Presentation 2:

Introduction to Web Services

Analyze HTTP Request/Response
REST vs SOAP Request/Response

Objectives

- Using cURL to analyze HTTP request/response
- Explain the difference between REST vs SOAP
- Using Postman to analyze REST request/response
- Using Postman to analyze SOAP request/response
- XML Example
- JSON Example

HTTP Communication Protocol

- HTTP: Hyper Text Transfer Protocol is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems.
- Client sends an HTTP request which contains:
  - Method: GET, POST, PUT, DELETE….
  - Headers
  - Body Content
- Notice: HTTP Method, Headers and Body here are related to **request message structure** NOT to HTML code

Analyze HTTP Request/Response

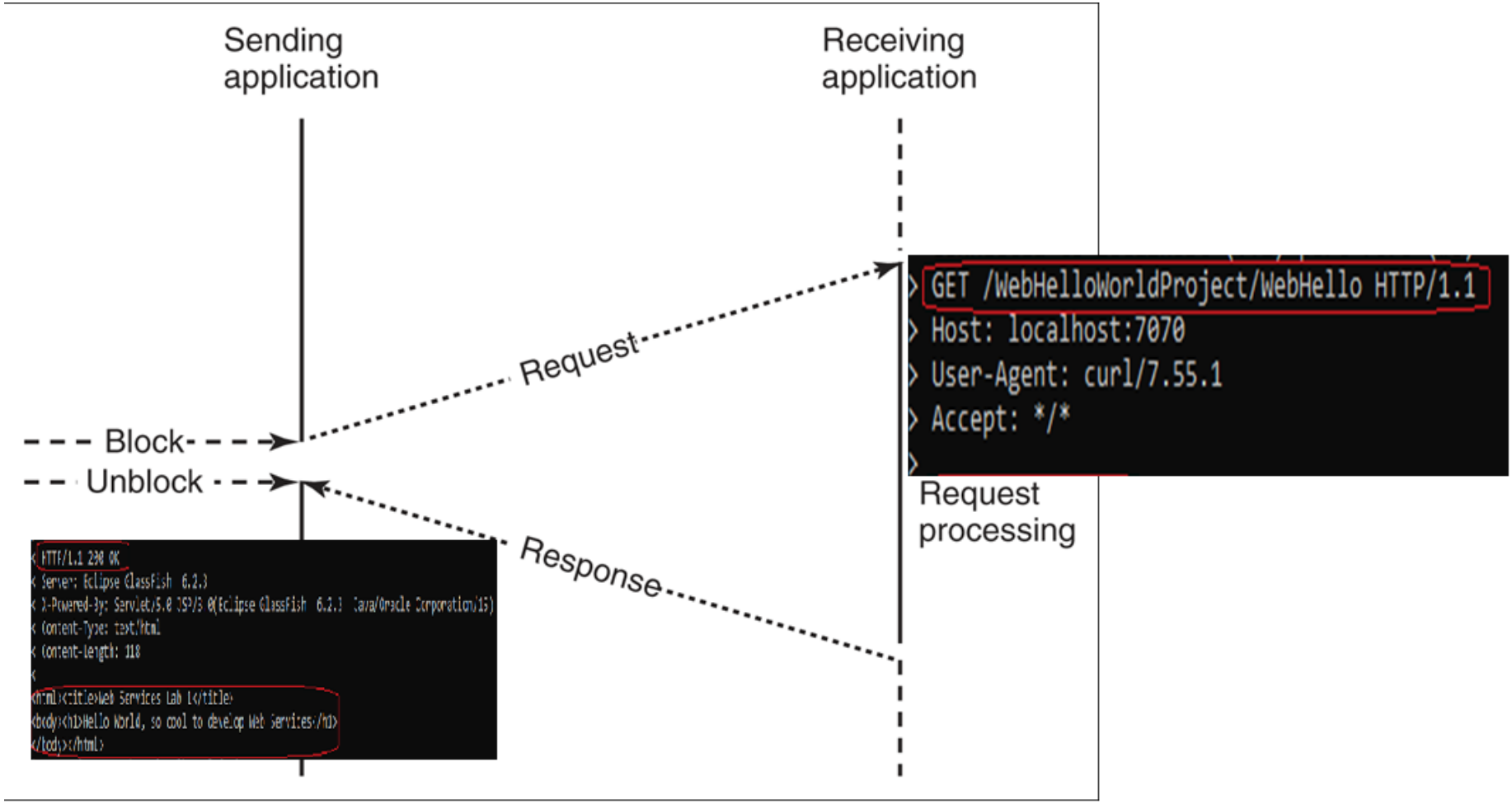


Figure Example of HTTP Request / Response Using CURL

Request from Browser

http://localhost:7070/WebHelloWorldProject/ WebHelloWorld

Analyze HTTP Request/Response using cURL

```
C:\windows\system32>curl -v http://localhost:7070/WebHelloWorldProject/WebHelloWorld
*   Trying ::1...
* TCP_NODELAY set
* Connected to localhost (::1) port 7070 (#0)
> GET /WebHelloWorldProject/WebHelloWorld HTTP/1.1
> Host: localhost:7070
> User-Agent: curl/7.55.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: Eclipse GlassFish  6.2.3
< X-Powered-By: Servlet/5.0 JSP/3.0(Eclipse GlassFish  6.2.3  Java/Oracle Corporation/15)
< Content-Type: text/html
< Transfer-Encoding: chunked
<
<body>Hello World, So cool to develop Web Services
<h1>Vanier College</h1>
<table border="1"><tr>
<td>Prd Price</td><td>Prd Qty</td><td>Total Billing</td></tr>
<tr><td>5.99</td><td>6</td><td><b>35.94</b></td></tr></table>
<br/>Montreal Uni
<br/>2022
<br/>3</body>
* Connection #0 to host localhost left intact
```

Request from Browser

http://localhost:7070/WebHelloWorldProject/WebHelloWorld

Analyze HTTP Request/Response using Postman

Postman

File Edit View Help

Home Workspaces API Network Reports Explore Search Postman

My Workspace New Import

Collections
APIs
Environments
Mock Servers
Monitors
Flows
History

Lab 2 Testing
- GET WebHelloWorldTesting
- GET Public REST Test 1
- GET Public REST Test 2
- GET Public REST Test 3
- POST Public SOAP Test 4
- POST Public SOAP Test 5
- POST Public SOAP Test 6

Lab 2 Testing / WebHelloWorldTesting

GET http://localhost:7070/WebHelloWorldProject/WebHelloWorld

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Headers 6 hidden

KEY	VALUE	DESC
Content-Type	text/html	
Key	Value	Desc

Body Cookies Headers (4) Test Results Status: 200 OK

Pretty Raw Preview Visualize HTML

```
<body>Hello World, So cool to develop Web Services
    <h1>Vanier College</h1>
    <table border="1">
        <tr>
            <td>Prd Price</td>
            <td>Prd Qty</td>
            <td>Total Billing</td>
        </tr>
        <tr>
            <td>5.99</td>
            <td>6</td>
```

REST API

- Using Postman to analyze HTTP request/response
- REST stands for Representational State Transfer
- REST is an architectural pattern
- REST uses Uniform Service locators to access to the components and resource
- REST is stateless
- REST was designed specifically for working with components supporting plain text, XML, HTML and JSON

REST Functionality

- FETCH Resource
  - Using method request GET.
- CREATE Resource
  - Using method request POST.
- UPDATE Resource
  - Using method request PUT.
- DELETE Resource
  - Using method request DELETE.

Analogy of REST Resources with Database Operations.

Web Server (GlassFish)

Client

WEB Application

GET http://....
POST htpp://...
PUT http://...
DELETE http://

-Servlet
-REST Resources
-SOAP Resources

-Classes (State in REST could be one class or set of classes)

Example of State in REST
1)GET
http://localhost/appName/books/
State: List of all books
Transfert: sending all books to Cli
State is one object? State is complex set of objects
Representational: as set of classes
Book Info → Book class
Publisher Info → Publisher class

Media Types
-Text
-HTML
-JSON
-XML

Database Server

Database CRUD

SELECT .... //Retrieve Records
INSERT .... //Insert new Record
UPDATE .... //Update existing Record
DELETE .... //DELETE Record

XML Example Format for Data Exchange between Client and Server

- SOAP only works with XML formats
- XML is a set of customized elements.

book.xml

```
<book>
  <book_id>10</book_id>
  <book_name>Web Services</book_name>
</book>
```

JSON Example Format for Data Exchange between Client and Server

- REST was designed specifically for working with components supporting plain text, XML, HTML and JSON
- JSON is a set of Key: value

book.json

```
{
    "book_id": "10",
    "book_name": "Web Services"
}
```

REST Resources

- Resource example:

GET http://localhost/appName/books/

GET http://localhost/appName/books/10

POST http://localhost/appName/book/11

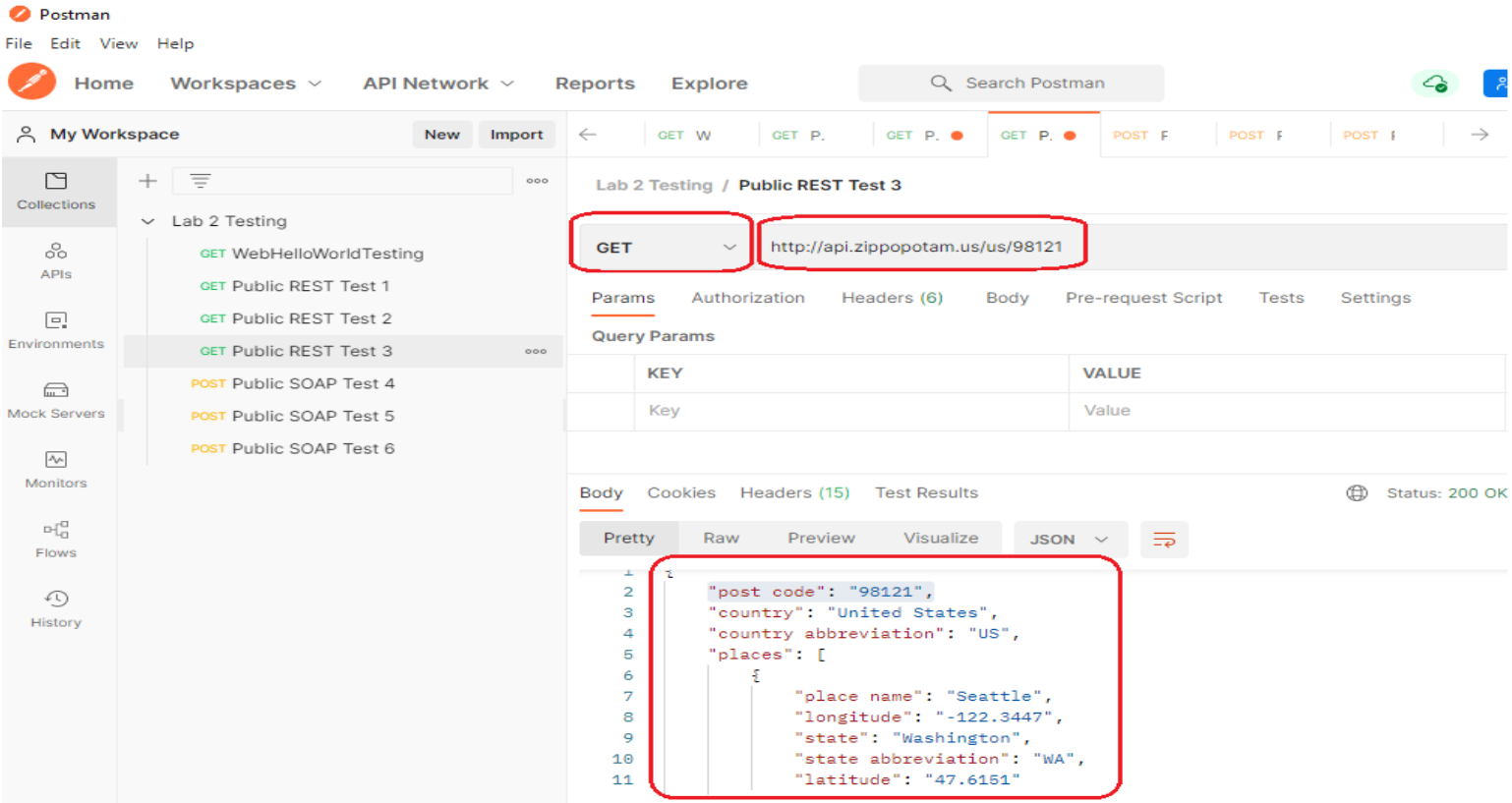
PUT http://localhost/appName/publishers/4

DELETE http://localhost/appName/book/10

Consuming Public REST API

- Using Postman to analyze HTTP request/response

http://api.zippopotam.us/us/98121



REST Body JSON Response Example

- REST is supporting many media type such as plain text, XML, HTML and JSON

Body Cookies Headers (15) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON


```
1  {
2      "post code": "98121",
3      "country": "United States",
4      "country abbreviation": "US",
5      "places": [
6          {
7              "place name": "Seattle",
8              "longitude": "-122.3447",
9              "state": "Washington",
10             "state abbreviation": "WA",
11             "latitude": "47.6151"
12         }
13     ]
14 }
```

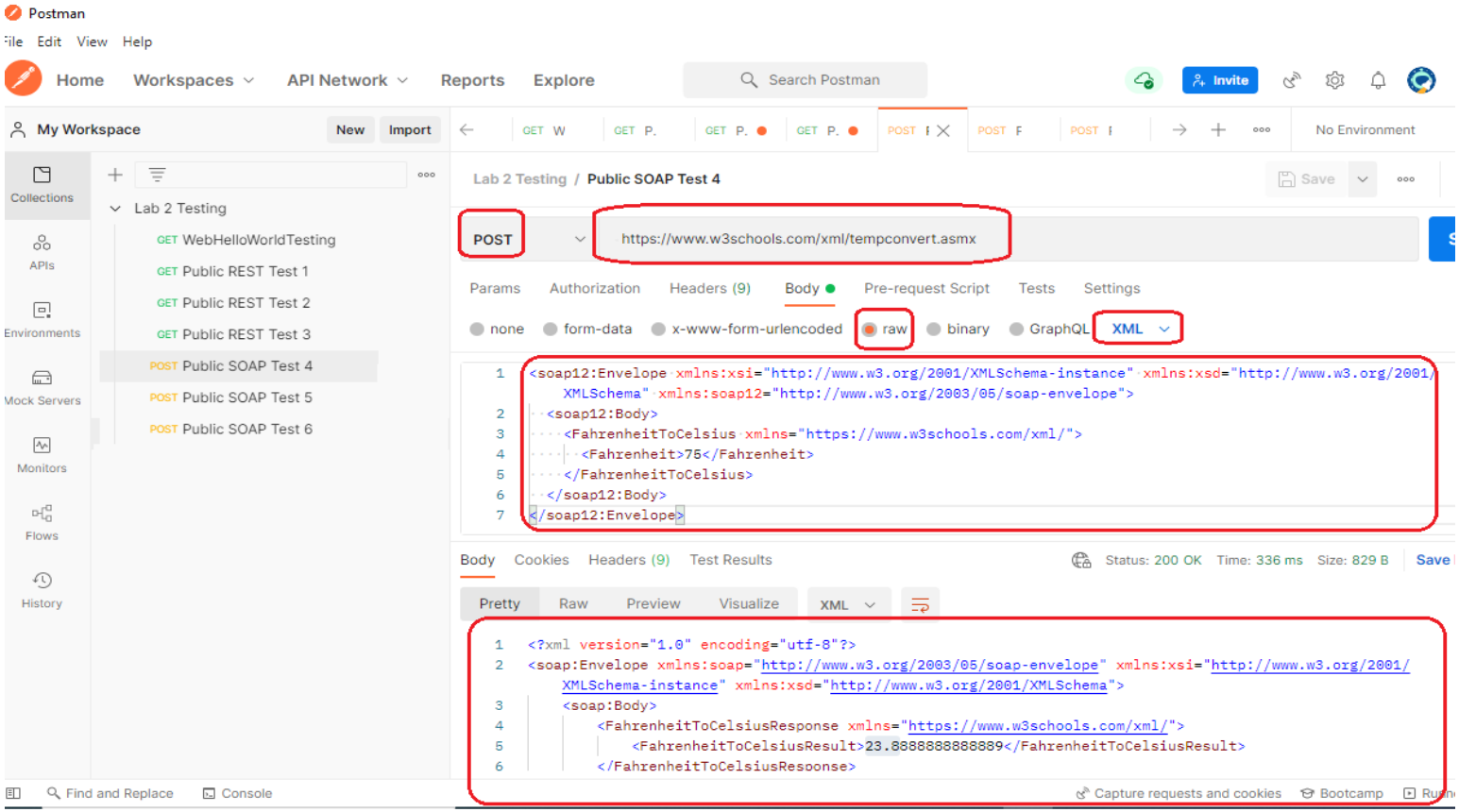
SOAP API

- Using Postman to analyze SOAP request/response
- SOAP stands for Simple Object Access Protocol
- SOAP is a protocol
- SOAP uses service interfaces to expose its functionality to client applications.
- SOAP is platform independent, can be stateful
- SOAP only works with XML formats
- SOAP cannot make use of REST whereas REST can make use of SOAP

Consuming Public SOAP API

- Using Postman to analyze SOAP request/response

https://www.w3schools.com/xml/tempconvert.asmx



SOAP XML Body Request Example

- SOAP only works with XML formats

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
 <soap12:Body>
    <FahrenheitToCelsius xmlns="https://www.w3schools.com/xml/">
       <Fahrenheit>75</Fahrenheit>
    </FahrenheitToCelsius>
</soap12:Body>
</soap12:Envelope>
```

SOAP XML Body Response Example

- SOAP only works with XML formats

Body Cookies Headers (9) Test Results Status: 200 OK Time: 336 ms Size: 829 B

Pretty Raw Preview Visualize XML

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/
    XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
3     <soap:Body>
4         <FahrenheitToCelsiusResponse xmlns="https://www.w3schools.com/xml/">
5             <FahrenheitToCelsiusResult>23.8888888888889</FahrenheitToCelsiusResult>
6         </FahrenheitToCelsiusResponse>
7     </soap:Body>
8 </soap:Envelope>
```

Links of used Public REST examples

- Used Public REST in Lab 2 YouTube Video:
  - http://openlibrary.org/search.json?q=the+lord+of+the+rings
  - https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2020-01-01&endtime=2020-01-02
  - http://api.zippopotam.us/us/98121

Links of used Public SOAP examples

- Used Public SOAP in Lab 2 YouTube Video:
  - https://www.w3schools.com/xml/tempconvert.asmx
  - http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso

    SOAP XML Body Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListOfContinentsByName xmlns="http://www.oorsprong.org/websamples.countryinfo">
    </ListOfContinentsByName>
  </soap12:Body>
</soap12:Envelope>
```

  - http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso

    SOAP XML Body Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListOfCountryNamesByName xmlns="http://www.oorsprong.org/websamples.countryinfo">
    </ListOfCountryNamesByName>
  </soap12:Body>
</soap12:Envelope>
```

Summary

- Using cURL to analyze HTTP request/response used in WebHelloWorldProject
- Using Postman to analyze HTTP request/response used in WebHelloWorldProject
- Explain the difference between REST vs SOAP
- Using Postman to analyze Public REST request/response
- Using Postman to analyze Public SOAP request/response
- Understanding Data Exchange Format XML
- Understanding Data Exchange Format JSON