javascript on frontend

lukas.weber@socialbakers.com

<<< Historie

- 1995 Netscape Brendan Eich
- původně k “oživení” HTML stránek
- s příchodem AJAXu základem pro webové aplikace
- Single Page Applications
- dnes je všude, prohlížeče, servery, HW

LEVEL 1

<<< Doba temna

- “Optimized for IE5” a animované “under construction” GIFy
- Mix HTML a <script> tagu
- Validace formulářů
- Převážně obťežující kód
- Bezpečnost …nebyla

Válka o web

- Konkurence
  - Macromedia Flash
  - JAVA Applet
- Války prohlížečů (Netscape prohrál)
- Žádný ekosystém

DOM

- Potřeba manipulovat se stránkou
- Nekompatibilní implementace mezi prohlížeči
- Model událostí
- Internet Explorer 5

LEVEL 2

Rehabilitace

- AJAX!
- Vznik PrototypeJS, Mootools, jQuery
- Unobtrusive javascript
- CSS Selectory
- Pořád matlanice html + js
- Bezpečnost se začala trochu řešit

AJAX?

- XMLHttpRequest
- Využití CPU klienta
- Menší přenosy dat
- Lepší uživatelská zkušenost
- Přesun od webových stránek k webovým aplikacím

LEVEL 3

SINGLE PAGE APPLICATIONS!!!

AJAX || AJAJ?

- XML prohrálo válku o internet, ať žije JSON!
- REST

Here comes troubles

- Ultimátní komplikace - zavedení **STAVU**

URLWTF?

- URL nemá obsah
- index.html …<body>nic</body>
- Nefunkční tlačítko back
- Firmy přichází o peníze

URLWTF!

- hashbang #!
- GoogleBot _escape_fragment

Dependency hell

- include #stdio.h … <script src="stdio.js"></script>
- CommonJS
- AMD

AMD

- require([‘myhelper’], function (myhelper) { … });
- define(‘myhelper’, [‘stdio’], function (stdio) { … });

CommonJS

- exports.myhelper = function () {};
- var helper = require('./myhelper');

Bower

- Řeší závislosti třetích stran
- bower init
- bower install jquery
- bower list —paths
- {“jquery": "bower_components/jquery/jquery.js"}

Chrocht, chrocht

- Grunt.js!
- Makefile v Javascriptu
- Ekosystém modulů
- grunt build

Transpilery

- CoffeeScript
- TypeScript
- ClojureScript
- TypeScript
- DART!

Architektura

- Model View Controler (MVVM, MVP, …)
- jQuery nestačí
- Vyšší forma abstrakce - frameworky

Backbone.js

- Implementuje Model a Controller
- Routování

Angular.js

- Vše jako backbone
- Deklarativní zápis díky direktivám přímo v HTML
- Dependency injection
- Binding - dirty checking
- Google

Closure Tools

- Mnohem starší
- Kompiler + knihovna
- Google, ale nepropaguje ji - iniciativa googlerů
- Správa závislostí

Closure Compiler

- Minfikuje javascript
- Optimalizuje
- Eliminuje mrtvé větve kódu - Advanced Kompilace

Closure Library

- Knihovna
- Unifikuje memory management, události, komponenty
- Know how developerů z Google
- Rock solid!
- S-křivka učení
- Víc psaní, víc práce…

Library

builder PRO

Library - Dependency Management

- namespaces
- goog.require(), goog.provide()
- calcdeps.py

builder PRO

Library - Exports

- komunikace s okolím + hinty pro compiler
- goog.exportProperty()
- goog.exportSymbol()

Library - OOP Helpery

- OOP v Javascriptu
  - založeno na prototypech místo na třídách
- goog.inherits() - dědičnost
- goog.base() - super, parent

builder PRO

Library - Type Assertions

- goog.isDef()
- goog.isNull()
- goog.isString() - string, String
- goog.isArray()/goog.isArrayLike()
- ...

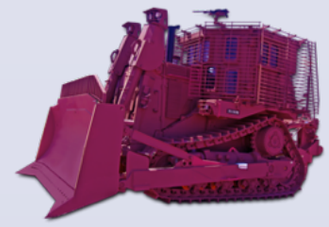
builder PRO

Library - Helpery pro funkce

- goog.bind() - “bindování” na kontext
- goog.partial() - fixování parametrů funkce

builder PRO

Library - fwd»



goog.assert

goog.math

goog.object

goog.array

goog.string

goog.dom

goog.events

goog.net

goog.async

goog.storage

goog.ui

goog.editor

goog.testing

goog.tweak

builder PRO

Základní objekty

builder PRO

goog.Disposable

- správa paměti?
- disposeInternal()
- isDisposed()

“MEMORY LEAK” je reference na objekt, který garbage collector nemůže odstranit!
(...WTF?)

builder PRO

goog.Disposable → **goog.EventTarget**

- správa událostí
- W3C DOM 2 Level Events
- addEventListener()
- dispatchEvent()
- get/setEventTarget()

builder PRO

goog.ui

builder PRO

goog.EventTarget → **goog.ui.Component**

- životní cyklus a hiearchie komponent
- renderování a dekorování
- helpery - DOM, Events
- konstanty
  - goog.ui.Component.EventType
  - goog.ui.Component.State
  - goog.ui.Component.Error

builder PRO

vytvoření instance
nastavení

vytvoření DOMu
nabindování eventů
...

život a práce

smrt a odklizení

new goog.ui.Componet

nová instance

render()

decorate()

createDom()
enterDocument()

canDecorate()
decorateInternal()
enterDocument()

zobrazení

dispose()

dispose()

disposeInternal()
exitDocument()

disposeInternal()

zrušeno

builder PRO

goog.ui.Component → **goog.ui.Control**

- hlavní třída pro prvky UI
- zpracování vstupů z klávesnice a myši
- podpora stavů a přechodů mezi nimi
- pluggable renderer framework
- další helpery (WAI-ARIA, class, ...)

builder PRO

goog.ui.Component → goog.ui.Control

Zpracování vstupu od uživatele

- setHandleMouseEvents()
  - setHandleMouse{Event}()
- getKeyEventTarget()
- getKeyHandler()
  - handleKeyEventInternal()

builder PRO

goog.ui.Component → goog.ui.Control

Stavy

- autostates
  - setSupportedState()
  - “vyšší smysl” != DOM
- goog.ui.Component.State
  - DISABLE
  - HOVER
  - ACTIVE
  - FOCUSED
  - SELECTED
  - CHECKED
  - OPENED

builder PRO

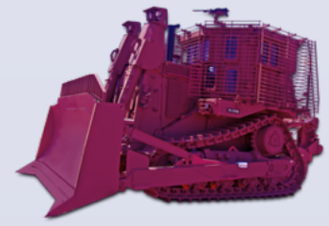
goog.ui.Component → goog.ui.Control

Události

- setDispatchTransitionEvents()
  - každý stav má dvě přechodové události
  - goog.ui.Component.State.CHECKED
    - goog.ui.Component.EventType.CHECK
    - goog.ui.Component.EventType.UNCHECK

builder PRO

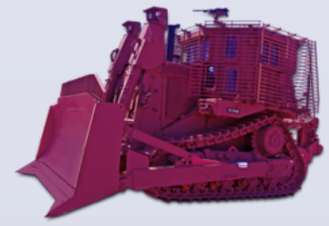
goog.ui.Component → goog.ui.Control



Pluggable renderer framework

- Control může renderování delegovat potomkovi goog.ui.ControlRenderer
- nejběžnější u komponent jako Menu, Button
- starší komponenty renderery nepoužívají

goog.ui.Component → goog.ui.Container



“Úplně debilní jméno pro Menu”

- Kontejnér pro Controly
  - umožňuje vytvořit takové komponenty jako Menu (spravuje šipky atd.)
  - deleguje na sebe události, které by jinak musely zpracovat Controly - tzn. ušetříte handlery, paměť, koťátka, ...

builder PRO

goog.net

builder PRO

goog.net

- goog.net.XhrIo / XhrPool / XhrManager
- goog.net.NetworkStatusMonitor
- goog.net.ImageLoader / FileDownloader
- goog.net.Cookies

Builder experience™

builder PRO

Builder experience™

- všechny komponenty musí fungovat i nevyrenderované (nesmíte se spolehát na DOM - DOM je view a view nevlastní data)
- minimalizovat počet handlerů událostí, snažit se omezit DOM Eventy
- model - setModel(), getModel(), nepředávat reference, nesnažit se o ActiveRecord-like modely
- v createDom() goog.style.getSize() pro zrovna přidané elementy vrátí většinou nulovou velikost nebo nějakou blbost - blokujete hlavní thread
- celkově se goog.style.getSize() vyhnout, je to neskutečně drahá záležitost => co můžete, řešte pomocí CSS
- když už něco počítáte, tak první hledejte v goog.style, goog.positioning
- nextTick() == setTimeout(function () {}, 0)

builder PRO

Builder experience™

- goog.async.Deferred pomáhá řešit callbackový peklo
- nepoužívat konstanty jako “change” ale namespace.ui.MyComponent.EventType.CHANGE
- nevěřte dokumentaci - jděte se podívat do zdrojáku, všechno je velice dobře okomentované
- když už musíte handlovat vlastní vstupy => goog.event.*Handler
- updatu Closure se nemusíte bát ani bez testů (je rock stable), nedostane se tam nic, co by předtim googleři nezkusili u sebe na produkci (600M useru!)
- pojmenovávejte i lambdy
- naučte se javascript - funkce je jen reference, můžete s ní pracovat jako s proměnnou, předávat jako parametry, vracet jako výsledek

builder PRO

React

- Facebook
- Inspirace u herních enginů -> brutální rychlost
- Syntetický DOM i event systém
- Minimální množství změn v DOMu

Performance

- DOM manipulace -> klesá framerate, blokuje
- Chrome Develeporer Tools -> Profiler, Heap dump
- 30-60 FPS
- Mobilní zařízení (málo paměti, vytížení CPU, …)

Performance

- Memory leaky - incrementální heap dumpy
- Stack introspection
- Flame graphs

LEVEL 4

Budoucnost >>>

- Mobile First
- WebSockets / HTTP 2.0
- WebRTC
- WebComponents
- WebGL
- Meteor.js
- Chrome OS
- ECMAScript 6/Harmony

A co dál?

- Scala, Clojure, Elixir/Erlang
- DART, Go!
- Učte se nové věci, nepřestávejte!
- Matematika z vás udělá hvězdy $!

Co čist?

- www.echojs.com
- news.ycombinator.com
- @addyosmani
- @paul_irish
- @techbakers

Game Over

Děkuji!

@techbakers

lukas.weber@socialbakers.com

twitter: @isnotgood