Web Technology for Test and Automation Applications

Fanie Coetzer - FSE
Demo

Operator

Technician

Engineers

Your boss

Test Sequencer
Goal

A

I know nothing

B

I know what it takes to get started on web applications in LabVIEW
What is the Web?

A bunch of standards that are built upon the internet. If everyone follows the standards then we can communicate with each other.
Learning the Web

• W3C has >50 working groups making more standards everyday
• “The web is changing so fast and there are so many acronyms”

All of this is built upon common concepts and standards
Learning the Web

Concepts
A. client-server model
B. client application
C. web service
D. web browser

Standards
A. HTTP
B. URL
C. XML, JSON, Text
D. HTML
Concept: Client-server model

Resource requestor
- Personal computer
- Smart phone
- Tablet

Resource provider
- Large data server
- Automation machine
- PLC/PAC
Concept: Client Application

- **Client application**: the software running on the client which initiates communication with the server.

**Examples**:  
- Google Maps app on your smartphone  
- LabVIEW application
Concept: Web Service

Web service— the software running on the server which responds to the requests of the client.
Concept: Web Service

Web service— the software running on the server which responds to the requests of the client.
Concept: Web Service

Web service— the software running on the server which responds to the requests of the client.

LabVIEW 2013 Project
Target
Web Service
Web Methods

Ex.
#4 Concept : Web Browser

Web Browser: A specific type of client application that renders standard types of web content.

Rendering in this case means taking text and translating it into an image.
Hypertext Transfer Protocol is THE language of the web

Follows the client-server model for communication

9 main communication commands:
GET – get a resource
POST – manipulate a resource
PUT – place/replace resource
DELETE – remove resource
Uniform Resource Locator OR “Web Address”
scheme://domain:port/path?query_string

http://www.ni.com/exampleURL?presenter=Fanie
Standard: URL

Uniform Resource Locator OR “Web Address”
scheme://domain:port/path?query_string

Ex.
Standard: Text, XML, JSON

URL is the request, content is the reply

Text = ASCII or Unicode text
XML = eXtensible Markup Language
JSON = JavaScript Object Notation
Standard: Text, XML, JSON

URL is the request, content is the reply

Text = ASCII or Unicode text
XML = eXtensible Markup Language
JSON = JavaScript Object Notation

Hello World!!!
Standard: Text, XML, JSON

URL is the request, content is the reply

Text = ASCII or Unicode text
XML = eXtensible Markup Language
JSON = JavaScript Object Notation

XML

```xml
<?xml version="1.0"?>
<Measurement>
  <Value>1.234</Value>
  <Units>Volts</Units>
</Measurement>
```

JSON

```json
{
  "Measurement": {
    "Value": 1.234,
    "Units": "Volts"
  }
}
```
Standard: HTML

HTML = HyperText Markup Language

```html
<!DOCTYPE html>
<html>
<body>
    <a href="http://www.w3schools.com">
        This is a link
    </a>
    <img src="img/cat.gif">
</body>
</html>
```
Thin Client

Web Thin Client – A user interface provided by an application through a web browser with a focus on being platform independent

Features
- Cross platform
- Does not require anything to be installed on the client

NOTE: A “Fat Client” is a client that must be installed and may not be platform independent.
Thin Client process

1. Type URL in Web Browser
2. Request is routed to the Web Server
3. Web Server responds with Thin Client content
4. Browser receives content and renders Thin Client
5. Browser periodically asks for updates from Web Server
How to make a Thin Client?

Web Browser

HTTP

Web Server

Thin Client content
- HTML
- JavaScript
- Cascading Style Sheets
Thin Client

Open Source Projects

• jQuery – A popular library that makes JavaScript easier

• jQuery UI – A UI library built upon jQuery. Has Graph, Charts, etc.

• Twitter’s Bootstrap – Free library that makes positioning elements on a webpage easier. Ascetically pleasing
Advantages of LV WebSevices

- Multiple client support
- Connection management
- True Thin Clients
- LV Windows and RT support
- Built-in Authentication
- Built-in Encryption (SSL)
Advantages of LV WebServices

Authentication

HTTP Method VI Settings

<table>
<thead>
<tr>
<th>Web Service VI</th>
<th>Method</th>
<th>URL Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abort.vi</td>
<td>GET</td>
<td>/Abort</td>
</tr>
<tr>
<td>CPU.vi</td>
<td>GET</td>
<td>/CPU</td>
</tr>
<tr>
<td>index.vi</td>
<td>GET</td>
<td>/</td>
</tr>
<tr>
<td>Memory.vi</td>
<td>GET</td>
<td>/Memory</td>
</tr>
<tr>
<td>Results.vi</td>
<td>GET</td>
<td>/Results</td>
</tr>
<tr>
<td>Run.vi</td>
<td>GET</td>
<td>/Run</td>
</tr>
<tr>
<td>Snapshot.vi</td>
<td>GET</td>
<td>/Snapshot</td>
</tr>
</tbody>
</table>

Web Service VI Properties

Required permissions

WebMessageHandler

Require API key
Advantages of LV Web Services

Encryption
Get your Embedded Certification!

National Instruments is now offers the Certified LabVIEW Embedded Developer Exam

Learn more ni.com/cled

Email certification@ni.com To Schedule Your Exam
Questions?
Web Technology for Test and Automation Applications

- Concepts
  - Client-server model
  - Client application
  - Web service
  - Web browser

- Standards
  - HTTP
  - URL
  - XML, JSON, Text
  - HTML

- Thin Client development tips
  - JavaScript, jQuery, jQueryUI
  - CSS
  - Bootstrap