

Storing Data

Thierry Sans

Modern Web Platform

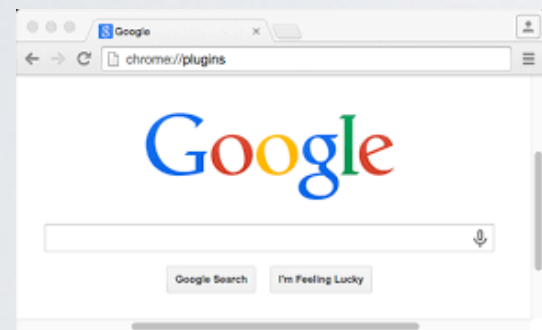
Client Side

Server Side

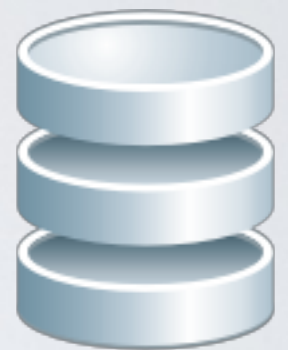


```
Macintosh HD -- top -- 80x24
Processes: 218 total, 2 running, 9 stuck, 199 sleeping, 901 threads  23:30:03
Load Avg: 1.40 1.75 1.00  CPU usage: 4.15% user, 4.40% sys, 91.44% idle
SharedLib: 164K resident, 0K data, 0K inshared.
MemRegions: 31278 total, 1892M resident, 117M private, 564M shared.
PhysMem: 588M used (113M wired), 18K unused
VM: 523G vsz, 1020M framework vsz, 0(0) swapins, 0(0) swapouts.
Networks: packets: 12105/8925K in, 11907/1964K out.
Disks: 88156/2285M read, 21235/425M written.

PID  COMMAND   %CPU  TIME    #TH  #PM  #PORT  MEM    PURG  CNPR  PGAR  PPID
592  screencap  0.0   00:00:02  7    0    55+   1932K  20K+  00    00    1
598  mdworker   0.0   00:00:01  3    0    44    2032K  00    00    590  1
598  mdworker   0.0   00:00:01  3    0    44    1572K  00    00    590  1
598  top        1.7   00:00:51  1/1   0    22+   2460K  00    00    580  584
584  bash       0.0   00:00:00  1    0    15    588K   00    00    584  583
583  login      0.0   00:00:01  3    1    28    1230K  00    00    583  482
574  auditd     0.0   00:00:00  2    0    25    508K   00    00    574  1
567  System Pre 0.0   00:03:23  3    0    270   39M   8364K  00    00    567  1
561  systemstat 0.0   00:00:01  2    1    19    1040K  00    00    561  1
560  com.apple.W 0.0   00:01:42  9    0    229   25M   00    00    560  1
558  com.apple.W 0.0   00:05:47  15   3    224   151M  1716K  00    00    558  1
555  bash       0.0   00:00:00  1    0    15    604K   00    00    555  554
554  login      0.0   00:00:01  3    1    28    1176K  00    00    554  482
550  bash       0.0   00:00:00  1    0    15    608K   00    00    550  549
```



Web API



Database

Why using a database

- Persistency
- Concurrency (avoid race conditions)
- Query
- Scalability

SQL vs NoSQL databases

Relational database (SQL database)

Data structure	tables and tuples
Query language	SQL
Inconvenient	inadequate for big data
Advantage	transactions (ACID)
Technology	<i>PostgreSQL, MySQL, MariaDB, SQLite, MSSQL</i>

NoSQL database

Data structure	key/value pairs
Query language	API style
Inconvenient	lack of consistency (non ACID)
Advantage	adequate for big data
Technology	<i>MongoDB, Redis, CouchDB</i>

ORM - Object Relational Mapping

➡ Mapping between (OOP) objects and the database structure

Examples

- *Sequelize for PostgreSQL, MySQL, MariaDB, SQLite*
- *Mongoose for MongoDB*

Connecting the REST API with a database

Do/Don't

- Do **retrieve selected elements only**
rather than retrieving an entire collection and filtering afterwards
- Do **define primary keys**
rather than relying on auto-generated ones
- Do **split data into different collections**
rather than storing list attributes
- Do **create join collections** whenever appropriate
(only for NoSQL database without join feature)

Retrieving collections with paginated results

- ➡ Only retrieve what you need from a potentially large collection

Examples

```
GET /messages[?page=0]
```

```
GET /messages?page=1
```

```
GET /messages[?max=100]
```

```
GET /messages?max=20
```