

Contents

- Real-time web applications?
- Node.js
- Express.js
- Paradigm shift: events
- Socket.io
- Express.io
- Demo

Real-time web applications?

- Interaction with other users in near real-time on a web browser
- Games, chats, collaboration tools



Demo
YARR
RBL
A picture
programm

Paradigm shift: Events

- Instead of HTTP methods, like GET and POST, let's communicate with events.
- Two-way event bus
- From a client to the server
- From the server to the client
- Challenging with HTTP's client-server architecture

Event-driven programming

- Many names:
 - publish/subscribe
 - dispatcher/listener
 - emitter/handler
- Used extensively in GUIs and JavaScript programming in general.

Akseli Palén

- Hypermedia student at Tampere University of Technology
- Web entrepreneur and enthusiastic
- Beliefs that the world can be saved by improving human communication

Thank you!

Any questions? Ask away!
You can also contact me:
maksipalen@gmail.com

socket.io

- Event-driven two-way communication between clients and a server
- Started in mid 2011
- First uses polling then upgrades connection to WebSockets

express.io

- Express.js + Socket.io
- Even simpler interface
- Started in 2012



node.js

- A popular web application platform
- JavaScript running on Google's v8 engine
- First release in 2009
- Currently more popular than Django but bit less popular than Ruby on Rails

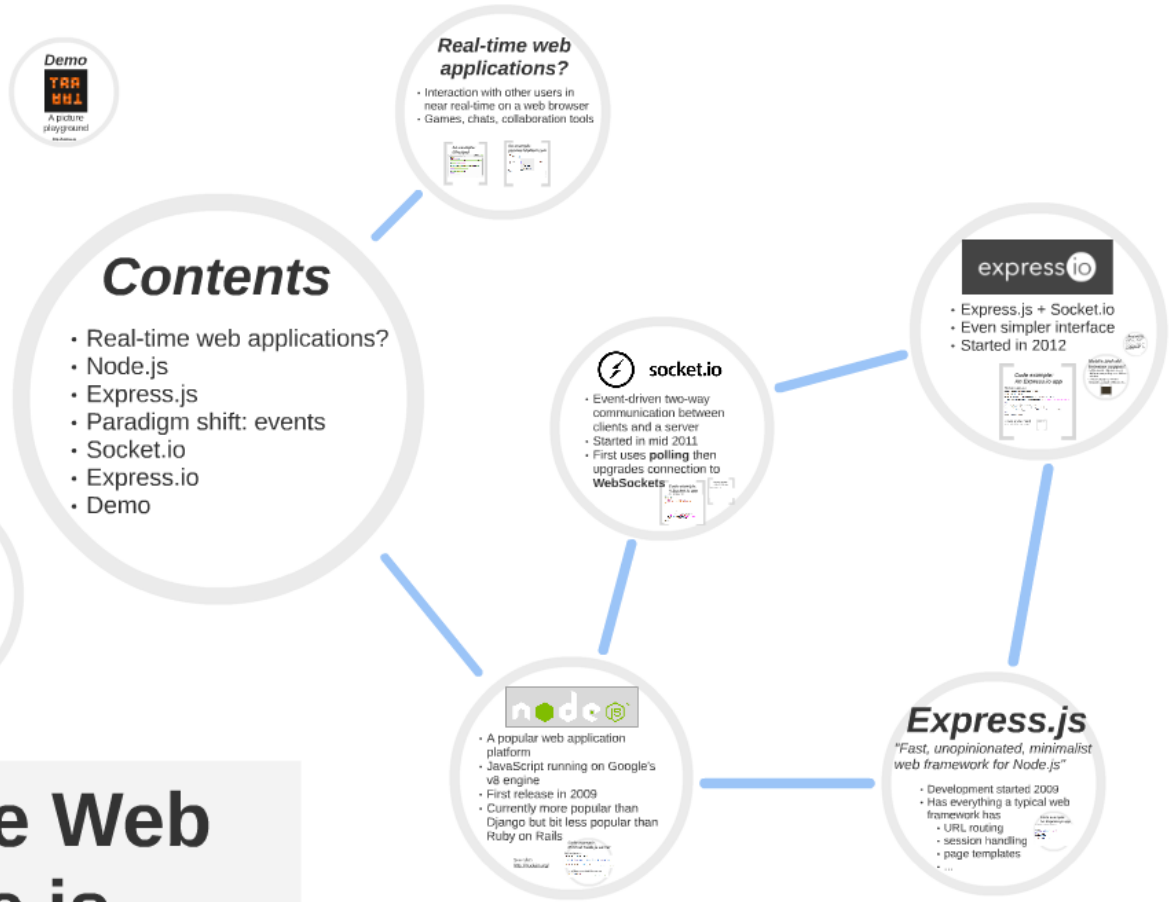
Express.js

"Fast, unopinionated, minimalist web framework for Node.js"

- Development started 2009
- Has everything a typical web framework has
 - URL routing
 - session handling
 - page templates

Webbisauna 2014

Express.io - Real-time Web Applications for Node.js



Webbisauna 2014

Express.io - Real-time Web Applications for Node.js

Akseli Palén

- Hypermedia student at Tampere University of Technology
- Web entrepreneur and enthusiastic
- Beliefs that the world can be saved by improving human communication

u!

away!

me:
com

Contents

- Real-time web applications?
- Node.js
- Express.js
- Paradigm shift: events
- Socket.io
- Express.io
- Demo

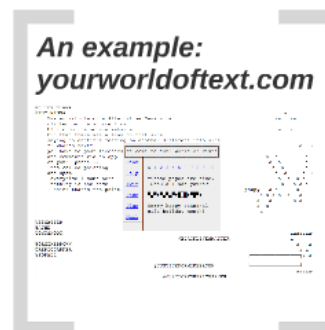
alén

ident at
rsity of

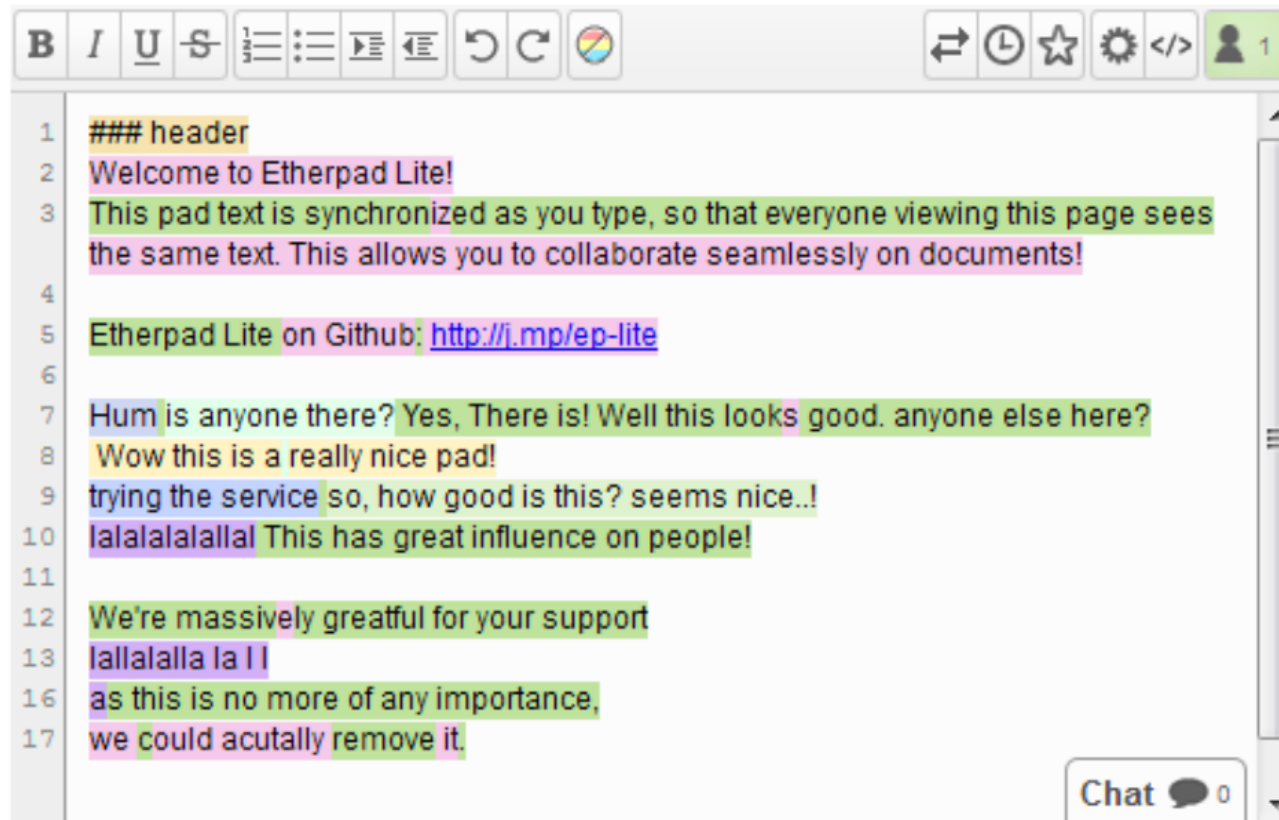
 Prezi
eur and

Real-time web applications?

- Interaction with other users in near real-time on a web browser
- Games, chats, collaboration tools



An example: Etherpad



The screenshot displays the Etherpad Lite interface. At the top, there is a toolbar with icons for bold (B), italic (I), underline (U), strikethrough (S), list creation, undo, redo, and a color picker. To the right of the toolbar are icons for share, clock, star, settings, code view, and a user profile icon labeled '1'. The main editing area contains a document with 17 lines of text, each with a line number on the left. The text includes a header, a welcome message, a paragraph about synchronization, a link to the Github page, and several lines of conversational text. A chat window is visible in the bottom right corner, labeled 'Chat' with a speech bubble icon and the number '0'.

1 **### header**

2 Welcome to Etherpad Lite!

3 This pad text is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents!

4

5 Etherpad Lite on Github: <http://j.mp/ep-lite>

6

7 Hum is anyone there? Yes, There is! Well this looks good. anyone else here?

8 Wow this is a really nice pad!

9 trying the service so, how good is this? seems nice..!

10 lalalalalal! This has great influence on people!

11

12 We're massively grateful for your support

13 lallalalla la l l

16 as this is no more of any importance,

17 we could acutally remove it.

Chat 0

An example: yourworldoftext.com

ES THE NIGGA
ITSY NIGGA

Saw an article on a Alternative News site
clicked on the source link.
OK I'm on the source website
but then there was a link on that site
saying to continue reading on another different info site
^^ What's next?
you have to post feedback
and comments via an app
on your Iphone ??
It's called phishing
and spam
everytime i come here
nothing is the same
I think that's the point.

Welcome to Your World of Text!

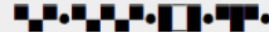
[Menu](#)

A [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

[Help](#)

^These pages are nice.
Should I add yours?

[Twit](#)



[Face](#)

[Plus](#)

Merry happy seasonal
gift holiday month!

[Mail](#)

VESMATTER
EATHE
DONTSHOOT

FORMIKEBROWN
ORERICGARNER
YFORALL

#BLACKLIVESMATTER

#JUSTICEFORMIKEBROWN

#JUSTICEFORERICGARNER

```

a
aa aa
a a
a
a a a
a a a
a a a
a a a a
a a a a
poopy a a a
a a a a a a
a a a a
a a
a
```

```

aaaaaaaa.
a
a a
a a
] a a
]
aaaaa.
```

A picture
playground

<http://johanaijz.nl>

near real-time on a web browser
• Games, chats, collaboration tools



Contents

- Real-time web applications?
- Node.js
- Express.js
- Paradigm shift: events
- Socket.io
- Express.io
- Demo

shift:

S

methods, like
's
events.

the server
to the client
HTTP's client-

Akseli Palén

- Hypermedia student at Tampere University of Technology
- Web entrepreneur and enthusiastic
- Beliefs that the world can be saved by improving human communication



- Even
com
clien
- Start
- First
upgr
Web



- A popular web application platform
- JavaScript running on Google's v8 engine
- First release in 2009
- Currently more popular than Django but bit less popular than Ruby on Rails

See also
<http://nodejs.org/>

Code example:
Minimal Node.js server

```
Server (app.js)
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8888, '127.0.0.1');
console.log('Server running at http://127.0.0.1:8888/');

And then just start the server
$ node app.js
Server running at http://127.0.0.1:8888/
```

Code example: Minimal Node.js server

Server (app.js)

```
var http = require('http');  
http.createServer(function handler(req, res) {  
  res.writeHead(200, {'Content-Type': 'text/plain'});  
  res.end('Hello World\n');  
}).listen(8888, '127.0.0.1');  
console.log('Server running at http://127.0.0.1:8888/');
```

And then just start the server

```
$ node app.js  
Server running at http://127.0.0.1:8888/
```

Express.js

"Fast, unopinionated, minimalist web framework for Node.js"

- Development started 2009
- Has everything a typical web framework has
 - URL routing
 - session handling
 - page templates
 - ...

Code example: An Express.js app

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hi!');
});

app.get('/user', function (req, res) {
  res.send('User');
});

app.listen(3000, function () {
  console.log('Server running at http://127.0.0.1:3000/');
});
```

Code example: An Express.js app

```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Got a GET request');
});

app.get('/user', funct...);
app.post('/user', funct...);
app.put('/user', funct...);
app.delete('/user', funct...);

app.listen(8888, function () {
  console.log('Server running at http://127.0.0.1:8888/');
});
```



A picture playground
<http://visuals.at>

- Interaction with other users in near real-time on a web browser
- Games, chats, collaboration tools



Contents

- Real-time web applications?
- Node.js
- Express.js
- Paradigm shift: events
- Socket.io
- Express.io
- Demo

Paradigm shift: events

HTTP methods, like GET, let's interact with **events**.
 Event bus
 Connect to the server
 Server to the client
 With HTTP's client-server structure



- Event-driven communication between clients and servers
- Started in 2005
- First used for upgrading WebSockets

Akseli Palén

- Hypermedia student at Tampere University of Technology
- Web entrepreneur and enthusiastic
- Beliefs that the world can be saved by improving human communication



Paradigm shift: Events

- Instead of HTTP methods, like GET and POST, let's communicate with **events**.
- Two-way event bus
 - From a client to the server
 - From the server to the client
- Challenging with HTTP's client-server architecture

Event-driven programming

- Many names:
 - publish/subscribe
 - dispatcher/listener
 - emitter/handler
- Used extensively in GUIs and JavaScript programming in general.

Code example: Event bus

Bind an event to a handler function

```
bus.on('chatMessage', function handle(msg) {
  console.log(msg);
});
```

Emit an event with some data

```
bus.emit('chatMessage', 'hey guys!');
```

Code example: Event bus

Bind an event to a handler function

```
bus.on('chatMessage', function handler(msg) {  
  console.log(msg);  
});
```

Emit an event with some data

```
bus.emit('chatMessage', 'Hey guys!');
```




A picture playground

<http://tamusa.at>

- Interaction with other users in near real-time on a web browser
- Games, chats, collaboration tools



Contents

- Real-time web applications?
- Node.js
- Express.js
- Paradigm shift: events
- Socket.io
- Express.io
- Demo

Paradigm shift: events

HTTP methods, like GET, let's us interact with **events**.
 Event-driven architecture
 Event bus
 Event sent to the server
 Event server to the client
 Event with HTTP's client-server architecture



- Event-driven communication between clients and servers
- Started in the late 1990s
- First used in the early 2000s
WebSockets

Akseli Palén

- Hypermedia student at Tampere University of Technology
- Web entrepreneur and enthusiastic
- Beliefs that the world can be saved by improving human communication



ets

Code example: A Socket.io app

Client (index.html)

```
...
<ul id="chat"></ul>
<form id="form">
  <input id="input" type="text" />
  <input type="submit" value="Send" />
</form>

<script>
  var socket = io.connect();
  socket.on('chatMessage', function (eventData) {
    $('#chat').append('<li>' + eventData.msg + '</li>');
  });
  $('#form').submit(function (submitEvent) {
    var message = $('#input').val();
    socket.emit('chatMessage', { msg: message });
    submitEvent.preventDefault();
  });
</script>
...
```

Server

```
...
var socket = io.socket();
var io = io.socket();
// Broadcast
io.on('connect', function () {
  socket.emit('chatMessage', { msg: 'Hello' });
});
...

```

Code example: A Socket.io app

Server (app.js)

```
...  
var socketio = require('socket.io');  
var io = socketio.listen(app);  
// Broadcast received chat messages to all connected clients  
io.on('connection', function (socket) {  
  socket.on('chatMessage', function (eventData) {  
    io.sockets.emit('chatMessage', eventData.msg);  
  });  
});  
...
```

expressio

- Express.js + Socket.io
- Even simpler interface
- Started in 2012

Code example: An Express.io app

```
Server (app.js)
app = require('express.io')();
app.http().io();
app.io.route('chatMessage', function (req) {
  app.io.broadcast('chatMessage', { msg: req.data.msg });
});
// Serve static client code
app.get('/', function (req, res) {
  res.sendfile(__dirname + '/client/index.html');
});
app.listen(8888);
```

Client (index.html)
Same as Socket.io client example



Scalability

- Express is process + Node.js friendly
- The code is simple and easy to maintain
- The code is easy to extend in order to support multiple instances
- The code is easy to extend in order to support multiple instances
- The code is easy to extend in order to support multiple instances



Mobile and old browser support

- Wide support due Socket.io's ability to use polling as a fallback solution.
- Almost all popular mobile browsers support WebSockets



Code example: An Express.io app

Server (app.js)

```
app = require('express.io') ();
app.http().io();
app.io.route('chatMessage', function (req) {
  app.io.broadcast('chatMessage', { msg: req.data.msg });
});
// Serve static client code
app.get('/', function (req, res) {
  res.sendfile(__dirname + '/client/index.html');
});
app.listen(8888);
```

Client (index.html)

Same as Socket.io client example



Code example: A Socket.io app

Client (index.html)

```
...
<ul id="chat"></ul>
<form id="form">
  <input id="input" type="text" />
  <input type="submit" value="Send" />
</form>

<script>
  var socket = io.connect();
  socket.on('chatMessage', function (eventData) {
    $('#chat').append('<li>' + eventData.msg + '</li>');
  });
  $('#form').submit(function (submitEvent) {
    var message = $('#input').val();
    socket.emit('chatMessage', { msg: message });
    submitEvent.preventDefault();
  });
</script>
...
```


Web Sockets - CR

Global **82.72%** + 1.8% = **84.52%**

unprefixed: **82.72%** + 1.68% = **84.4%**

Bidirectional communication technology for web apps

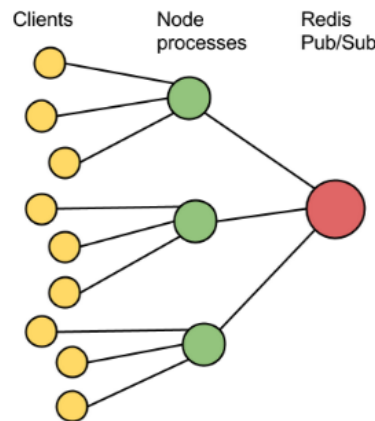
Current aligned Usage relative Show all

IE	Firefox	Chrome	Safari	Opera	iOS Safari*	Opera Mini*	Android Browser*	Chrome for Android
		31						
		33						
		35					4.1	
8		36	5.1				4.3	
9	31	37	7		7.1		4.4	
10	32	38	7.1		8		4.4.4	
11	33	39	8	26	8.1	8	37	39
TP	34	40		27				
	35	41		28				
	36	42						

Source: <http://caniuse.com/#feat=websockets>

Scalability

- Express.io process = Node.js process
- To scale things up, many Node.js processes are run on single or multiple machines
- How to share an event bus with many processes: **socket.io-adapter**
- For example **socket.io-redis** makes use of **Publish/Subscribe** feature of **Redis** database



Contents

- Real-time web applications?
- Node.js
- Express.js
- Paradigm shift: events
- Socket.io
- Express.io
- Demo

alén

ident at
rsity of

 Prezi
eur and

Demo



A picture
playground

<http://taataa.at>



Thank you!

Any questions? Ask away!

You can also contact me:
akseli.palen@gmail.com