

VPN-Enabled Web Browser

Sam Messina, Greg Brinkman

Basic Idea

- Basic structure is an RMI client-server relationship
- A Java-based web browser would act as our client
- Our server would retrieve HTML from a website and send it back to the client
- Server and client are tied together via RMI
 - The "Remote Method" in this case gets the HTML for a given URL and returns it in a string

Iterations Of Our Project

Blissful Ignorance

- We stated out with the in-class web browser lab
- We were able to get a working prototype
- Swing-based JEditorPane cannot display HTML versions greater than 3.2
 - Pro: it loaded the old Space Jam site just fine
 - Con: it loaded essentially nothing else

Enter JavaFX: The Rich Client Platform

- JavaFX has a number of advanced UI-based classes
- We used JavaFX's Stage, WebView, and WebEngine to render our browser

O HyperLinkListener, HyperLinkListener! Wherefore art thou HyperLinkListener?

- While JEditorPane allows you to implement your own HyperLinkListener, WebEngine does not
 - This is a problem because clicking a link didn't send the request through our server

Solution? Add a click listener.

- Whenever a user clicks, the browser checks the clicked tag for an "href" attribute.
- If found, the browser navigates to the URL found in the href attribute, sending its request through our server

Stack to the Future!

- We use a stack for the back button
- As we move to a new link, push the old url to a stack.
- When you click back, pop off that stack and go to url returned from the stack

Moving Forward by Going Backward

- ForwardStack acts as a mirror to the backStack
- As you pop a url off the backStack, push that same url to the forwardStack and ViceVersa.

As It Stands

- Our Browser now sends all links and URLs through an external server using RMI.
- While static pages can be retrieved from HTML without hiccup
- Some images, javascript, css, etc get lost in the transfer

How To Run

Packages

- You will need the following packages:
 - JavaFX
 - Java 7 JDK/JRE
 - rmiregistry

Commands

Run the following commands from the root of the project:

- Server:

```
$ javac *.java
$ rmiregistry &
$ java Server &
```

- Client:

```
$ javac *.java
$ rmiregistry &
$ Java Browser SERVER_IP_ADDRESS
```

- The browser will pop up, and all requests will be displayed in the host terminal