

WEBSOCKETS

bidirectional full-duplex communication

Motivation

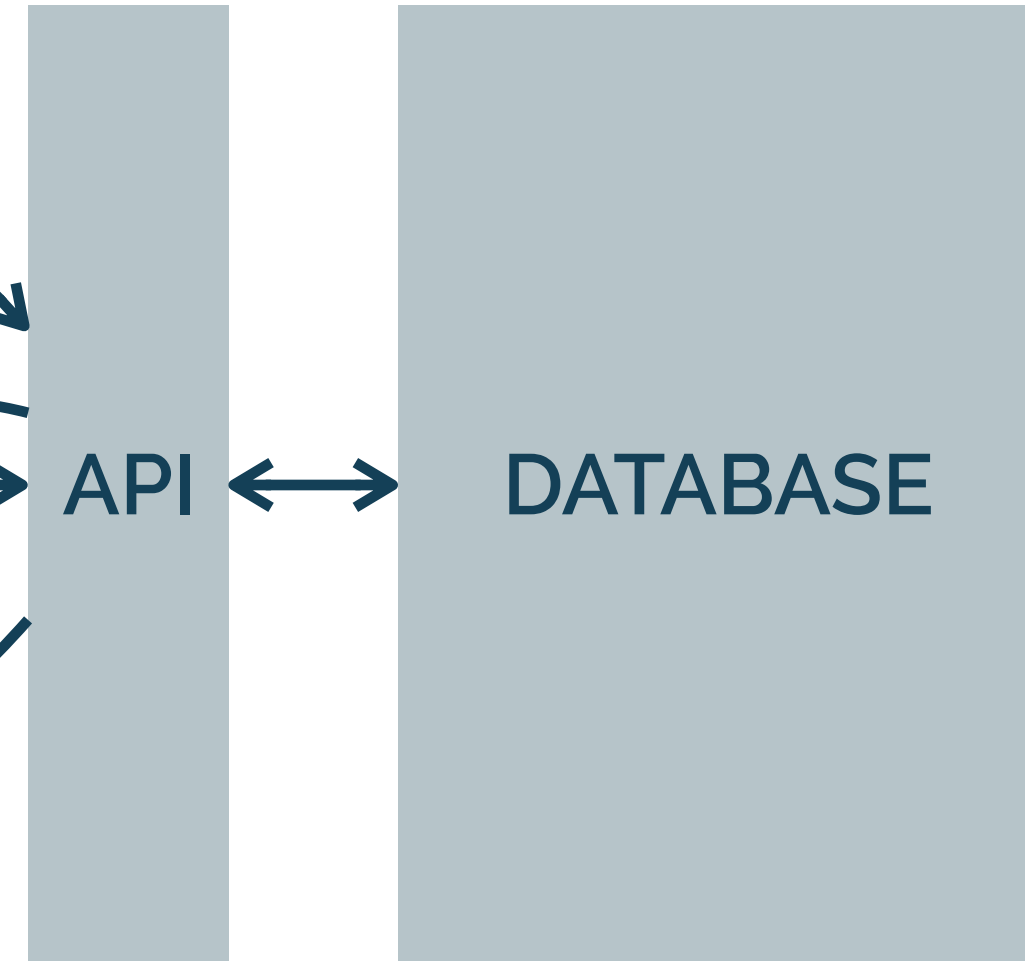
Client



HTTP POST

HTTP GET

Server



Client



HTTP POST

HTTP GET

API

DATABASE

Server

AJAX allows server requests to be made in the background

*...but if clients don't request,
servers can't respond!*

What types of applications
need servers to push data?

LOW-LATENCY, REAL-TIME

Multiplayer online games

Chat applications

Realtime updating social streams

How can servers push data?

POLLING

aka faking it

keep making requests to the server to see if there's any new information

performance problems: server has to process HUGE number of connections a second

COMET



a set of more advanced techniques

not an acronym!

hidden iframes (i.e., *forever frames*)

AJAX with long polling

unstandardized hacks, performance issues

en.wikipedia.org/wiki/Comet_%28programming%29

BROWSER PLUGINS

Adobe Flash, Java

raw push real-time data to clients through raw TCP
socket connections with servers

plug-ins not guaranteed to be installed, firewall issues

WebSockets

WEBSOCKETS

bidirectional full-duplex communication

WebSocket API introduced in HTML5 (2009)

persistent connection between the client and server

send data back and forth without HTTP overhead

en.wikipedia.org/wiki/WebSocket

Client Request

GET /chat HTTP/1.1

Host: server.example.com

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Sec-WebSocket-Protocol: chat, superchat

Sec-WebSocket-Version: 13

Origin: http://example.com

Server Response

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept:

HSmrc0sMlYUkAGmm5OPpG2HaGWk=

Sec-WebSocket-Protocol: chat

Open a WebSocket Connection

```
var connection = new WebSocket('ws://  
html5rocks.websocket.org/echo');
```


Attach Event Handlers

```
// When the connection is open, send some data to the server
connection.onopen = function () {
    connection.send('Lorenzo Llamas'); // Send the message 'Lorenzo
Llamas' to the server
};

// Log messages from the server
connection.onmessage = function (e) {
    console.log('Server: ' + e.data);
};

// Log errors
connection.onerror = function (error) { /*...*/ };

// Close connection
connection.onclose = function() { /*...*/ }
```

CROSS ORIGIN COMMUNICATION

supports communication between clients and servers on any domain

server decides which domains to allow connections from

PROBLEMS

Immediate security concerns (Opera 11, Safari 5)

Protocol was revamped and now supported by all modern browsers

Incompatibility with HTTP upgrade system and some proxy servers

even if client supports it, can't establish a connection!

WEBSOCKETS TODAY

use libraries that use earlier fallbacks whenever
WebSocket is not available

`socket.io`

Socket.IO

If at first you don't succeed...

WebSockets

Adobe Flash Socket

Ajax long polling

Ajax multipart streaming

Forever iFrame

JSONP Polling

“Socket.IO enables real-time bidirectional event-based communication.

It works on every platform, browser or device, focusing equally on reliability and speed.”

Demo

Client Code

```
<script>

    var socket = io();

    $('form').submit(function() {

        socket.emit('chat message', $('#m').val());

        $('#m').val('');

        return false;

    });

    socket.on('chat message', function(msg) {

        $('#messages').append($('- ').text(msg));

    });

</script>

```

socket.io

Server Code

```
app.get('/', function(req, res) {  
  res.sendFile('index.html');  
});  
  
io.on('connection', function(socket) {  
  socket.on('chat message', function(msg) {  
    io.emit('chat message', msg);  
  });  
});  
  
http.listen(3000, function() {  
  console.log('listening on *:3000');  
});
```

socket.io

NEXT CLASS: USERS

courses.engr.illinois.edu/cs498rk1/