

Spis treści

Wprowadzenie	9
Underscore.js	11
Czym jest Underscore.js?	11
Dlaczego warto poznać Underscore.js?	11
Dlaczego tutaj o nim mówię?	12
Instalacja Underscore.js	12
Przykłady użycia	13
_.each(list, iterator)	17
_.map(list, iterator)	18
_.reduce(list, iterator, memo)	19
_.reduceRight(list, iterator, memo)	20
_.find(list, iterator)	21
_.filter(list, iterator)	22
_.where(list, properties)	22
_.findWhere(list, properties)	25
_.reduce(list, iterator)	26
_.every(list, iterator)	27
_.some(list, iterator)	28
_.contains(list, value)	29
_.invoke(list, methodName)	30
_.pluck(list, propertyName)	31
_.max(list, iterator) i _.min(list, iterator)	32
_.sortBy(list, iterator)	33
_.groupBy(list, iterator)	34
_.countBy(list, iterator)	36
_.shuffle(list)	37
_.size(list)	38
_.first(array) i _.take(array, n)	38
_.initial(array, n), _.last(array, n)	
i _.rest(array, index)	40

<code>_.compact(array)</code>	41
<code>_.flatten(array, shallow)</code>	42
<code>_.without(array, values)</code>	43
<code>_.union(arrays)</code>	43
<code>_.intersection(arrays)</code>	44
<code>_.difference(array, array)</code>	45
<code>_.object(list, values)</code>	45
<code>_.indexOf(array, value)</code> i <code>_.lastIndexOf(array, value)</code>	46
<code>_.sortedIndex(list, value)</code>	46
<code>_.range(start, stop, step)</code>	47
<code>_.bind(function, object, arguments)</code>	48
<code>_.partial(function, arguments)</code>	49
<code>_.memoize(function)</code>	50
<code>_.delay(function, wait, arguments)</code>	51
<code>_.defer(function, arguments)</code>	52
<code>_.throttle(function, wait)</code>	53
<code>_.debounce(function, wait)</code>	54
<code>_.once(function)</code> i <code>_.after(counter, function)</code>	55
<code>_.wrap(function, wrapper)</code>	56
<code>_.compose(functions)</code>	57
<code>_.chain(object)</code> i <code>_(object).value()</code>	58
<code>_.tap(object, interceptor)</code>	59
<code>_.keys(object)</code> , <code>_.values(object)</code> , <code>_.pairs(object)</code> oraz <code>_.invert(object)</code>	60
<code>_.functions(object)</code>	62
<code>_.extend(destination, sources)</code> , <code>_.pick(object, keys)</code> , oraz <code>_.omit(object, keys)</code>	62
<code>_.clone(object)</code>	64
<code>_.has(object, key)</code>	65
<code>_.isEqual(object1, object2)</code>	66
<code>_.is</code> (na sto sposobów)	67
<code>_.noConflict()</code>	67
<code>_.times(n, iterator)</code> i <code>_.random(min, max)</code>	67
<code>_.mixin(object)</code>	68
<code>_.unique(prefix)</code>	69
<code>_.escape(text)</code> i <code>_.unescape(text)</code>	69
<code>_.result(object, property)</code>	70
<code>_.template()</code>	71
Podsumowanie	71

Backbone.js	73
Na początek o edytorze tekstowym słów kilka. Sublime Text	73
Czym jest Backbone.js	75
Coś za coś	76
Dlaczego warto poznać Backbone.js?	76
Niezbędne minimum	77
Bierzemy się do pracy	77
Modele – podstawy	77
Modele – zdarzenia	82
Modele – walidacja	86
Widoki – podstawy teoretyczne	88
Widoki – podstawy praktyczne	89
Routes (wreszcie napiszemy coś większego)	93
Routes – praktyczny przykład działania	94
Chwila oddechu	102
Część serwerowa	107
Pierwsza komunikacja z serwerem	107
Pobieranie danych z serwera	112
Podsumowanie	115
AngularJS	117
Czym jest AngularJS?	117
Dlaczego warto używać AngularJS?	118
Jaki mamy plan?	119
Zasiejmy ziarno	119
Uruchomienie aplikacji	125
Małe porządki	127
Kontrolery	128
Zdarzenia	138
Dyrektywy – czyli takie trochę inne argumenty	141
Bindowanie	142
Cloak	145
nonBindable	146
Wyrażenia	147
Filtry	148
Tworzenia własnego filtra	152
Wiązanie obustronne	154
Walidacja	156
Serwisy	160
Pierwszy własny serwis	162
Serwisy wbudowane	164

Lokalizacja	169
Timeout	172
Ruting	175
Dyrektywy	185
Tworzymy dyrektywy	186
Dyrektywa korzystająca z danych zewnętrznych	190
Dyrektywy reagujące na zmiany wartości	192
Podsumowanie	196
Knockout.js	197
Od czego zacząć?	197
Środowisko	201
Podstawy	205
Zmienne obserwowane (<i>observables</i>)	209
Wiązania	215
toJSON	228
Podsumowanie	230
Podsumowanie, podziękowanie i kilka słów ode mnie	231