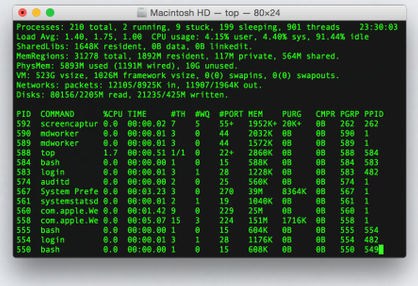
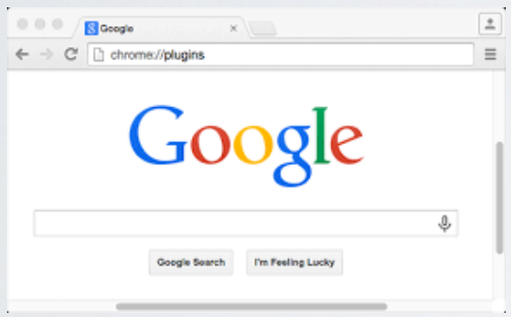
Storing Data

Thierry Sans

Modern Web Platform

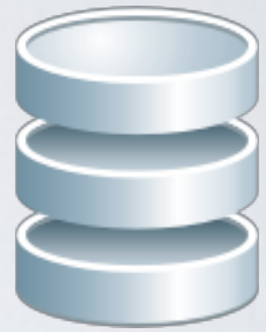
Client Side





Server Side

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	*PostgreSQL, MySQL, MariaDB, SQLite, MSSQL*

NoSQL database

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

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
```