# REST API Introduction



<sup>24-03-10</sup> CMPT 213

Slides 12

© Dr. B. Fraser

# Topics

How to request and send data to a server?
 How to design a server's API?

#### HTTP

## Overview

- Front-end = client-side; browser
- Back-end = server side
- Why make web-based app?
  - server to allow interaction between users
  - server to store resources or do heavy processing
  - centrally managed deployment and admin

## Server Interaction

- Browser getting data from webserver
  - browser does HTTP GET on URL
  - server sends back a web page (HTML, CSS, JS)
- Font-end/Back-end Interaction
  - client-side makes requests to server's RESTful API's endpoints (URLS)
  - data transmitted in JSON (or XML)

# HTTP

- HTTP:..
- URL:..

- Ex: http://www.sfu.ca/~bfraser/answers
<protocol>://<domain name>/<path>
<protocol>://<domain name>:<port>/<path></path>

Protocol ports

 HTTP: 80 (or 8080 alt)
 HTTPS: 443 (or 8443 alt)
 S = Secure

# **HTTP Methods**

- HTTP methods: What does the client want to happen at a URL?
- These are the..
  - retrieve some information from the URL: does not change server state
    - : Submit a new entity (object) to the URL
      - : Delete some entity (object) at the URL
  - : Replace an entity at the URL with new value
  - ... omitting HEAD, CONNECT, OPTIONS, TRACE, PATCH

## HTTP Response Status Codes

- Each request message (a GET, POST, ...) returns a response code:
  - 200:..
  - 201:..
  - 400: Bad Request (client-side error)
  - 401: Unauthorized (who are you?)
  - 403: Forbidden (I know who you are, but still not allowed)
  - 404:..
  - 500: Server-side error
  - (... many omitted!)

## Sending Data to the Server

- Front end can send data to the server via:
  - : Put data in path variables
    - Ex: GET http://my.com/api/person/5
      - : for GET only;
    - no raw special characters (Ex: %20 = space)
    - Ex: https://www.google.com/search?q=hi+world
      - : All HTTP messages have header
    - Ex: authentication or apiKey "ApiKey:abc123"
    - : Block of data (often text such as JSON)
      - Ex: {"name":"Dr. Evil","age":95,"laugh":"Mwahah"}

## **URL** Path Variables Details

#### • Path Variable Idea

 URL encodes groups or categories as though they are "folders", and items as "files"

#### • Example

https://coursys.sfu.ca/2050sp-cmpt-276-d1/students/hiwld

It seems like we are browsing into folders for a specific file

# **Query String Details**

- Query String: the common way to send data for GET
   Use to encode..
  - Ex: search queries
- Common Format http://my.com/s?key=value&otherkey=othervalue
- Demo

curl -k -i -X GET https://www.adafruit.com/?q=wire

### Request to Server & Reply



## Postman Request & Response

24-(

GET     https://www.adafruit.com/?q=wire     Send        Params     Authorization     Headers (7)     Body     Pre-request Script     Tests     Settings     Cookles       Query Params     VALUE     DESCRIPTION     ***     Bulk Edit       Image: Setting in the set of the	https	://www.adafr	uit.com/?q=wire					🖺 Save	~ / E										
Params       Authorization       Headers (7)       Body       Pre-request Script       Tests       Settings       Cookies         Query Params       KEY       VALUE       DESCRIPTION       ever       Bulk Edit         Body       Cookies (2)       Headers (20)       Test Results       @ 200 0K       S25 ms       51.48 KB       Save Response       Value         Body       Cookies (2)       Headers (20)       Test Results       @ 200 0K       S25 ms       51.48 KB       Save Response       Value         Date       Mon, 06 Mar 2023 08:11:15 GMT       Body       Cookies (2)       Headers (20)       Test Results       @ 200 0K       S25 ms       51.48 KB         Connection ©       Body       Cookies (2)       Headers (20)       Test Results       @ 200 0K       S25 ms       51.48 KB         Set-cookie ©       I       BitOCTYPE html       Image: "en-US">       Image: "en-U	GET	~	https://www.ad	afruit.com/?q=w	ire				Send ~	Send -									
KEY       VALUE       DESCRIPTION       eve       Bulk Edit         Image: constant of the second	Param Quer	s Author	ization Heade	ers (7) Body	Pre-request \$	Script Tests	Settings		Cookie	<b>A</b>									
Body       Cookles (2)       Headers (20)       Test Results       Cookles (2)       Response ~         KEY       VALUE         Date ©       Mon, 06 Mar 2023 08:11:15 GMT         Content-Type ©       Body       Cookles (2)       Headers (20)       Test Results         Connection ©       set-cookle ©       1       Blody Cookles (2)       Headers (20)       Test Results       Cookles (2)         Pretty       Raw       Preview       Visualize       HTML ~       Trap         1       Blody Cookles (2)       Headers (20)       Test Results       Cookles (2)       Cookles (2)         Set-cookle ©       1       Blody Cookles (2)       Headers (20)       Test Results       Trap         2       chtml lang="en-US">         3       chtml lang="en-US">         4       chead>       Set-cookle ©       Set-cook		KEY q			VALUE			DESCRIPTION	••• Bulk Ed	••• Bulk Edit									
Body       Cookles (2)       Headers (20)       Test Results       Concession         KEY       VALUE         Date ③       Mon, 06 Mar 2023 08:11:15 GMT         Content-Type ④       Body       Cookles (2)       Headers (20)         Transfer-Encoding ④       Body       Cookles (2)       Headers (20)         Transfer-Encoding ④       Body       Cookles (2)       Headers (20)         Set-cookle ④       1       Concent-Type ④       Electron Lange "en-US">         set-cookle ④       2 <html lang="en-US">         set-cookle ④       3       <html lang="en-US">         set-cookle ④       5       cmeta name="globalsign-domain-verification" content="395EvTKgTnwb20iKcv68nItcl7lbY_JFqavc5s         6       cmeta name="globalsign-domain-verification" content="395EvTKgTnwb20iKcv68nItcl7lbY_JFqavc5s</html></html>																			
Body       Cookles (2)       Headers (20)       Test Results       QALUE         VALUE       VALUE       VALUE         Date ③       Mon, 06 Mar 2023 08:11:15 GMT         Content-Type ④       Body       Cookles (2)       Headers (20)         Transfer-Encoding ④       Pretty       Raw       Preview       Visualize         Connection ④       1       @IDOCTYPE       html       Image: Preview       Image: Preview         1       @IDOCTYPE       html       Image: Preview       Visualize       HTML       Image: Preview         2 <html< td="">       lang="en-US"&gt;       Image: Preview       Visualize       HTML       Image: Preview       Image: Pre</html<>	_			_															
Date ③       Mon, 06 Mar 2023 08:11:15 GMT         Content-Type ④       Body Cookies (2) Headers (20) Test Results       © 200 OK 525 ms 51.48 KB         Transfer-Encoding ④       Pretty Raw Preview Visualize HTML        >         Connection ④       1       Clocotion Encoding Encoded       Image: Ten US">>         set-cookie ④       3            set-cookie ④       3            set-cookie ④       5            set-cookie ④       5            set-cookie ④       5            set-cookie ④       5             Set-cookie ④       3             set-cookie ④       5             Set-cookie ④       5             Set-cookie ④       3              Set-cookie ④       5               Set-cookie ④       6	Body	Cookies (2) Headers (20) Test Results																	
Content-Type ③       Body Cookies (2) Headers (20) Test Results       @ 200 OK 525 ms 51.48 KB         Transfer-Encoding ③       Pretty       Raw       Preview       Visualize         Connection ④       1       @ 1DOCTYPE html ⑧       2 <html lang="en-US">         set-cookie ④       2       <html lang="en-US">         set-cookie ④       3       <html lang="en-US">         set-cookie ④       5       <meta content="width=device-width, initial-scale=1.0" globalsign-domain-verification"="" name="globalsign-domain-verification" uterport"=""/>         7       <meta content="Adafruit Industries" name="author"/></html></html></html>		Date ④				Mon. 06 Mar 2023 08:11:15 GMT													
Content-type ()       Body Cookles (2) Headers (20) Test Results       Cache-control ()         Image: Transfer-Encoding ()       Pretty       Raw       Preview       Visualize       HTML ()       Image: Test Results       Image		Content Tuna @				Won, 00 War 2020	00.11.10 0141												
Transfer-Encoding ①       Pretty       Raw       Preview       Visualize       HTML       TP         Set-cookie ①       1       ③:DOCTYPE       html ②       1       ④:DOCTYPE       html ③         set-cookie ①       2 <html lang="en-US">       3         set-cookie ①       3       4       <head>         set-cookie ①       5       <meta content="width=device-width, initial-scale=1.0" name="globalsign-domain-verification" viewport"=""/>         cache-control ③       8       <meta content="Adafruit Industries" name="author"/></head></html>		Body Cook			ies (2) Headers (20) Test Results			200 OK	😤 200 OK 525 ms 51.48 KB										
Connection ①       Pretty       Raw       Pretty       Number of the state of the		Transfer-Encoding (1)		Daw Draview Viewelize 1774															
set-cookie ①       1       IDOCTYPE html         set-cookie ①       2 <html lang="en-US">         set-cookie ①       3         set-cookie ①       3         set-cookie ①       4         set-cookie ①       5         set-cookie ①       5         cmeta name="globalsign-domain-verification" content="395EvTKgTnwb20iKcV68nItcl7lbY_JFqavc5s         expires ③       6         cache-control ④       8         cache-control ④       8</html>		Connection ()		Raw Previe	w visualize	HIML V	- <del>e</del>												
set-cookie ③       2 <html lang="en-US">         set-cookie ④       3         set-cookie ④       4         set-cookie ④       5         expires ④       5         cache-control ④       8         cache-control ④       8</html>		set-cookie (i) 1		DOCTYPE html															
set-cookie ③       4 <head>         set-cookie ④       4       <head>         expires ④       5       <meta content="width=device-width, initial-scale=1.0" name="globalsign-domain-verification" viewport"=""/>         7       <meta charset="utf-8"/>         8       <meta content="Adafruit Industries" name="author"/></head></head>		set-cookie (i)		html lang="en-US">															
expires ③       6 <meta content="width=device-width, initial-scale=1.0" name="viewport"/> cache-control ③       6 <meta charset="utf-8"/> cache-control ④       8 <meta content="Adafruit Industries" name="author"/>		set-cookie (3)		4 <he< th=""><th colspan="5"><pre>4 <head> 5 <meta content="395EvTKgTnwb2&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;nItcl7lbY_JFqavc5s&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;cache-control ③       7       &lt;meta charset=" name="globalsign-domain-verification" utf-8"=""/>         8       <meta content="Adafruit Industries" name="author"/><th></th><td colspan="2">expires (i) 6</td><td>6</td><td colspan="6"><meta content="width=device-width, initial-scale=1.0" name="viewport"/></td></head></pre></th></he<>	<pre>4 <head> 5 <meta content="395EvTKgTnwb2&lt;/pre&gt;&lt;/th&gt;&lt;th&gt;nItcl7lbY_JFqavc5s&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;cache-control ③       7       &lt;meta charset=" name="globalsign-domain-verification" utf-8"=""/>         8       <meta content="Adafruit Industries" name="author"/><th></th><td colspan="2">expires (i) 6</td><td>6</td><td colspan="6"><meta content="width=device-width, initial-scale=1.0" name="viewport"/></td></head></pre>						expires (i) 6		6	<meta content="width=device-width, initial-scale=1.0" name="viewport"/>					
	3	cache-control (i) 7		<meta charset="utf-8"/> <meta content="Adafruit Industries" name="author"/>															

# HTTP Body details

- HTTP messages can include a body
  - Used by POST and PUT to send data
  - Often a JSON structure or binary data



GET /~bfraser/ HTTP/1.1 Host: www.sfu.ca Connection: keep-alive Cache-Control: no-cache User-Agent: Mozilla/5.0 ... Accept: text/html,application/...



24-03-10

Image: http://netlab.ulusofona.pt/rc/book/2-application/2\_02/index.htm 14

<title>Index of /~bfraser</title>...

#### **REST API**

# **API & REST**

• API:..

 How a program exposes its functionality for other programs to use.

• REST:..

- It works with HTTP caching and semantics to improve performance
- REST is founded on some principles, not a strict prescription.
   So what is "RESTful" is up to interpretation
- TLA: Three Letter Acronym

## Example Tic Tac Toe Model



## **REST Example**

- Example: Tic-tac-toe game
  - Base URL: my.com
  - /games GET (list), POST (new)
  - /games/52 GET (info), POST (change info)
  - /games/52/moves GET (list), POST (new)
  - /games/52/moves/1 GET (info), POST (change info)
- Full Example
  - GET my.com/games/52/moves/1
    - In games API, retrieve info on game #52's move #1

# REST Example (cont)

#### Get Game Info

#### curl -X GET localhost/games/52

#### HTTP/1.1 200 OK



#### • Data Structure

struct {
 int id;
 string user1;
 string user2;
 string href;

Simple data structure to send data from back-end to front-end

24-03-10

Martin Fowler: https://martinfowler.com/eaaCatalog/dataTransferObject.html <sup>19</sup>

# REST Example (cont)

#### Get Moves

curl -X GET localhost/games/52/moves

```
HTTP/1.1 200 OK
      2,
                "id":
                "user":
                          "Brian",
                "row":
                          1,
                "col":
                          1
           },
{
                "id":
                          6,
                          "AI3",
                "user":
                "row":
                          3,
                "col":
                          1
           }
24-03-10
```

#### • Make a move

curl -X POST -d {
 "user": "Brian",
 "row": 3,
 "col": 3
} localhost/games/52/moves

# **RESTful API Design**

- Design API around things and actions
   Structure URL for the hierarchical nature of the data
- Things (nouns)
  - Data you want to expose
- Actions (verbs)
  - C POST (or PUT)
  - R GET

– U POST (or PUT if you are updating the whole item at once, not just part).

– D DELETE

# RESTful API Design (cont)

- GET (and PUT) must be idempotent:
- POST is a catch all for doing anything.
- Properties of RESTful

Server returns self-descriptive

resources

Server maintains nothing about state of the connection; everything comes from HTTP headers, etc

Cache as much as possible to

reduce server load

- <...omitted more...>

# Summary

#### • HTTP

- Protocol for accessing resources via URL's

- HTTP Methods
  - GET, POST, DELETE, PUT, etc.
- Data in URL, Query String, Header, Body
- REST
  - Design URLs for Hierarchical data
  - REST properties