Advanced Node.js

Claudia Hauff
cse1500-ewi@tudelft.nl

TU Delft

Learning goals

- **Organize** Node.js code into modules
- **Understand and employ** the concept of middleware
- **Employ** routing
- **Employ** templating

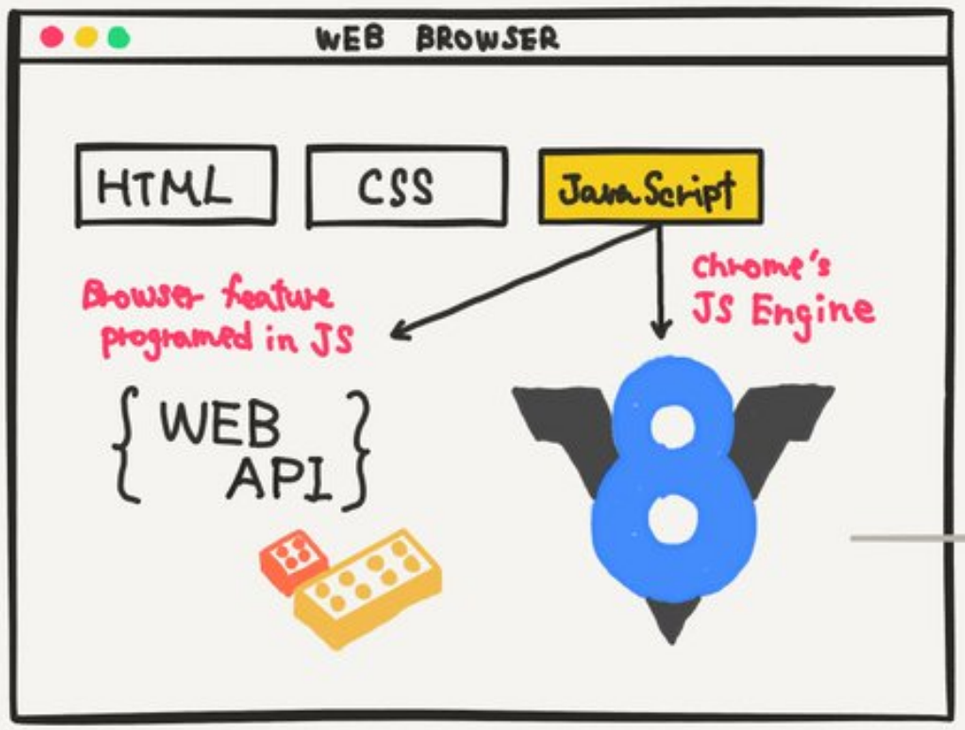
Organization and reusability of Node.js code

drawing by Mariko Kosaka (@kosamari)

@kosamari
March 3, 2017

What is Node.JS ??

What's browser JS ??

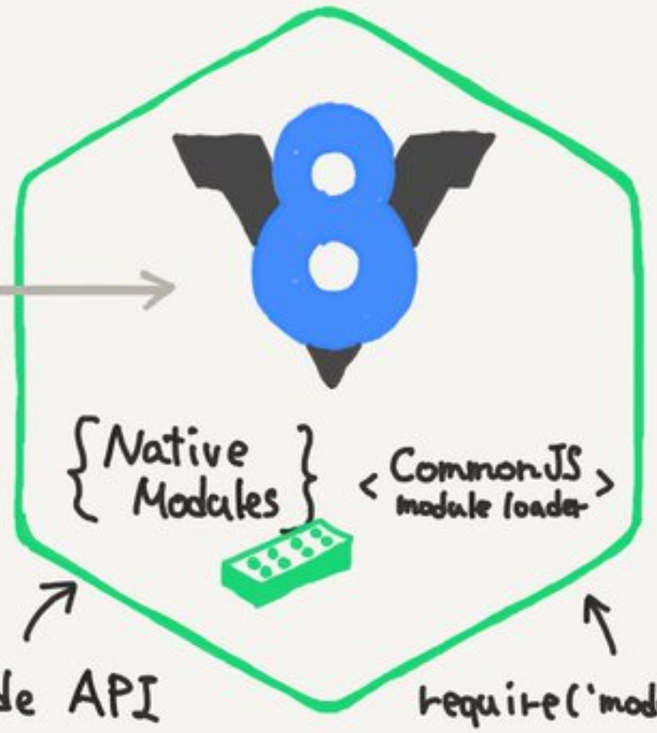


Web Platform
- Window
- Fetch
- Web Audio etc...

Language features
- ES6
- Promise
- Typed Array etc...

Web API & JavaScript Language are often described as the JavaScript (in intro books)

Node is a JavaScript ENVIRONMENT with Special API (like http) and default module loader.



Node API
- http
- fs
- process etc...

Node runs on EVERYWHERE.

So far…

- All server-side code maintained **within a single file**
  - Often makes sense for client-side JS (**minification**)
- Possible for small projects
- Larger projects suffer in this setting
  - **Debugging** is cumbersome
  - **Team-work** is cumbersome
  - **Programming** is cumbersome

```
/* ESLint global variables information */
/* global Setup, Status, Messages, englishDict*/

/* basic constructor of game state */
function GameState(visibleWordBoard, sb, socket){

    this.playerType = null;
    this.MAX_ALLOWED = Setup.MAX_ALLOWED_GUESSES;
    this.wrongGuesses = 0;
    this.visibleWordArray = null;
    this.alphabet = new Alphabet();
    this.alphabet.initialize();
    this.visibleWordBoard = visibleWordBoard;
    this.targetWord = null;
    this.statusBar = sb;

    this.initializeVisibleWordArray = function(){
        this.visibleWordArray = new Array(this.targetWord.length);
        this.visibleWordArray.fill(Setup.HIDDEN_CHAR);
    };

    this.getPlayerType = function () {
        return this.playerType;
    };

    this.setPlayerType = function (p) {
        console.assert(typeof p == "string", "%s: Expecting a string, got a %s", arguments.callee.name, typeof p);
        this.playerType = p;
    };

    this.setTargetWord = function (w) {
        console.assert(typeof w == "string", "%s: Expecting a string, got a %s", arguments.callee.name, typeof w);
        this.targetWord = w;
    };

    this.getVisibleWordArray = function(){
        return this.visibleWordArray;
    };

    this.incrWrongGuess = function(){
        this.wrongGuesses++;

        if(this.whoWon() == null){
            //kill a balloon
            let id = "b"+this.wrongGuesses;
            document.getElementById(id).className += " balloonGone";
            setTimeout(function () {
                new Audio("../data/pop.wav").play();
            }, 500);
        }
    };

```

interaction.js
first ~50 lines (266 in total)

minification

```
/* ESLint global variables information */
/* global Setup, Status, Messages, englishDict*/

/* basic constructor of game state */
function GameState(visibleWordBoard, sb, socket){

    this.playerType = null;
    this.MAX_ALLOWED = Setup.MAX_ALLOWED_GUESSES;
    this.wrongGuesses = 0;
    this.visibleWordArray = null;
    this.alphabet = new Alphabet();
    this.alphabet.initialize();
    this.visibleWordBoard = visibleWordBoard;
    this.targetWord = null;
    this.statusBar = sb;

    this.initializeVisibleWordArray = function(){
        this.visibleWordArray = new Array(this.targetW
        this.visibleWordArray.fill(Setup.HIDDEN_CHAR);
    };

    this.getPlayerType = function () {
        return this.playerType;
    };

    this.setPlayerType = function (p) {
        console.assert(typeof p == "string", "%s: Expe
        this.playerType = p;
    };

    this.setTargetWord = function (w) {
        console.assert(typeof w == "string", "%s: Expe
        this.targetWord = w;
    };

    this.getVisibleWordArray = function(){
        return this.visibleWordArray;
    };

    this.incrWrongGuess = function(){
        this.wrongGuesses++;

        if(this.whoWon() == null){
            //kill a balloon
            let id = "b"+this.wrongGuesses;
            document.getElementById(id).className += "
            setTimeout(function () {
                new Audio("../data/pop.wav").play();
            }, 500);
        }
    };

```

```
function GameState(t,e,a){this.playerType=null,this.MAX_ALLOWED=Setup.MAX_ALLOWED_GUESSES,
    this.wrongGuesses=0,this.visibleWordArray=null,this.alphabet=new Alphabet,this.
    alphabet.initialize(),this.visibleWordBoard=t,this.targetWord=null,this.statusBar=e,
    this.initializeVisibleWordArray=function(){this.visibleWordArray=new Array(this.
    targetWord.length),this.visibleWordArray.fill(Setup.HIDDEN_CHAR)},this.getPlayerType=
    function(){return this.playerType},this.setPlayerType=function(t){console.assert("
    string"==typeof t,"%s: Expecting a string, got a %s",arguments.callee.name,typeof t),
    this.playerType=t},this.setTargetWord=function(t){console.assert("string"==typeof t,"
    %s: Expecting a string, got a %s",arguments.callee.name,typeof t),this.targetWord=t},
    this.getVisibleWordArray=function(){return this.visibleWordArray},this.incrWrongGuess=
    function(){if(this.wrongGuesses++,null==this.whoWon()){let t="b"+this.wrongGuesses;
    document.getElementById(t).className+=" balloonGone",setTimeout(function(){new Audio("
    ../data/pop.wav").play()},500)}},this.whoWon=function(){return this.wrongGuesses>Setup
    .MAX_ALLOWED_GUESSES?"A":this.visibleWordArray.indexOf("#")<0?"B":null},this.
    revealLetters=function(t,e){console.assert("string"==typeof t,"%s: Expecting a
    string, got a %s",arguments.callee.name,typeof t),console.assert(e instanceof Array,"
    %s: Expecting an array",arguments.callee.name);for(let a=0;a<e.length;a++)this.
    visibleWordArray[e[a]]=t},this.revealAll=function(){this.visibleWordBoard.setWord(this
    .targetWord)},this.updateGame=function(t){console.assert("string"==typeof t,"%s:
    Expecting a string, got a %s",arguments.callee.name,typeof t);var s=this.alphabet.
    getLetterInWordIndices(t,this.targetWord);0==s.length?this.incrWrongGuess():this.
    revealLetters(t,s),this.alphabet.makeLetterUnAvailable(t),this.visibleWordBoard.
    setWord(this.visibleWordArray);var i=Messages.O_MAKE_A_GUESS;i.data=t,a.send(JSON.
    stringify(i));let r=this.whoWon();if(null!=r){this.revealAll();var n=document.
    querySelectorAll(".alphabet");let t;if(Array.from(n).forEach(function(t){var e=t.clo
    neNode(!0);t.parentNode.replaceChild(e,t)}),t=r==this.playerType?Status.gameWon:Status
    .gameLost,t+=Status.playAgain,e.setStatus(t),"B"==this.playerType){let t=Messages.
    O_GAME_WON_BY;t.data=r,a.send(JSON.stringify(t))}a.close()}}}function AlphabetBoard(t)
    {this.initialize=function(){var e=document.querySelectorAll(".alphabet");Array.from(e)
    .forEach(function(e){e.addEventListener("click",function a(s){var i=s.target.id;new
    Audio("../data/click.wav").play(),t.updateGame(i),e.removeEventListener("click",a,!1)}
    )})}}function disableAlphabetButtons(){var t=document.getElementById("alphabet").get
    ElementsByTagName("div");for(i=0;i<t.length;i++)t.item(i).className+="
    alphabetDisabled"}!function(){var t=new WebSocket(Setup.WEB_SOCKET_URL),e=new
    VisibleWordBoard,a=new StatusBar;createBalloons();var s=new GameState(e,a,t),i=new
    AlphabetBoard(s);t.onmessage=function(r){let n=JSON.parse(r.data);if(n.type==Messages.
    T_PLAYER_TYPE)if(s.setPlayerType(n.data),"A"==s.getPlayerType()){
    disableAlphabetButtons(),a.setStatus(Status.player1Intro);let i=-1,r=Status.prompt,n=
    null;for(;i<0;)null==(n=prompt(r))?r=Status.prompt:(n=n.toUpperCase()).length<Setup.
    MIN_WORD_LENGTH||n.length>Setup.MAX_WORD_LENGTH?r=Status.promptAgainLength:0==/^[a-zA-
    Z]+$/.test(n)?r=Status.promptChars:0==englishDict.hasOwnProperty(n.toLocaleLowerCase()
    )?r=Status.promptEnglish:i=1;a.setStatus(Status.chosen+n),s.setTargetWord(n),s.
    initializeVisibleWordArray(),e.setWord(s.getVisibleWordArray());let o=Messages.
    O_TARGET_WORD;o.data=n,t.send(JSON.stringify(o))}else a.setStatus(Status.
    player2IntroNoTargetYet);n.type==Messages.T_TARGET_WORD&&"B"==s.getPlayerType()&&(s.
    setTargetWord(n.data),a.setStatus(Status.player2Intro),s.initializeVisibleWordArray(),
    i.initialize(),e.setWord(s.getVisibleWordArray())),n.type==Messages.T_MAKE_A_GUESS&&"A
    "==s.getPlayerType()&&(a.setStatus(Status.guessed+n.data),s.updateGame(n.data))},t.
    onopen=function(){t.send("{}")},t.onclose=function(){null==s.whoWon()&&a.setStatus(
    Status.aborted)},t.onerror=function(){}}();
```

all of `interaction.js`; 50% bytes saved

interaction.js
first ~50 lines (266 in total)

```
function GameState(t,e,a){
    this.wrongGuesses=0,th
    alphabet.initialize(),
    this.initializeVisible
    targetWord.length),thi
    function(){return this
    string"==typeof t,"%s:
    this.playerType=t},thi
    %s: Expecting a string
    this.getVisibleWordArr
    function(){if(this.wro
    document.getElementById
    ../data/pop.wav").play
    .MAX_ALLOWED_GUESSES?"
    revealLetters=function
    string, got a %s",argu
    %s: Expecting an array
    visibleWordArray[e[a]]
    .targetWord)},this.upd
    Expecting a string, go
    getLetterInWordIndices
    revealLetters(t,s),thi
    setWord(this.visibleWo
    stringify(i));let r=th
    querySelectorAll(".alp
    neNode(!0);t.parentNod
    .gameLost,t+=Status.pl
    O_GAME_WON_BY;t.data=r
    {this.initialize=funct
    .forEach(function(e){e
    Audio("../data/click.w
    )})}}function disableA
    ElementsByTagName("div
    alphabetDisabled"}!fun
    VisibleWordBoard,a=new
    AlphabetBoard(s);t.onm
    T_PLAYER_TYPE)if(s.set
    disableAlphabetButtons
    null;for(;i<0;)null==(
    MIN_WORD_LENGTH||n.len
    Z]+$/.test(n)?r=Status
    )?r=Status.promptEngli
    initializeVisibleWordA
    O_TARGET_WORD;o.data=n
    player2IntroNoTargetYe
    setTargetWord(n.data),
    i.initialize(),e.setWo
    "==s.getPlayerType()&&
    onopen=function(){t.se
    Status.aborted)},t.one
```

```
function GameState(t, e, a) {
 this.playerType = null, this.MAX_ALLOWED = Setup.MAX_ALLOWED_GUESSES, this.wrongGuesses =
    0, this.visibleWordArray = null, this.alphabet = new Alphabet, this.alphabet.
   initialize(), this.visibleWordBoard = t, this.targetWord = null, this.statusBar = e,
   this.initializeVisibleWordArray = function() {
  this.visibleWordArray = new Array(this.targetWord.length), this.visibleWordArray.fill(
    Setup.HIDDEN_CHAR)
 }, this.getPlayerType = function() {
  return this.playerType
 }, this.setPlayerType = function(t) {
  console.assert("string" == typeof t, "%s: Expecting a string, got a %s", arguments.
    callee.name, typeof t), this.playerType = t
 }, this.setTargetWord = function(t) {
  console.assert("string" == typeof t, "%s: Expecting a string, got a %s", arguments.
    callee.name, typeof t), this.targetWord = t
 }, this.getVisibleWordArray = function() {
  return this.visibleWordArray
 }, this.incrWrongGuess = function() {
  if (this.wrongGuesses++, null == this.whoWon()) {
   let t = "b" + this.wrongGuesses;
   document.getElementById(t).className += " balloonGone", setTimeout(function() {
    new Audio("../data/pop.wav").play()
   }, 500)
  }
 }, this.whoWon = function() {
  return this.wrongGuesses > Setup.MAX_ALLOWED_GUESSES ? "A" : this.visibleWordArray.
    indexOf("#") < 0 ? "B" : null
 }, this.revealLetters = function(t, e) {
  console.assert("string" == typeof t, "%s: Expecting a string, got a %s", arguments.
    callee.name, typeof t), console.assert(e instanceof Array, "%s: Expecting an array",
    arguments.callee.name);
  for (let a = 0; a < e.length; a++) this.visibleWordArray[e[a]] = t
 }, this.revealAll = function() {
  this.visibleWordBoard.setWord(this.targetWord)
 }, this.updateGame = function(t) {
  console.assert("string" == typeof t, "%s: Expecting a string, got a %s", arguments.
    callee.name, typeof t);
  var s = this.alphabet.getLetterInWordIndices(t, this.targetWord);
  0 == s.length ? this.incrWrongGuess() : this.revealLetters(t, s), this.alphabet.
    makeLetterUnAvailable(t), this.visibleWordBoard.setWord(this.visibleWordArray);
  var i = Messages.O_MAKE_A_GUESS;
  i.data = t, a.send(JSON.stringify(i));
  let r = this.whoWon();
  if (null != r) {
   this.revealAll();
   var n = document.querySelectorAll(".alphabet");
   let t;
```

"unminimizing" beautification

```
/* ESLint global variables information */
/* global Setup, Status, Messages, englishD

/* basic constructor of game state */
function GameState(visibleWordBoard, sb, so

    this.playerType = null;
    this.MAX_ALLOWED = Setup.MAX_ALLOWED_GU
    this.wrongGuesses = 0;
    this.visibleWordArray = null;
    this.alphabet = new Alphabet();
    this.alphabet.initialize();
    this.visibleWordBoard = visibleWordBoar
    this.targetWord = null;
    this.statusBar = sb;

    this.initializeVisibleWordArray = funct
        this.visibleWordArray = new Array(t
        this.visibleWordArray.fill(Setup.HI
    };

    this.getPlayerType = function () {
        return this.playerType;
    };

    this.setPlayerType = function (p) {
        console.assert(typeof p == "string"
        this.playerType = p;
    };

    this.setTargetWord = function (w) {
        console.assert(typeof w == "string"
        this.targetWord = w;
    };

    this.getVisibleWordArray = function(){
        return this.visibleWordArray;
    };

    this.incrWrongGuess = function(){
        this.wrongGuesses++;

        if(this.whoWon() == null){
            //kill a balloon
            let id = "b"+this.wrongGuesses;
            document.getElementById(id).cla
            setTimeout(function () {
                new Audio("../data/pop.wav"
            }, 500);
        }
    };
```

```
var _0x13fe=['name','setTargetWord','assert','getVisibleWordArra
  getElementById','\x20balloonGone','../data/pop.wav','play','wh
  revealLetters','%s:\x20Expecting\x20an\x20array','length','rev
  getLetterInWordIndices','makeLetterUnAvailable','O_MAKE_A_GUESS','send','stringify','
  querySelectorAll','.alphabet','from','forEach','cloneNode','parentNode','replaceChild','
  gameLost','setStatus','O_GAME_WON_BY','close','addEventListener','click','target','
  ../data/click.wav','div','item','\x20alphabetDisabled','WEB_SOCKET_URL','onmessage','
  parse','data','T_PLAYER_TYPE','getPlayerType','player1Intro','prompt','toUpperCase','
  MIN_WORD_LENGTH','MAX_WORD_LENGTH','promptAgainLength','test','promptChars','
  hasOwnProperty','toLocaleLowerCase','promptEnglish','O_TARGET_WORD','type','
  T_TARGET_WORD','T_MAKE_A_GUESS','updateGame','aborted','MAX_ALLOWED','
  MAX_ALLOWED_GUESSES','wrongGuesses','visibleWordArray','alphabet','initialize','
  visibleWordBoard','targetWord','statusBar','initializeVisibleWordArray','playerType','
  setPlayerType','string','%s:\x20Expecting\x20a\x20string,\x20got\x20a\x20%s','callee'];(
  function(_0xb67384,_0x27295a){var _0x156d7c=function(_0x52a1f8){while(--_0x52a1f8){
  _0xb67384['push'](_0xb67384['shift']());}};_0x156d7c(++_0x27295a);}(_0x13fe,0x172));var
  _0x14ff=function(_0x594e55,_0x3e24bd){_0x594e55=_0x594e55-0x0;var _0x5cea58=_0x13fe[
  _0x594e55];return _0x5cea58;};function GameState(_0x583931,_0x5cdcd4,_0x2f0ad4){this['
  playerType']=null;this[_0x14ff('0x0')]=Setup[_0x14ff('0x1')];this[_0x14ff('0x2')]=0x0;
  this[_0x14ff('0x3')]=null;this['alphabet']=new Alphabet();this[_0x14ff('0x4')][_0x14ff('
  0x5')]();this[_0x14ff('0x6')]=_0x583931;this[_0x14ff('0x7')]=null;this[_0x14ff('0x8')]=
  _0x5cdcd4;this[_0x14ff('0x9')]=function(){this[_0x14ff('0x3')]=new Array(this[_0x14ff('
  0x7')]['length']);this[_0x14ff('0x3')]['fill'](Setup['HIDDEN_CHAR']);};this['
  getPlayerType']=function(){return this[_0x14ff('0xa')];};this[_0x14ff('0xb')]=function(
  _0x442164){console['assert'](typeof _0x442164==_0x14ff('0xc'),_0x14ff('0xd'),arguments[
  _0x14ff('0xe')][_0x14ff('0xf')],typeof _0x442164);this['playerType']=_0x442164;};this[
  _0x14ff('0x10')]=function(_0x4694eb){console[_0x14ff('0x11')](typeof _0x4694eb==_0x14ff(
  '0xc'),'%s:\x20Expecting\x20a\x20string,\x20got\x20a\x20%s',arguments[_0x14ff('0xe')][
  _0x14ff('0xf')],typeof _0x4694eb);this[_0x14ff('0x7')]=_0x4694eb;};this[_0x14ff('0x12')]
  =function(){return this[_0x14ff('0x3')];};this[_0x14ff('0x13')]=function(){this[_0x14ff(
  '0x2')]++;if(this['whoWon']()==null){let _0xb7d465='b'+this[_0x14ff('0x2')];document[
  _0x14ff('0x14')](_0xb7d465)['className']+=_0x14ff('0x15');setTimeout(function(){new
  Audio(_0x14ff('0x16'))[_0x14ff('0x17')]();},0x1f4);}};this[_0x14ff('0x18')]=function(){
  if(this[_0x14ff('0x2')]>Setup[_0x14ff('0x1')]){return'A';}if(this[_0x14ff('0x3')][
  _0x14ff('0x19')]('#')<0x0){return'B';}return null;};this[_0x14ff('0x1a')]=function(
  _0x11dabc,_0x591157){console[_0x14ff('0x11')](typeof _0x11dabc=='string',_0x14ff('0xd'),
  arguments[_0x14ff('0xe')][_0x14ff('0xf')],typeof _0x11dabc);console['assert'](_0x591157
  instanceof Array,_0x14ff('0x1b'),arguments[_0x14ff('0xe')][_0x14ff('0xf')]);for(let
  _0x2694ba=0x0;_0x2694ba<_0x591157[_0x14ff('0x1c')];_0x2694ba++){this[_0x14ff('0x3')][
  _0x591157[_0x2694ba]]=_0x11dabc;}};this[_0x14ff('0x1d')]=function(){this[_0x14ff('0x6')]
  [_0x14ff('0x1e')](this[_0x14ff('0x7')]);};this['updateGame']=function(_0x5c88a1){console
  [_0x14ff('0x11')](typeof _0x5c88a1==_0x14ff('0xc'),_0x14ff('0xd'),arguments['callee']['
  _0x14ff('0xf')],typeof _0x5c88a1);var _0x3a7220=this[_0x14ff('0x4')][_0x14ff('0x1f')](
  _0x5c88a1,this[_0x14ff('0x7')]);if(_0x3a7220[_0x14ff('0x1c')]==0x0){this[_0x14ff('0x13')
  ]();}else{this[_0x14ff('0x1a')](_0x5c88a1,_0x3a7220);}this[_0x14ff('0x4')][_0x14ff('0x20
  ')](_0x5c88a1);this[_0x14ff('0x6')][_0x14ff('0x1e')](this[_0x14ff('0x3')]);var _0x438e50
  =Messages[_0x14ff('0x21')];_0x438e50['data']=_0x5c88a1;_0x2f0ad4[_0x14ff('0x22')](JSON[
  _0x14ff('0x23')](_0x438e50));let _0x3a6e16=this[_0x14ff('0x18')]();if(_0x3a6e16!=null){
  this[_0x14ff('0x1d')]();var _0x5cdfe1=document[_0x14ff('0x24')](_0x14ff('0x25'));Array[
  _0x14ff('0x26')](_0x5cdfe1)[_0x14ff('0x27')](function(_0x35f612){var _0x169fe3=_0x35f612
  [_0x14ff('0x28')](!![]);_0x35f612[_0x14ff('0x29')][_0x14ff('0x2a')](_0x169fe3,_0x35f612)
  ;});let _0x4d88fc;if(_0x3a6e16==this[_0x14ff('0xa')]){_0x4d88fc=Status['gameWon'];}else{
  _0x4d88fc=Status[_0x14ff('0x2b')];}_0x4d88fc+=Status['playAgain'];_0x5cdcd4[_0x14ff('
  0x2c')](_0x4d88fc);if(this[_0x14ff('0xa')]=='B'){let _0x464e12=Messages[_0x14ff('0x2d')]
```

Code obfuscation

Node.js modules

- Code can be organised in **modules**
- Not all functions and variables in a module are exposed
  - Exposed elements have to be made known explicitly
- Modules can be **published** to npmjs.com
  - Makes distribution of modules to other developers easy, e.g.

```
npm install –save alexa-sdk
```

Search packages

Search

Need private packages and team management tools? Check out npm Orgs. »

express

4.17.1 • Public • Published 5 months ago

Readme | 30 Dependencies | 36,616 Dependents | 263 Versions

express

Fast, unopinionated, minimalist web framework for node.


```
const express = require('express')
const app = express()

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

install

```
> npm i express
```

weekly downloads

10,170,383

version	license
4.17.1	MIT

open issues	pull requests
116	57

homepage	repository
expressjs.com	github

last publish

5 months ago

```
claudiahauff@wlan-145-94-186-167:/local/pienapple-front-master $ npm list
pienapple-front@1.0.0 /local/pienapple-front-master
├┬ autoprefixer@6.4.0
│ ├── browserslist@1.3.5
│ ├── caniuse-db@1.0.30000517
│ ├── normalize-range@0.1.2
│ ├── num2fraction@1.2.2
│ ├┬ postcss@5.1.1
│ │ └── js-base64@2.1.9
│ └── postcss-value-parser@3.3.0
├┬ babel-core@6.11.4
│ ├┬ babel-code-frame@6.11.0
│ │ ├┬ chalk@1.1.3
│ │ │ ├── ansi-styles@2.2.1
│ │ │ ├── escape-string-regexp@1.0.
│ │ │ ├┬ has-ansi@2.0.0
│ │ │ │ └── ansi-regex@2.0.0
│ │ │ ├── strip-ansi@3.0.1
│ │ │ └── supports-color@2.0.0
│ │ ├── esutils@2.0.2
│ │ └── js-tokens@2.0.0
│ ├┬ babel-generator@6.11.4
│ │ └┬ detect-indent@3.0.1
│ │   ├── get-stdin@4.0.1
│ │   └┬ repeating@1.1.3
│ │    └┬ is-finite@1.0.1
│ │     └── number-is-nan@1.0.0
│ ├── babel-helpers@6.8.0
│ ├── babel-messages@6.8.0
│ ├┬ babel-register@6.11.6
│ │ ├── core-js@2.4.1
```

npm list

== (same value) vs. === (same type and value)

```
» var x = 3; typeof(x);
← "number"
» var y = NaN; typeof(y);
← "number"
» x == 3
← true
» x !== 3
← false
» y == NaN
← false
» y !== NaN
← true
```

5 lines (4 sloc) | 82 Bytes

```
1  'use strict';
2  module.exports = Number.isNaN || function (x) {
3      return x !== x;
4  };
```

Strict equality treats NaN as unequal to every other value -- including itself.

TECHNOLOGY LAB —

Rage-quit: Coder unpublished 17 lines of JavaScript and “broke the Internet”

Dispute over module name in npm registry became giant headache for developers.

npm search

```
2018-11-28 00:45:50   Claudias-MacBook-Air in ~/GitHub/Web-Teaching/demo-code/balloons-game
± |master → origin ↓11 {5} ✓| → npm search express
NAME                      | DESCRIPTION          | AUTHOR          | DATE       | VERSION  | KEYWORDS
express                   | Fast,…               | =dougwilson…    | 2018-10-11 | 4.16.4   | express framework sinatra web rest restful router app api
path-to-regexp            | Express style path…  | =blakeembrey…   | 2018-08-26 | 2.4.0    | express regexp route routing
cors                      | Node.js CORS…        | =dougwilson…    | 2018-11-04 | 2.8.5    | cors express connect middleware
morgan                    | HTTP request logger… | =dougwilson     | 2018-09-11 | 1.9.1    | express http logger middleware
apollo-server-express     | Production-ready…    | =abernix…       | 2018-11-26 | 2.2.3    | GraphQL Apollo Server Express Connect Javascript
express-validator         | Express middleware…  | =ctavan…        | 2018-07-23 | 5.3.0    | express validator validation validate sanitize sanitization xss
express-graphql           | Production ready…    | =asiandrummer…  | 2018-10-29 | 0.7.1    | express restify connect http graphql middleware api
helmet                    | help secure…         | =adam_baldwin…  | 2018-11-07 | 3.15.0   | security headers express connect x-frame-options x-powered-by cs
serve-favicon             | favicon serving…     | =dougwilson     | 2018-03-29 | 2.5.0    | express favicon middleware
@babel/helper-optimise-ca | Helper function to…  | =nicolo-ribaudo… | 2018-08-27 | 7.0.0    |
ll-expression             |                      |                 |            |          |
@feathersjs/express       | Feathers Express…    | =corymsmith…    | 2018-09-21 | 1.2.7    | feathers feathers-plugin
@babel/helper-member-expr | Helper function to…  | =nicolo-ribaudo… | 2018-08-27 | 7.0.0    |
ession-to-functions       |                      |                 |            |          |
babel-helper-optimise-cal | Helper function to…  | =hzoo…          | 2017-04-07 | 6.24.1   |
l-expression              |                      |                 |            |          |
babel-helper-explode-assi | Helper function to…  | =hzoo…          | 2017-04-07 | 6.24.1   |
gnable-expression         |                      |                 |            |          |
micromatch                | Glob matching for…   | =doowb =es128…  | 2018-03-22 | 3.1.10   | bash expand expansion expression file files filter find glob glo
aws-serverless-express    | This library…        | =amzn-oss…      | 2018-08-20 | 3.3.5    | aws serverless api gateway lambda express
@nguniversal/express-engi | Express Engine for…  | =nguniversal    | 2018-10-19 | 7.0.2    | express ssr universal
ne                        |                      |                 |            |          |
ignore                    | Ignore is a manager… | =kael           | 2018-11-03 | 5.0.4    | ignore .gitignore gitignore npmignore rules manager filter regex
express-rate-limit        | Basic IP…            | =nfriedly       | 2018-11-12 | 3.3.2    | express-rate-limit express rate limit ratelimit rate-limit middl
escape-string-regexp      | Escape RegExp…       | =sindresorhus   | 2016-02-21 | 1.0.5    | escape regex regexp re regular expression string str special cha
```

Code status	Stage	Rule	Example version
First release	New product	Start with 1.0.0	1.0.0
Backward compatible bug fixes	Patch release	Increment the third digit	1.0.1
Backward compatible new features	Minor release	Increment the middle digit and reset last digit to zero	1.1.0
Changes that break backward compatibility	Major release	Increment the first digit and reset middle and last digits to zero	2.0.0

npmjs documentation

Semantic versioning:

`Major.Minor.Patch`

(breaking backwards compatibility requires Major increment)

```
2018-11-28 09:47:53  wlan-145-94-167-249 in ~/GitHub/Web-Teaching/demo-code/balloons-game
± |master {5} ✔| → more package.json
{
  "name": "balloons",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "node app.js 3000",
    "test": "jest --coverage"
  },
  "dependencies": {
    "cookie-parser": "~1.4.3",
    "debug": "~2.6.9",
    "ejs": "~2.5.7",
    "express": "^4.16.3",
    "http-errors": "~1.6.2",
    "morgan": "~1.9.0",
    "ws": "^6.0.0"
  },
  "devDependencies": {
    "eslint": "^5.4.0",
    "jest": "^23.5.0"
  }
}
```

tilde range: `>=1.4.3 <1.5.0`

caret range: `>=4.16.3 <5.0.0`

https://github.com/npm/node-semver

A file-based module system

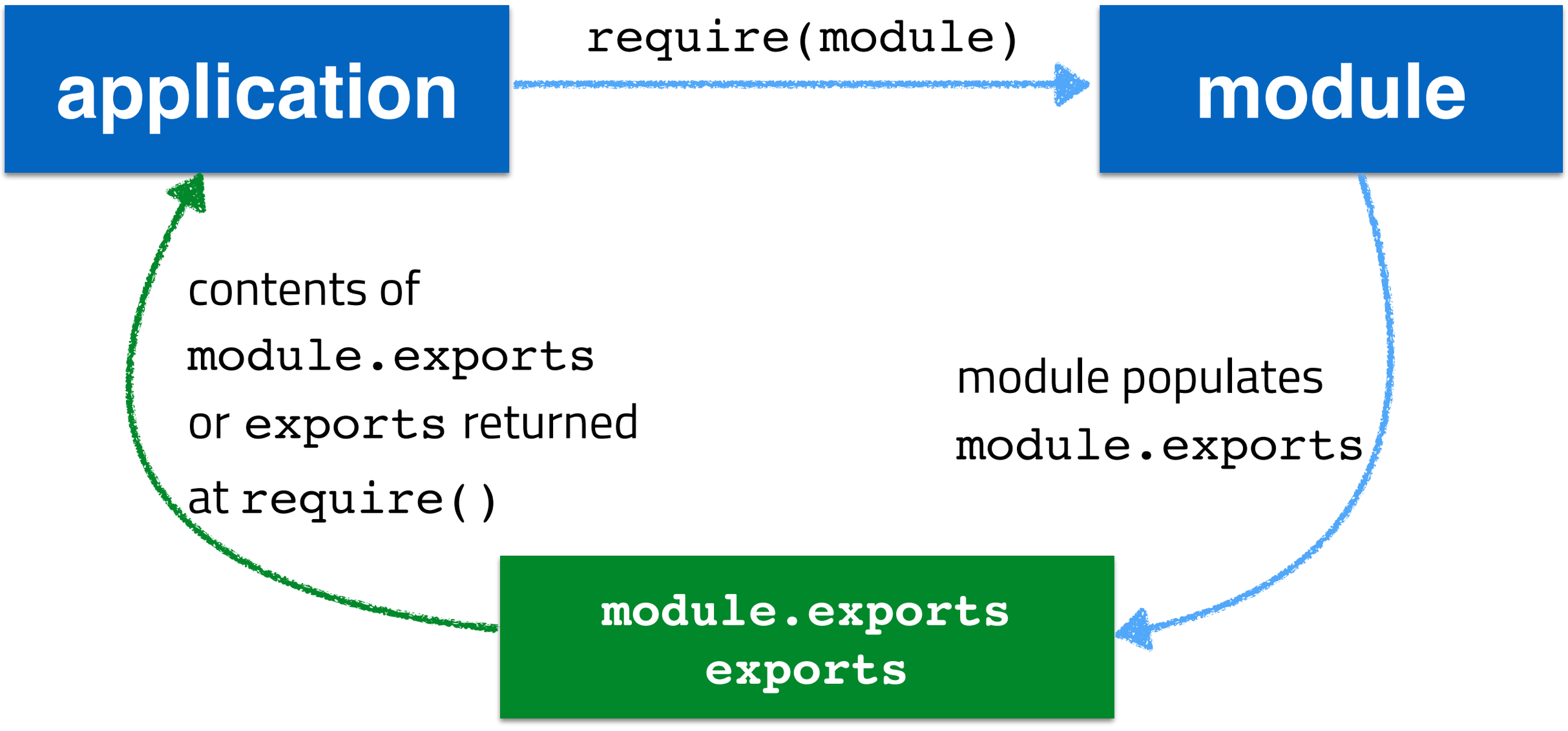
Do you remember how much effort we put into the module design pattern?

- A **file is its own module**; no pollution of the global namespace
- A file accesses its **module definition** through the `module` object
- **`module.exports`** (or its alias **`exports`**) makes code externally available
- To import a module, use the globally available **`require`** function

`console.log(module)`

```
module {
  id: '.',
  exports: {},
  parent: null,
  filename: '/path/to/nodejs-file.js',
  loaded: false,
  children: [],
  paths:
   [ '/Users/path/path2/node_modules',
    '/Users/path/node_modules',
     '/Users/node_modules' ]
}
```

App-module cycle



A first example

foo.js

```
var fooA = 1;
module.exports = "Hello!";
module.exports = function() {
   console.log("Hi from foo!");
};
```

bar.js

```
var foo = require('./foo');
foo();
require('./foo')();
console.log(foo);
console.log(foo.toString());
console.log(fooA);
console.log(module.exports);
```

Node.js **runs** the referenced JavaScript file in **a new scope** and returns the **final value** of **`module.exports`**

```
Hi from foo!
```

```
[Function]
```

```
{}
```

```
function () {
    console.log("Hi from foo!");
}
```

```
ReferenceError
```

`require()`

- `require()` is blocking
- `module.exports` is **cached**, i.e. the first time `require(a_file)` is called, `a_file` is read from disk, and subsequently the **in-memory object is returned**

- Useful to share states between modules (configuration, etc.)

```
var t1 = new Date().getTime();//milliseconds resolution
var foo1 = require('./foo');
console.log(new Date().getTime() - t1);
// > 0 (unless the machine is very fast)

var t2 = new Date().getTime();
var foo2 = require('./foo');
console.log(new Date().getTime() - t2); // approx 0
```

Higher resolution: `process.hrtime();//returns array [seconds, nanoseconds]`

`module.exports`

- `module.exports={}` is by default present in every Node.js file (a new empty object)
- Node.js provides an alias: `exports = module.exports`

```
module.exports.foo = function () {
    console.log('foo called');
};

module.exports.bar = function () {
    console.log('bar called');
};
```

equivalent

```
exports.foo = function () {
    console.log('foo called');
};

exports.bar = function () {
    console.log('bar called');
};
```

- Important: do **not assign** to `exports` directly, **attach** to it instead (otherwise the reference to `module.exports` is broken); assign only to `module.exports`

Client/server codesharing

`module.exports={}` is by default present in every Node.js file (a new empty object)

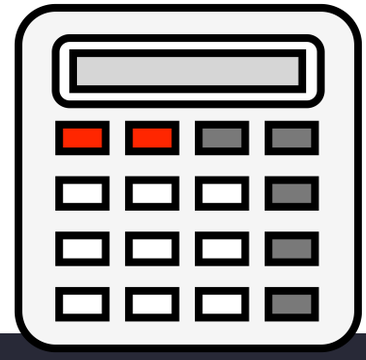
```
(function(exports) {
  exports.MAX_ALLOWED_GUESSES = 7;
  exports.MIN_WORD_LENGTH = 5;
  exports.MAX_WORD_LENGTH = 15;
  exports.WEB_SOCKET_URL = "ws://localhost:3000";
  exports.HIDDEN_CHAR = "#";
})(typeof exports === "undefined" ? (this.Setup = {}) : exports);
```

server-side

conditional operator

The code is deployed in the browser.

Creating a rounding module



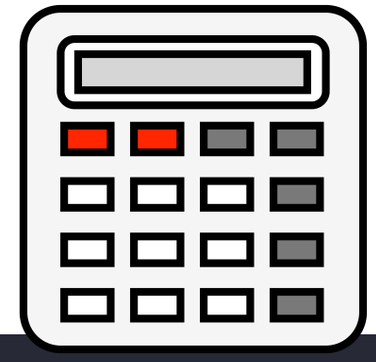
```
/* not exposed */
var errorString = "Grades must be a number between 1 and 10.";

function roundGradeUp(grade) {
  if (isValidNumber(grade) == false) {
    throw errorString;
  }
  return Math.ceil(grade) > 10 ? 10 : Math.ceil(grade); //max. is always 10
}

function isValidNumber(grade) {
  if (
    isNaN(grade) == true ||
    grade < exports.minGrade ||
    grade > exports.maxGrade
  ) {
    return false;
  }
  return true;
}

/* exposed */
exports.maxGrade = 10;
exports.minGrade = 1;
exports.roundGradeUp = roundGradeUp;
exports.roundGradeDown = function(grade) {
  if (isValidNumber(grade) == false) {
    throw errorString;
  }
  return Math.floor(grade);
};
```

grades.js

not exposed in this module;
application cannot use it

determines what exactly is exposed to the outer world

Using a module



```
var express = require("express");
var url = require("url");
var http = require("http");
var grading = require("./grades");
var app;

var port = process.argv[2];
app = express();
http.createServer(app).listen(port);

app.get("/round", function(req, res) {
  var query = url.parse(req.url, true).query;
  var grade = query["grade"] != undefined ? query["grade"] : "0";

  //accessing module functions
  res.send(
    "UP: " +
      grading.roundGradeUp(grade) +
      ", DOWN: " +
      grading.roundGradeDown(grade)
  );
});
```

require returns the contents of the exports obj.

adding our module (current directory)

accessing module functions

Middleware in Express

Middleware components

- **Small**, **self-contained** and **reusable** across applications
- Middleware components take **three arguments**:

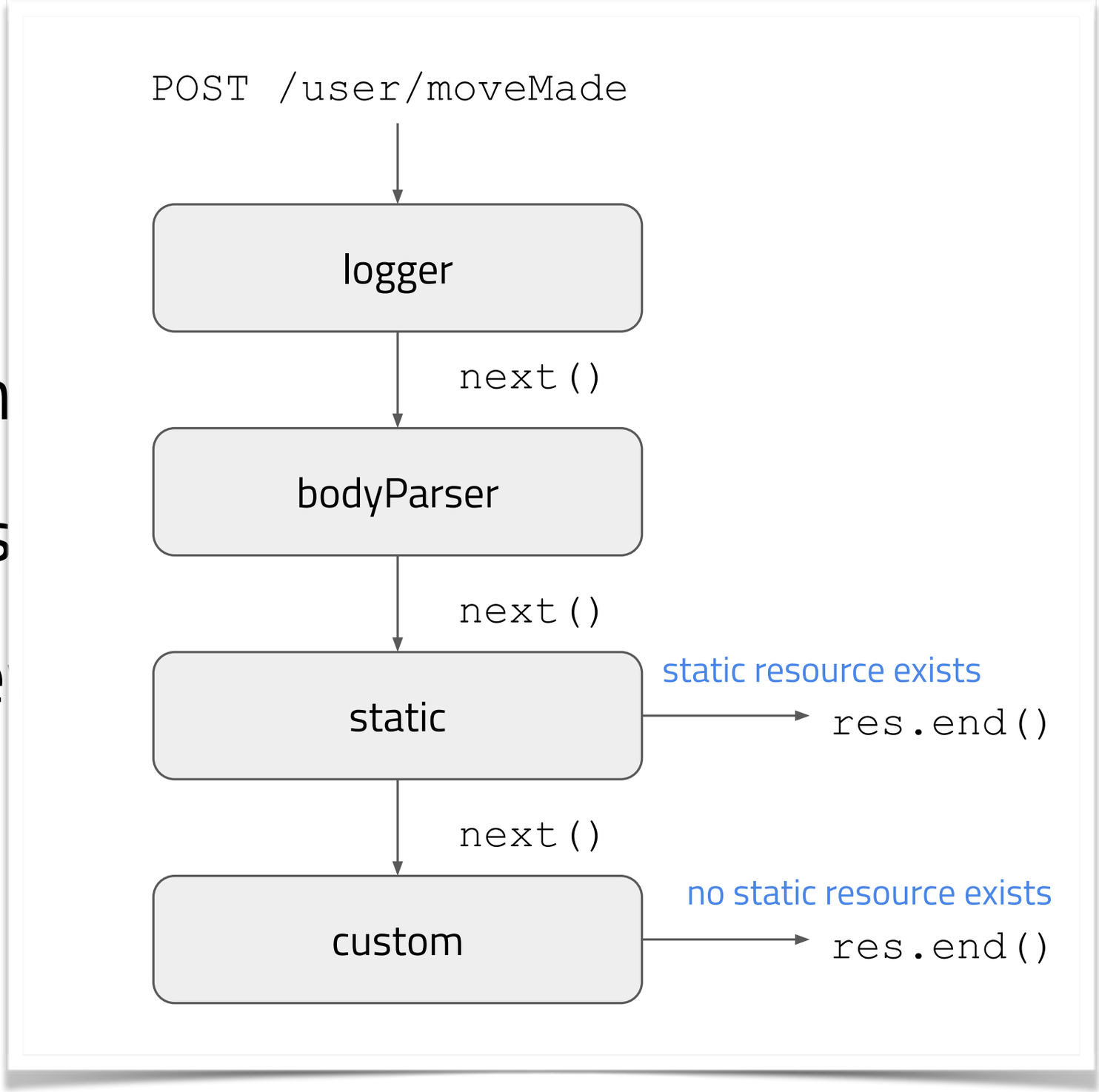
  - **HTTP request** object
  - **HTTP response** object
- Callback function **`next()`** to indicate that the component is finished and the *dispatcher* (orchestration component) can move to the next component

Middleware abilities

- **Execute** code
- **Alter** the request and response objects
- **End** the request-response cycle
- **Call** the next middleware function in the middleware stack

Middleware abilities

- **Execute** code
- **Alter** the request an
- **End** the request-res
- **Call** the next middle
  middleware stack



A simple logger component

- **Goal**: create a log file which records the request method and the URL of the request coming into the server
- **Required**: JavaScript function which accepts the request and response objects and the `next()` callback function

```
var express = require("express");

//a middleware logger comp
function logger(request, response, next) {
  console.log("%s\t%s\t%s", new Date(), request.method, request.url);
  next();
}
var app = express();
app.use(logger);
app.listen(3001);
```

string substitutions

control shifts to next middleware function

register middleware component

An authorisation component

```
app.use(function(req, res, next) {
  var auth = req.headers.authorization;
  if (!auth) {
    return next(new Error("Unauthorized access!"));
  }

  //extract username and password
  var parts = auth.split(" ");
  var buf = new Buffer(parts[1], "base64");
  var login = buf.toString().split(":");
  var user = login[0];
  var password = login[1];

  //compare to 'correct' username/password combination
  //hardcoded for demonstration purposes
  if (user === "user" && password === "password") {
    next();
  } else {
    return next(new Error("Wrong username/password combination!"));
  }
});
```

We can pass errors through next.

Only a correct login/password combination leads to next().

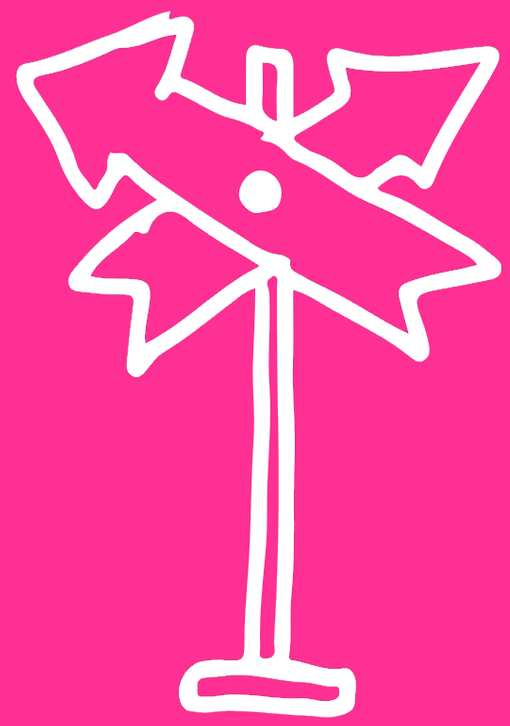
Components are configurable

- **Goal**: middleware components are reusable across applications
- **Approach**: wrap original middleware function in a setup function

```
function setup(options) {
  // setup logic
  return function(req, res, next) {
    // middleware logic
  };
}

app.use(setup({ param1: "value1" }));
```

important: function call is made!

Routing

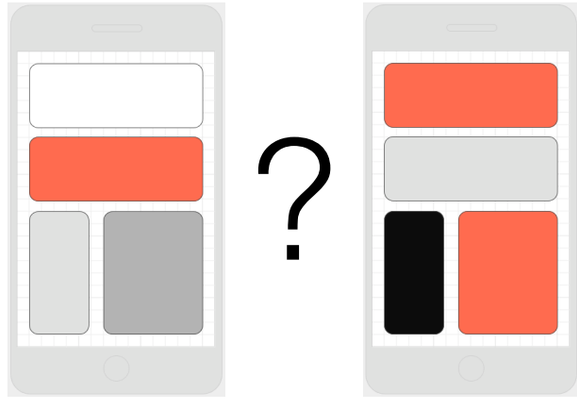


Overview

- Mechanism by which requests (as specified by a URL and HTTP method) **are routed to the code** that handles them
  - **Request types**: `GET /user` vs. `POST /user`
  - **Request routes**: `GET /user` vs. `GET /admin`

- In the past: **file-based routing**

Routes and Express



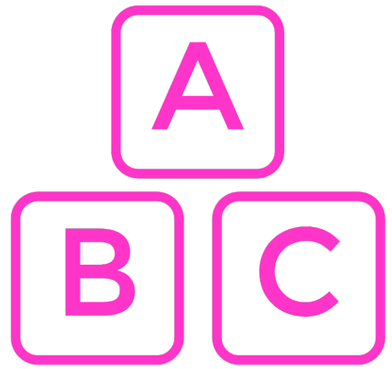
- We have used routes already
- **Route handlers are middleware**

```
//clients requests todos
app.get("/todos", function(req, res, next) {
  //hardcoded "A-B" testing
  if (Math.random() < 0.5) {
    return next();
  }
  console.log("Todos in schema A returned");
  res.json(todosA);
});

app.get("/todos", function(req, res, next) {
  console.log("Todos in schema B returned");
  res.json(todosB);
});
```

half the requests will move on

Two route handlers defined for a single route

Routes and Express




```
//A-B-C testing
app.get(
  "/todos",
  function(req, res, next) {
    if (Math.random() < 0.33) {
      return next();
    }
    console.log("Todos in schema A returned");
    res.json(todosA);
  },
  function(req, res, next) {
    if (Math.random() < 0.66) {
      return next();
    }
    console.log("Todos in schema B returned");
    res.json(todosB);
  },
  function(req, res) {
    console.log("Todos in schema C returned");
    res.json(todosC);
  }
);
```

A single `app.get()` can contain multiple handlers.

Idea: create **generic functions** that can be dropped into **any route.**

Routing paths and string patterns

- Routes are converted into **regular expressions** (*patterns to match character combinations in strings*) by Express
- **Route path types**: strings (*you have seen those*), **string patterns** (*we cover those now*), regular expressions (*we ignore those*)
- Express supports only a **subset** of the standard regex meta-characters
- Available regex meta-characters: **`+ ? * ( ) [ ]`**

Routing paths and string patterns

* in other languages/frameworks: zero or more occurrences of the **preceding element** (regex)**!**

- Routes are converted into **regular expressions** (*patterns to match character combinations in strings*) by Express
- **Route path types**: strings (*you have seen those*), **string patterns** (*we cover those now*), regular expressions (*we ignore those*)
- Express supports only a **subset** of the standard regex meta-characters
-

Character	Description	Regex	Matched expressions
+	one or more occurrences	ab+cd	abcd, abbcd, ...
?	zero or one occurrence	ab?cd	acd, abcd
*	zero or more occurrences of any char (wildcard)	ab*cd	abcd, ab1234cd, ...
[...]	match anything inside for one character position	ab[cd]?e	abe, abce, abde
(...)	boundaries	ab(cd)?e	abe, abcde

FYI only! We ignore those in this course.

regular expression, delimited by / /

```
app.get(/.*users$/, function(req,res){
  res.send(…)
});
```

```
(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]
+)*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\
[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-
z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:(2(5[0-5]|
[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:(2(5[0-5]|[0-4]
[0-9])|1[0-9][0-9]|[1-9]?[0-9])|[a-z0-9-]*[a-z0-9]:(?:[\x01-
\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-
\x09\x0b\x0c\x0e-\x7f])+)\])
```

Routing parameters

Enable **variable input** as part of the route

```
var todoTypes = {
  important: ["TI1500", "ADS", "Calculus"],
  urgent: ["Dent
  unimportant: ['
};

app.get("/todos/:type", function(req, res, next) {
  var todos = todoTypes[req.params.type];
  if (!todos) {
    return next(); // will eventually fall through to 404
  }
  res.send(todos);
});
```

Will match any string that does not contain /. It is available with key `type` in the `req.params` object.

```
localhost:3000/todos/urgent
localhost:3000/todos/unimportant
```

Routing parameters

localhost:3000/todos/important/tomorrow

```
var todoTypes = {
  important: {
    today: ["TI1500"],
    tomorrow: ["ADS", "Calculus"]
  },
  urgent: {
    today: ["Dentist", "Hotel booking"],
    tomorrow: []
  },
  unimportant: {
    today: ["Groceries"],
    tomorrow: []
  }
};
app.get("/todos/:type/:level", function(req, res, next) {
  var todos = todoTypes[req.params.type][req.params.level];
  if (!todos) {
    return next();
  }
  res.send(todos);
});
```

No restrictions on the number of variable input!

Organizing routes

- Keeping routes in the main application file becomes unwieldy as the codebase grows
- Move routes into a module and pass `app` instance into the module

routes.js

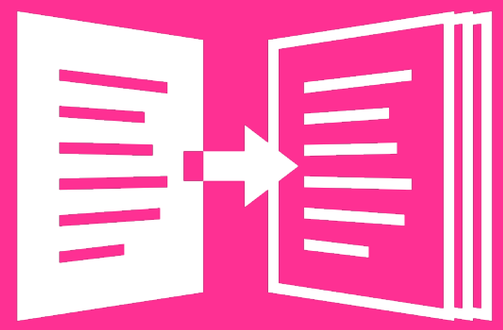
```
module.exports = function(app){
   app.get('/', function(req,res){
      res.send(...);
   }))
   //...
};
```

app.js

```
require('./routes.js')(app);
```

Templating

with EJS



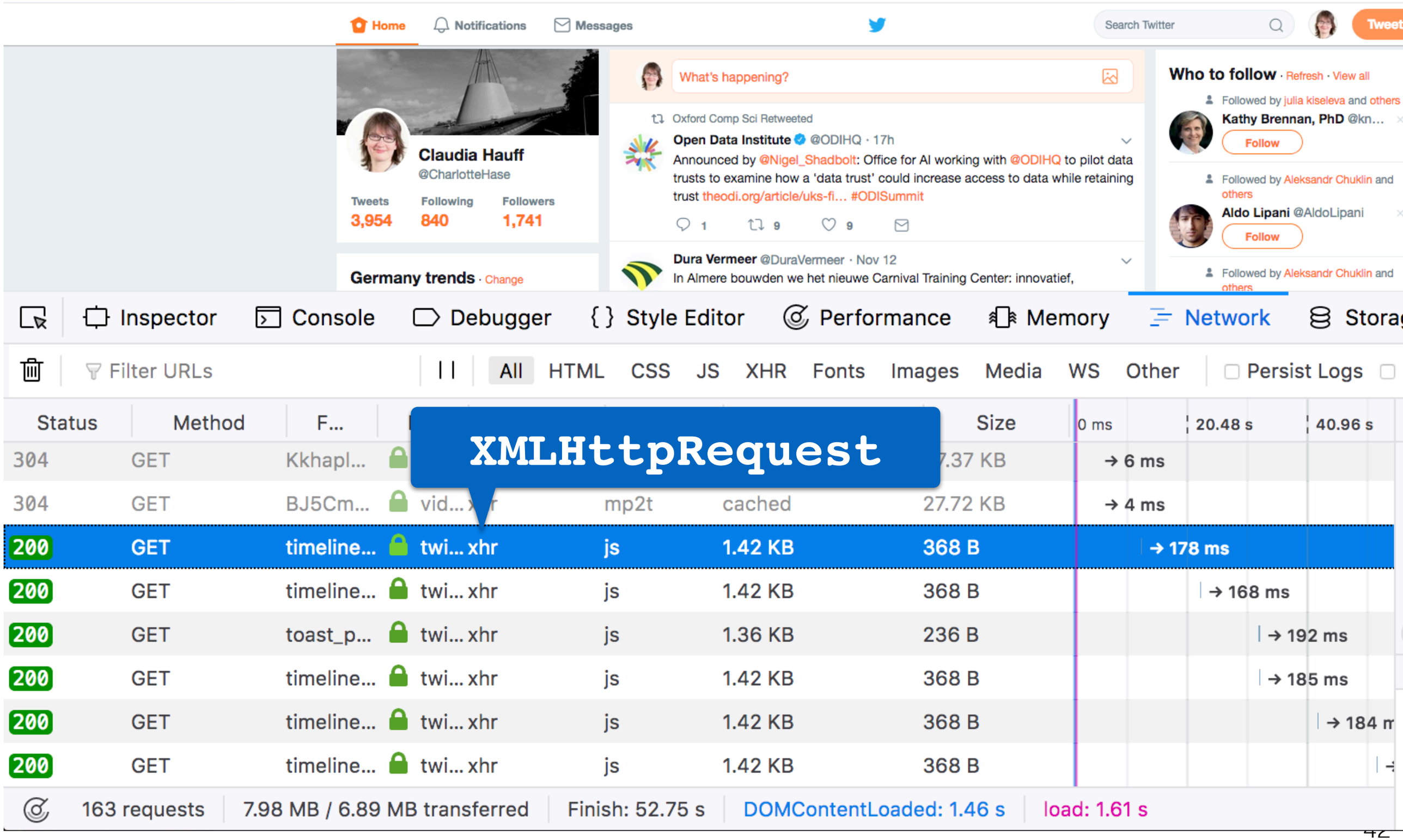
Recall … Express & HTML

```
1  var express = require("express");
2  var url = require("url");
3  var http = require("http");
4  var app;
5
6  var port = process.argv[2];
7  app = express();
8  http.createServer(app).listen(port);
9
10 app.get("/greetme", function (req, res) {
11  var query = url.parse(req.url, true).query;
12  var name = ( query["name"]!=undefined) ? query[
13                "name"] : "Anonymous";
14  res.send("<html><head></head><body><h1>
15            Greetings "+name+"</h1></body></html>
16            ");
17 });
18
19 app.get("/goodbye", function (req, res) {
20  res.send("Goodbye you!");
21 });
```

Error-prone, **not maintainable**, fails at anything larger than a toy project.

Recall … Ajax

one extra request/response cycle



Instead … templates

- **Goal:** write as little HTML "by hand" as possible

- Separate logic from presentation markup
- Different **template engines** exist for Node.js
- We use **EJS** (**E**mbedded **J**ava**S**cript), a popular **template engine** and language: **npm install ejs** (https://www.npmjs.com/package/ejs)

Model-View-Controller (MVC)

- Standard **design pattern** to keep **logic**, **data** and **presentation** separate

- User request of a resource from the server triggers a cascade of events:

  1. **controller** requests application data from the **model**
  2. **controller** passes the data to the **view**
  3. **view** formats the data for the user (template engines)

EJS on the Node REPL

- **Node REPL**: **R**ead–**E**val–**P**rint-**L**oop implementation to quickly test Node snippets (an alternative to VSC)

- Start the REPL via:

```
node -e “require(‘repl’).start({ignoreUndefined:true})”
```

<%= outputs the value into the template (HTML escaped)

```
var ejs = require("ejs");
var template = "<%= message %>";
//<%= outputs the value into the template (HTML escaped)
var context = { message: "Hello template!" };
console.log(ejs.render(template, context));
```

EJS on the Node REPL

- **Node REPL**: **R**ead–**E**val–**P**rint-**L**oop implementation to quickly test Node snippets (an alternative to VSC)

- Start the REPL via:

```
node -e “require(‘repl’).start({ignoreUndefined:true})”
```

`<%-` outputs the value into the template (unescaped); enables cross-site scripting attacks

```
var ejs = require("ejs");
var template = "<%- message %>"; //UNESCAPED
//var template = '<%= message %>';      //ESCAPED
var context = { message: "<script>alert('hi!');</script>" };
console.log(ejs.render(template, context));
```

EJS: user-defined functions

Often you want to make slight changes: **transform** or **filter**

```
var ejs = require("ejs");
var people = ["wolverine", "paul", "picard"];
var transformUpper = function(inputString) {
  return inputString.toUpperCase();
};
var template = '<%= helperFunc(input.join(", ")); %>';
var context = {
  input: people,
  helperFunc: transformUpper //user-defined function
};
console.log(ejs.render(template, context));
```

define your own function

function

array

define the context object

JavaScript within EJS templates

```
var ejs = require("ejs");

var template =
  "
<% if(movies.length>2){movies.forEach(function(m){console.log(m.title)})}
 %>
";

var context = {
  movies: [
    { title: "The Hobbit", release: 2014 },
    { title: "X-Men", release: 2016 },
    { title: "Superman V", release: 2014 }
  ]
};

ejs.render(template, context);
```

Array.prototype.forEach()

use <% for control-flow purposes; no output

applied to each element

Configuring views with Express

- **Setting** the **views** directory (the directory containing the templates)

```
app.set('views', __dirname + '/views');
```

- **Setting** the **template engine**

```
app.set('view engine', 'ejs');
```

- Creating **template files**

Exposing data to views

```
var express = require("express");
var url = require("url");
var http = require("http");
var app;

var port = process.argv[2];
app = express();
http.createServer(app).listen(port, function() {
  console.log("Ready on port " + port);
});

var todos = [];
todos.push({ message: "Final exam", dueDate: "January 2016" });
todos.push({ message: "Prep for assignment 6", dueDate: "05/01/2016"
todos.push({ message: "Sign up for exam", dueDate: "06/01/2016" });

app.set("views", __dirname + "/views");
app.set("view engine", "ejs");

app.get("/todos", function(req, res) {
  res.render("todos", { title: "My list of TODOs", todo_array: todos
});
```

the list of todos we want to serve to the clients

informing Express about the view templates

Exposing data to views

```
var express = require("express");
var url = require("url");
var http = require("http");
var app;

var port = process.argv[2];
app = express();
http.createServer(app).listen(port, function() {
  console.log("Ready on port " + port);
});

var todos = [];
todos.push({ message: "Final exam", dueDate: "January 2016" });
todos.push({ message: "Prep for assignment 6", dueDate: "05/01/2016"
todos.push({ message: "Sign up for exam", dueDate: "06/01/2016" });

app.set("views", __dirname + "/views");
app.se                                    );

app.ge                                  eq, res) {
  res.render("todos", { title: "My list of TODOs", todo_array: todos
})
```

the list of todos we want to serve to the clients

`render()` indicates the use of a template

informing Express about the view templates

template to use

arguments for the template

EJS template file

```
<!DOCTYPE html>
<html>
  <head>
    <title><%= title %></title>
  </head>
  <body>
    <h1>TODOs</h1>
    <div>
      <% todo_array.forEach(function(todo) { %>
      <div>
        <h3><%=todo.dueDate%></h3>
        <p><%=todo.message%></p>
      </div>
      <% }) %>
    </div>
  </body>
</html>
```

JavaScript between <% %> is **executed**.

JavaScript between <%= %> **adds output** to the result file.

- Work on **Assignments 5/6!**

- Sign up for your group's spot in the queue (so far we have only ~100 signups!).

- Next week's tutorial works through past exam questions.