en.wikipedia.org/wiki/Avaya
2. http://toolsqa.wpengine.com/selenium-webdriver/
3. https://cb-sec-web07.ccqpune.avaya.com/cobrowse_test_app/cobrowse/agent/
4. https://cb-sec-web07.ccqpune.avaya.com/cobrowse_test_app/cobrowse/customer/