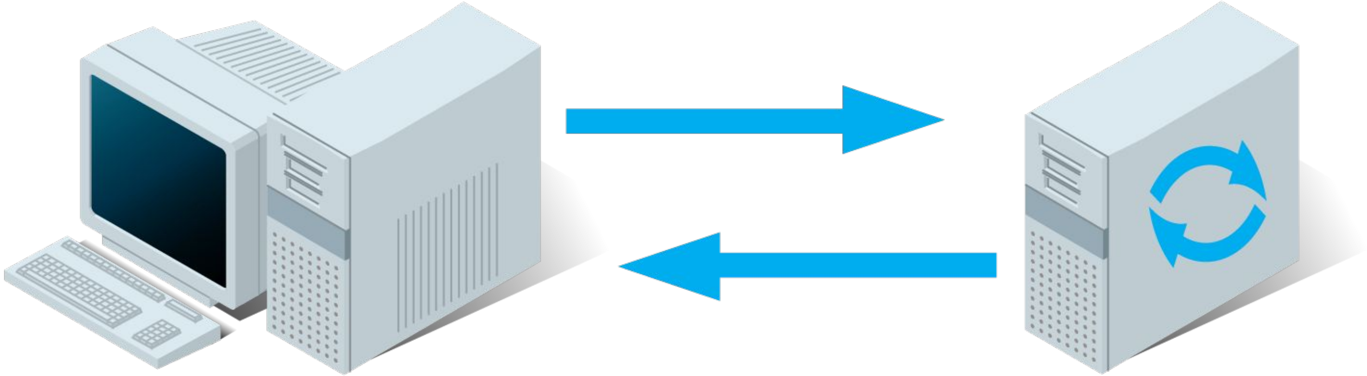
Polling (Long and Short) Sessions

Dr. Michael Whitney

** Heavily influenced by W3C PHP & TutorialsPoint PHP Sessions

Polling: Regular HTTP

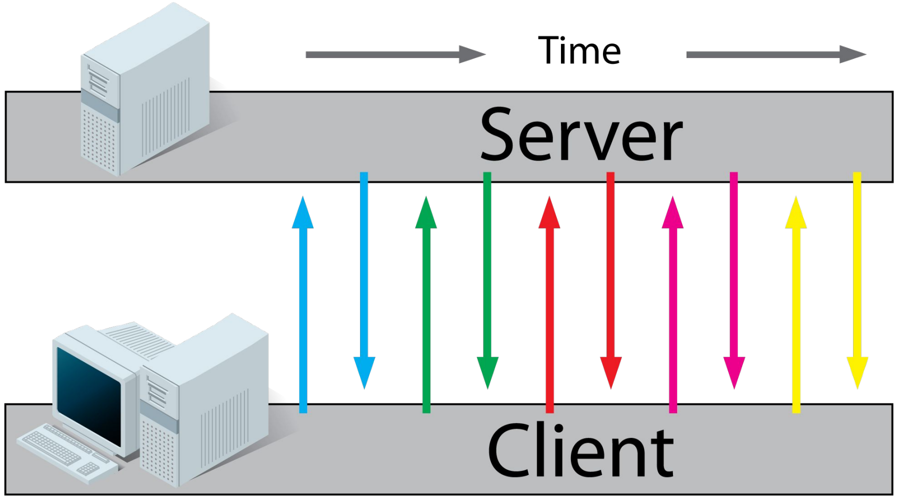
1. A client requests a webpage from a server.
2. The server calculates the response
3. The server sends the response to the client.



*** http://stackoverflow.com/questions/11077857/what-are-long-polling-websockets-server-sent-events-sse-and-comet

Short Polling

1. A client requests a webpage from a server using regular HTTP.
2. The requested webpage executes JavaScript which requests a file from the server at regular intervals (e.g. 0.5 seconds).
3. The server calculates each response and sends it back, just like normal HTTP traffic.



*** http://stackoverflow.com/questions/11077857/what-are-long-polling-websockets-server-sent-events-sse-and-comet

Short Polling: Good vs Bad?

Good

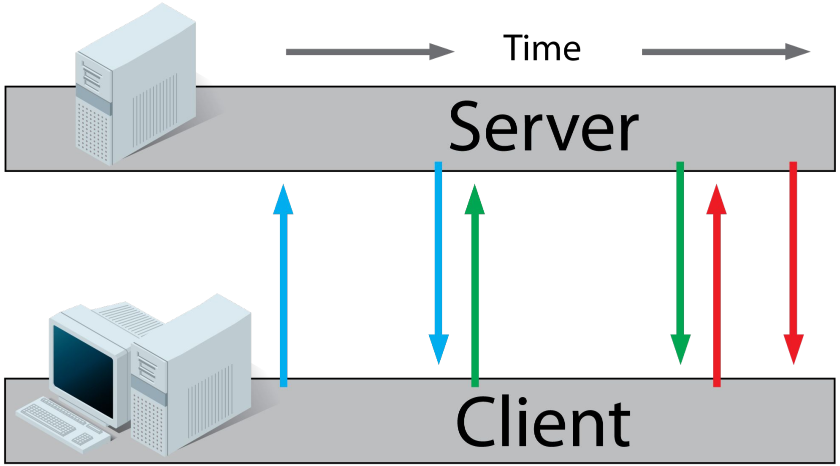
- Keep application up to date

Bad

- This cost a lot of requests on server

Long Polling (hanging GET)

1. A client requests a webpage from a server using regular HTTP.
2. The requested webpage executes JavaScript which requests a file from the server.
3. The server does not immediately respond with the requested information but waits until there's new information available.
4. When there's new information available, the server responds (push) with the new information.
5. The client receives the new information and immediately sends another request to the server, re-starting the process.



*** http://stackoverflow.com/questions/11077857/what-are-long-polling-websockets-server-sent-events-sse-and-comet

Long Polling: Good vs Bad?

Good

- Makes a near "real-time" application possible.

Bad

- Can overwhelm a server by requesting so many connections to remain open.

Let's Code Polling and Sessions

Server Chat

A Simple Page: getPage(page)

Create a page that loads info from another php

```
var pages = {}; // set up an array for pages
function getPage(page) {
        if(pages[page] === undefined) // if nothing is in the associative array for the specific page
                {
                        pages[page] = $.ajax({                    // set up connection with ajax
                        type: "GET",                              // Request data with query string in URL
                        url: page + ".php",                       // add .php to page
                        dataType: "json",                         // what expecting back from server
                });
                        return pages[page];
                 } else {                                         // if the data-url is already in the array
                        return pages[page]; } }
```

A Simple Page: $(doc).ready

Create a page that loads info from another php

```
  $(document).ready(function(){                     // wait for the page to load
        $("#menu li").click(function(){             // when the menu li is clicked
          var page = $(this).attr("data-url")       // get the url
          getPage(page).done(function(data){        // when promise is done
              $("#content").html(data.content);
          });
          getPage(page).fail(function(){            // when getPage fails
              $("#content").html("something went wrong");
          });
  });

});
```

A Simple Page: home.php

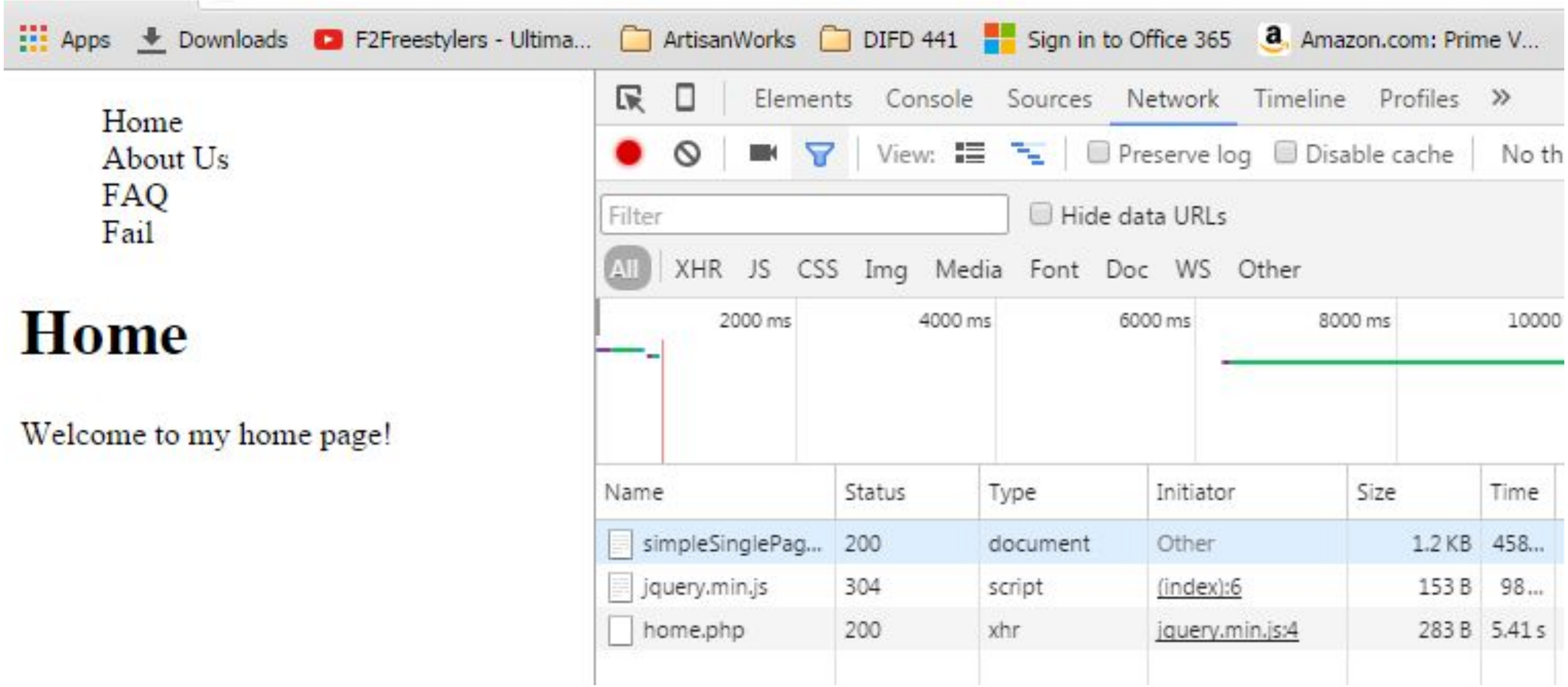
Create a page that loads info from another php

```
   <?php
          sleep(5);    // delay execution of the current script
                       // php Associative Array - http://www.w3schools.com/php/php_arrays.asp
          $content = array("content" => "<h1>Home</h1>Welcome to my home page!");

                  // used to send raw HTTP header - http://php.net/manual/en/function.header.php
          header("Content-Type: application/json");

                  // encode $content as a json - http://php.net/manual/en/function.json-encode.php
          echo json_encode($content);
                  // this returns {"content": <h1>Home</h1>Welcome to my home page!}
?>
```

Simple Single Page Site



Home
About Us
FAQ
Fail

Home

Welcome to my home page!

Name	Status	Type	Initiator	Size	Time
simpleSinglePag...	200	document	Other	1.2 KB	458...
jquery.min.js	304	script	(index):6	153 B	98...
home.php	200	xhr	jquery.min.js:4	283 B	5.41 s

Chatter Box: mySQL

Connecting to mySQL

```
$myhost = "127.0.0.1";
$mydbname = "c9";
$myuser = "csci441";
$mypass = "";

try
{                    // PHP Data Objects (PDO)
    $db = new PDO("mysql:host=$myhost;dbname=$mydbname", "$myuser", "$mypass");
    //Makes PDO throw exceptions for invalid SQL
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
```

Chatter Box: chat.php

```
<?php
    require("mysql.php");
```

- Same as include() except it will halt script if has error
- Files are included based on the file path given

Chatter Box: $_SERVER

```
if($_SERVER['REQUEST_METHOD'] == "POST")
```

- array containing information such as headers, paths, and script locations
- entries in this array are created by the web server

Chatter Box: $_SERVER

header("HTTP/1.1 400 Bad Request - Only GET and POST are supported.");

- Send a raw HTTP header

Chatter Box: die(); / exit();

exit ([string $status])

- Output a message and terminate the current script
- exit("unable to open file ($filename)");
- do not need parenthesis
- exit; exit(); exit(0);

Chatter Box: isset

isset — Determine if a variable is set and is not NULL

```
if(!isset($_POST["user"])) {
      header("HTTP/1.1 400 Bad Request - Must specify
'user' when POSTing.");
      die();
      }
```

Chatter Box: Prepared statements

```
$stmt = $db->prepare("INSERT INTO Chats
    (user, message) VALUES (:user, :message)");
```

- A prepared statement or a parameterized statement is used to execute the same statement repeatedly with high efficiency.

ChatterBox: PDOStatement::bindValue

```
$stmt->bindValue(":user", $_POST["user"]);
```

- Binds a value to a parameter

ChatterBox: fetch

```
while($row = $stmt->fetch(PDO::FETCH_ASSOC))
{
    $result[] = $row;
}
```

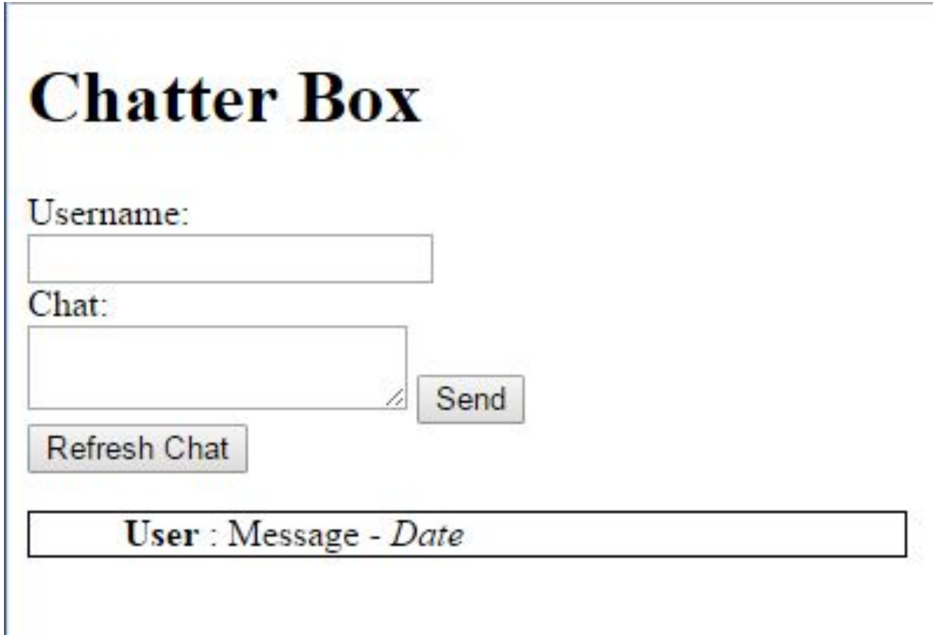
- Fetches the next row from a result set

ChatterBox: header / json_encode

```
header("Content-Type: application/json");
echo json_encode($result);
```

- Returns the JSON representation of a value

ChatterBox



Chatter Box

Username:

Chat:

User : Message - *Date*