



Inhaltsverzeichnis

Arno Hollosi

Von Geodaten bis NoSQL: Leistungsstarke PHP-Anwendungen

Aktuelle Techniken und Methoden für Fortgeschrittene

ISBN (Buch): 978-3-446-42995-6

ISBN (E-Book): 978-3-446-43122-5

Weitere Informationen oder Bestellungen unter

<http://www.hanser-fachbuch.de/978-3-446-42995-6>

sowie im Buchhandel.

Inhalt

Vorwort	XV
---------------	----

Teil I

Den Benutzer im Fokus: Geodaten, Zeitzonen und Schriftzeichen	1
1 Geodaten	3
1.1 Geolokalisierung in HTML5	4
1.1.1 Abfrage der aktuellen Position	4
1.1.2 Ergebnis der Positionsabfrage	5
1.1.3 Verfolgen von Positionsänderungen	6
1.1.4 Optionen für die Positionsabfrage	7
1.1.5 Zusammenfassung	8
1.2 Lokalisierung über IP-Adresse	8
1.2.1 Installation der GeoIP-PHP-Erweiterung	8
1.2.2 Installation der Datenbank	9
1.2.3 Auswerten der Daten	9
1.2.4 Vor- und Nachteile	11
1.3 GEOS	11
1.3.1 Installation	11
1.3.2 Geometrische Formen und Datenformate	12
1.3.3 Erzeugen von geometrischen Objekten	15
1.3.4 Abfragen, ob ein Objekt ein anderes beinhaltet	17
1.3.5 Abfragen von Objektüberschneidungen	19
1.3.6 Verändern und Kombinieren von Objekten	20
1.3.7 Abfragen von Eigenschaften	23
1.3.8 Bestimmen von Distanzen, Längen und Flächen	26
1.3.9 Zusammenfassung	27
1.4 Projektionen	27
1.4.1 Arten von Projektionen	28
1.4.2 Georeferenzsysteme	31
1.4.3 Auswahl eines Georeferenzsystems	33
1.5 Zusammenfassung	36

2	Geodaten speichern	37
2.1	MySQL und Geodaten	37
2.1.1	Datentypen und Datenformate	38
2.1.2	Beispieldaten: Landkreise und Geoinformationen	40
2.1.3	Einfache geometrische Funktionen	43
2.1.4	Abfragen geometrischer Beziehungen	45
2.1.5	Erstellen eines geometrischen Index	46
2.1.6	Zusammenfassung	48
2.2	PostGIS	48
2.2.1	Installation	49
2.2.2	Anlegen der Datenbank	50
2.2.3	Erstellen einer Tabelle mit Geodaten	51
2.2.4	Laden von Shape-Dateien	52
2.2.5	Ausgabedatenformate	53
2.2.6	Geography-Datentyp	55
2.2.7	Erstellen geometrischer Indexe	58
2.2.8	Beispiele für Geo-Abfragen	60
2.2.9	Kombinieren mit Maßzahlen (Measure)	65
2.3	Routenberechnung mit SpatialLite	70
2.3.1	Installation	70
2.3.2	Laden und Indizieren von OSM-Daten	72
2.3.3	Suche des kürzesten Weges	73
2.3.4	Zusammenfassung	75
2.4	Kartendarstellung mit OpenLayers	76
2.4.1	Software für einen Kartendienst	76
2.4.2	Vorteile von OpenLayers	77
2.4.3	Einfache Kartendarstellung	77
2.4.4	Route in einem Layer darstellen	79
2.5	Zusammenfassung	81
3	Zeit und Zeitzonen	83
3.1	Zeitzonen und koordinierte Weltzeit (UTC)	83
3.1.1	Angabe von Zeitzonen	84
3.1.2	Veränderungen der Zeitzonen	85
3.1.3	Zeitzonendatenbank	85
3.2	Sommerzeit	86
3.3	Zeitangaben in PHP	87
3.3.1	Verwendung von Unix-Zeitstempeln	87
3.3.2	DateTime-Objekte	88
3.4	Zeitdatentypen in MySQL	90
3.4.1	Vorbereiten und Aktualisieren der Datenbank	91
3.4.2	Umrechnungen	91
3.4.3	Zeit- und Datumsfunktionen	92
3.4.4	Empfohlene Datentypen	93

3.5	Nicht-Gregorianischer Kalender	94
3.6	Ermitteln der Zeitzone des Benutzers	96
3.6.1	Abfragen der Client-Zeit mit JavaScript	96
3.6.2	Zeitzone über Ort bestimmen	97
3.6.3	Vor- und Nachteile	98
3.7	Zusammenfassung	98
4	Unicode	99
4.1	Zeichenkodierung	99
4.1.1	Kodierung und Decodierung	100
4.1.2	Zeichensatz	101
4.2	Unicode-Grundlagen	102
4.2.1	Codepoints	103
4.2.2	Abstrakte Zeichen und Äquivalente	103
4.2.3	Grapheme und Komposita	104
4.2.4	UTF-8 (Zeichenkodierung)	105
4.2.5	UTF-16 (Zeichenkodierung)	106
4.2.6	UTF-32 (Zeichenkodierung)	108
4.2.7	Zusammenfassung	108
4.3	PHP und Unicode	109
4.3.1	Unicode ist ein Muss	109
4.3.2	Unicode im PHP-Quellcode	109
4.3.3	Unicode ohne Unicode-Unterstützung	110
4.3.4	Validieren und Normalisieren von UTF-8-Zeichenfolgen	112
4.3.5	Multibyte-Strings (PHP-Modul mbstring)	117
4.3.6	Reguläre Ausdrücke	118
4.3.7	Grapheme und die Länge eines Strings	119
4.3.8	Zusammenfassung	123
4.4	MySQL und Unicode	124
4.4.1	Unterstützte Kodierungen	124
4.4.2	Kodierung der Client-Verbindung	124
4.4.3	Sortierfolge und Textvergleich	125
4.4.4	Angaben der gewünschten Sortierfolge	126
4.4.5	Auswirkung auf Indexe	129
4.4.6	Fazit	130
4.5	Zusammenfassung	131

Teil II

Interaktive Websites: Kommunikation zwischen Client und Server ... 133

5 Interaktive Websites

5.1	JavaScript Object Notation (JSON)	136
5.1.1	JSON-Syntax	136
5.1.2	Datentypen	137

5.1.3	JSON & PHP	138
5.1.4	JSON parsen mit JavaScript	142
5.1.5	Zusammenfassung	146
5.2	JavaScript Object Notation with Padding (JSONP)	146
5.2.1	Funktionsweise	146
5.2.2	Sicherheit	148
5.2.3	Zusammenfassung	150
5.3	XMLHttpRequest	150
5.3.1	Basisfunktionalität (XHR Level 1)	150
5.3.2	Beispiel: Nachladen von Textinhalten	153
5.3.3	Beispiel: Nachladen von XML-Inhalten	156
5.4	Cross-Origin Resource Sharing (CORS)	159
5.4.1	Einfache CORS-Anfragen	159
5.4.2	Komplexe CORS-Abfragen (preflighted requests)	161
5.4.3	Mitsenden bzw. Akzeptieren von Cookies	162
5.4.4	Überblick über alle CORS-HTTP-Header	163
5.5	XMLHttpRequest - Level 2	164
5.5.1	Erweiterte Ereignissteuerung (XHR Level 2)	164
5.5.2	Neue Datentypen (XHR Level 2)	167
5.6	Navigation in AJAX-Anwendungen	172
5.6.1	URL-Fragmente und Hashbang (#!)	172
5.6.2	HTML5-Session-History	173
5.7	Zusammenfassung	176
6	Servergesteuerte Aktualisierung (Server Push)	177
6.1	Kontinuierliches Abfragen (Polling)	177
6.2	Long Polling und Server-sent Events (SSE)	178
6.2.1	Server-sent Events: Client-Seite	179
6.2.2	Server-sent Events: Protokoll	182
6.2.3	Server-sent Events: Serverseite	183
6.2.4	Analyse der notwendigen Ressourcen	185
6.3	Eventbasierter PHP-Server	185
6.3.1	Eventbasierte Architektur	186
6.3.2	libevent	188
6.3.3	PHP-Server mit libevent	188
6.3.4	Beispiel: PHP-Server für Server-sent Events	194
6.3.5	Ressourcen und Geschwindigkeit	197
6.3.6	Zusammenfassung	197
6.4	WebSocket	198
6.4.1	Ablauf	198
6.4.2	Handshake	199
6.4.3	Datenübertragung mit Frames	202
6.4.4	Beispiel: Eventbasierter PHP-WebSocket-Server	206

6.4.5	Client-Seite	210
6.4.6	Fazit	216
6.5	Zusammenfassung	216

Teil III

Immer schneller: Leistungsstarke Webseiten 217

7 Frontend optimieren 219

7.1	Warum ist das Frontend wichtig?	219
7.2	Caching-Mechanismen im HTTP-Protokoll	222
7.2.1	Caching-Dauer begrenzen	222
7.2.2	Veränderliche Inhalte	226
7.2.3	Bedingungen für die Zwischenspeicherung	229
7.2.4	Kombinieren der HTTP-Header	230
7.2.5	Grenzen und Probleme	231
7.3	Auswirkungen des TCP-Protokolls	231
7.3.1	Aufbau einer TCP-Verbindung	231
7.3.2	Durchsatz einer TCP-Verbindung	232
7.3.3	Browser, Verbindungen und Datendurchsatz	234
7.4	Reduzieren der Anzahl der HTTP-Requests	235
7.4.1	Caching	235
7.4.2	Eliminieren von Umleitungen (HTTP Redirect)	236
7.4.3	Zusammenführen von CSS- und JavaScript-Dateien	237
7.4.4	CSS-Sprites	238
7.4.5	Kleine Bilder direkt einbinden (data-Protokoll)	240
7.5	Reduzieren der Größe der Ressourcen	241
7.5.1	Komprimierung bei der Übertragung	241
7.5.2	JavaScript	243
7.5.3	CSS	244
7.5.4	Bilder	244
7.5.5	HTML	245
7.6	Darstellungsgeschwindigkeit	245
7.6.1	Einbinden von Stylesheets	246
7.6.2	Einbinden von JavaScript	246
7.6.3	Dynamisches Nachladen von JavaScript	248
7.6.4	Minimieren von Darstellungsänderungen (Reflow)	249
7.7	Weitere Techniken	250
7.7.1	Flush early	250
7.7.2	Anzahl der Domänen (Sharding)	252
7.7.3	Cookies	252
7.8	HTML 5-Web Storage	253
7.8.1	Web Storage	254
7.8.2	Web Storage als Cache nutzen	258

7.9	Messen und Analysieren	262
7.9.1	Ladezeiten und HTTP-Wasserfall	262
7.9.2	Werkzeuge zur schnellen Analyse	264
7.9.3	Detailanalyse	265
7.10	Zusammenfassung	266
8	Caching von Inhalten	267
8.1	Einsatz und Ablauf von Caching	267
8.1.1	Zwischenspeichern mit einem Reverse Proxy	268
8.1.2	Zwischenspeichern in PHP	269
8.1.3	Geeignete Inhalte auswählen	270
8.2	Aktualisieren von gecachten Inhalten	273
8.2.1	Synchron aktualisieren	273
8.2.2	Asynchron aktualisieren	274
8.2.3	Zu bestimmten Zeitpunkten aktualisieren	274
8.2.4	Vor Ablauf aktualisieren	274
8.2.5	Bei Zugriff aktualisieren	275
8.3	Verhindern eines Cache-Sturms	275
8.4	Zusammenfassung	277
9	Varnish und Edge Side Includes	279
9.1	Installation und Administration	279
9.1.1	Installation vom Quellcode	279
9.1.2	Starten von Varnish	280
9.1.3	Administration	280
9.1.4	Konfigurationsdateien	282
9.2	Reverse Proxy	283
9.2.1	Festlegen eines Webservers	284
9.2.2	Cache-Eigenschaften	285
9.2.3	Mehrere Webserver für unterschiedliche URLs	286
9.2.4	Lastverteilung	288
9.2.5	Cookies und Caching	289
9.3	Edge Side Includes	290
9.3.1	Syntax	292
9.3.2	Konfiguration in Varnish	292
9.3.3	Beispiel	294
9.3.4	Alternative in JavaScript	296
9.4	Zusammenfassung	296
10	Alternative PHP Cache (APC)	297
10.1	Installation	297
10.2	Überblick und Konfiguration	298
10.2.1	Opcode-Cache	298

10.2.2	Benutzer-Cache	299
10.2.3	Konfiguration	299
10.3	Daten mit APC cachen	301
10.4	Ablauf von Daten in APC	303
10.5	Aktualisieren von Einträgen vor Ablauf	304
10.5.1	Übersicht	304
10.5.2	Sichtweise des aufrufenden Programms	305
10.5.3	CachedValue-Klasse	307
10.5.4	Auslesen des Eintrages (get-Methode)	308
10.5.5	Speichern des Eintrages (store-Methode)	310
10.5.6	Verhalten bei parallelen Zugriffen	310
10.6	APC anfüttern	312
10.7	Monitoring	313
10.7.1	APC-eigenes Skript (apc.php)	313
10.7.2	APC-Funktionen zum Monitoring	314
10.8	Zusammenfassung	316
11	Memcached	317
11.1	Funktionsweise	317
11.1.1	Übersicht und Prinzipien	317
11.1.2	Datenverteilung auf Server und konsistentes Hashing	319
11.1.3	Lebensdauer von Einträgen	320
11.2	Installation	321
11.2.1	Installieren des Servers	321
11.2.2	Installieren des PHP-Moduls	322
11.3	Daten cachen	324
11.3.1	Zu Memcached verbinden	324
11.3.2	Daten in Memcached speichern und abfragen	326
11.3.3	Kombinieren mehrerer Abfragen	328
11.4	Gruppieren von Werten	329
11.4.1	Beispielprogramm: Benutzerklasse	330
11.4.2	Beispielprogramm: Nachrichtenklasse	331
11.4.3	Beispielprogramm: Nachricht schreiben und speichern	331
11.4.4	Beispielprogramm: Asynchrones Auslesen aller Nachrichten	333
11.5	Aktualisieren von Einträgen	335
11.5.1	Überblick	335
11.5.2	CachedValue-Klasse	336
11.5.3	Auslesen eines Eintrages (get-Methode)	337
11.5.4	Neuberechnung eines Eintrages	338
11.5.5	Sichtweise des aufrufenden Programms	339
11.5.6	Verhalten bei parallelen Zugriffen	340
11.6	Ausfallsicherheit	341

11.7	Hinweise zum Praxiseinsatz	342
11.7.1	Auswahl der Server	342
11.7.2	Memcached und Sessions	343
11.7.3	Cachen des Fehlerfalls	343
11.7.4	Speichern von statischen Inhalten	344
11.7.5	Monitoring	344
11.8	Zusammenfassung	346

Teil IV

Datenbanken à la carte: MySQL und NoSQL	347
--	------------

12 MySQL für Fortgeschrittene	349
--	------------

12.1	Sakila-Beispieldatenbanken	349
12.2	Index-Strategien	351
12.2.1	Grundlagen	351
12.2.2	Verwenden von Indexen	353
12.2.3	Abdeckender Index (Covering Index)	356
12.2.4	Sortieren von Daten	357
12.2.5	Indexgröße	360
12.3	Optimieren von Abfragen	362
12.3.1	Abfragen von nicht benötigten Daten	362
12.3.2	Seitenweises Blättern mit LIMIT	363
12.3.3	Bereichsabfragen mit mehrspaltigen Indexen	364
12.3.4	Beispiel: JOIN optimieren	366
12.3.5	Zusammenfassung	368
12.4	Transaktionen	368
12.4.1	SQL-Transaktionen	369
12.4.2	Isolationsstufen von Transaktionen	371
12.4.3	MVCC und Lock-Mechanismen der Isolationsstufen	372
12.4.4	Transaktionen mit PHP	376
12.4.5	Fazit	378
12.5	Zusammenfassung	379

13 Redis	381
-----------------------	------------

13.1	Warum NoSQL?	381
13.2	Übersicht	384
13.3	Installation	385
13.3.1	Installation als Boot-Service	385
13.3.2	Installation des PHP-Moduls	386
13.4	Aufbauen der Verbindung zu Redis	387
13.4.1	Persistente Verbindungen	387
13.4.2	Optionen	387

13.5	Datentypen	388
13.5.1	Zeichenketten	388
13.5.2	Listen	395
13.5.3	Mengen	402
13.5.4	Sortierte Mengen	412
13.5.5	Hashes	420
13.6	Weitere Funktionen	423
13.6.1	Befehle zur Schlüsselverwaltung	424
13.6.2	Ablauf von Einträgen	425
13.6.3	Sortieren	426
13.7	Transaktionen und Pipelining	430
13.8	Persistieren der Datenbank	432
13.8.1	Snapshot der Datenbank	432
13.8.2	Speichern per Logdatei	433
13.9	Zusammenfassung	434
14	MongoDB	435
14.1	Installation	435
14.1.1	Mongo-Shell	436
14.1.2	Installation der PHP-Erweiterung	436
14.2	Datenbanken und Datenschema	437
14.2.1	Verbinden und Auswahl der Datenbank	437
14.2.2	Auswahl und Erzeugen von Kollektionen	438
14.2.3	Dokumente und Datentypen	439
14.2.4	Entwerfen des Datenschemas	441
14.3	CRUD-Operationen	443
14.3.1	Einfügen von Dokumenten	444
14.3.2	Fehlerbehandlung	445
14.3.3	Primäre Schlüssel	447
14.3.4	Finden von Dokumenten	448
14.3.5	Aktualisieren von Dokumenten (Update)	449
14.3.6	Aktualisieren verschachtelter Dokumente	452
14.3.7	Löschen von Dokumenten und Kollektionen	456
14.4	MongoDB-Abfragesprache	457
14.4.1	Operatoren	457
14.4.2	Verschachtelte Dokumente und Arrays	459
14.4.3	Komplexe Bedingungen (\$where)	461
14.4.4	Sortieren	463
14.4.5	Gruppieren	465
14.5	Indizieren von Feldern	468
14.5.1	Indexarten	469
14.5.2	Verwenden von Indexen	470
14.5.3	Analysieren von Abfragen mit explain()	471

14.6	Map/Reduce	475
14.6.1	Prinzip	476
14.6.2	Ablauf und Parameter	478
14.6.3	Inkrementelle Berechnung	482
14.7	Replikation und Verfügbarkeit	485
14.7.1	Funktionsweise	485
14.7.2	Anlegen einer Replikationsgruppe	485
14.7.3	Verwenden in PHP	486
14.8	Sharding und Skalierung	487
14.8.1	Funktionsweise	488
14.8.2	Sharding konfigurieren	488
14.8.3	Verwendung	489
14.8.4	Sharding-Schlüssel auswählen	490
14.9	Zusammenfassung	491
	Bildnachweise	493
	Index	495