

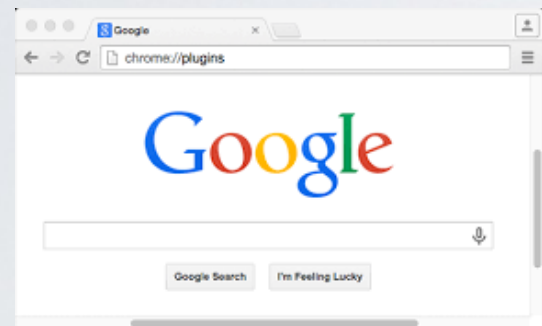
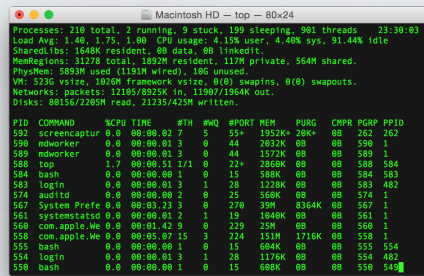
# (REST) Web API

Thierry Sans

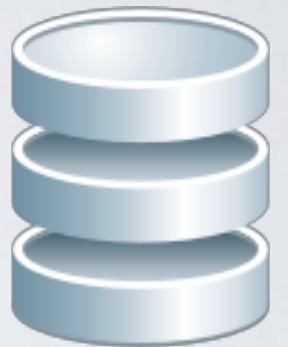
# Modern Web Platform

Client Side

Server Side



Web API



Database

➡ The server side is more or less of a **storage system**

# REST - Representational State Transfer

Design a remote API for a storage system by using HTTP

- **Function names** : method and URL
- **Function arguments** : URL and request body
- **Returned value** : status code and response body

# REST concepts

Mostly storage systems are mean to store

- **Collections** (or ressources)
- **Elements** that belongs to one or several collections

# Examples

	HTTP request	HTTP response
Create a new message	<code>POST /messages/ "Hello World"</code>	<code>200 "78"</code>
Get all messages	<code>GET /messages/</code>	<code>200 "['Hello world', ...]"</code>
Get a specific messages	<code>GET /messages/78/</code>	<code>200 "Hello World"</code>
Delete a specific messages	<code>DELETE /messages/78/</code>	<code>200 "success"</code>



# Relationships


Type	Example
one-to-one	<code>/users/sansthie/profile/firstname/</code>
one-to-many	<code>/users/sansthie/messages/89/</code>
many-to-many	<code>/users/sansthie/teams/8/ /teams/8/users/sansthie/</code>

# CRUD - manipulating data

Basic functions of persistent storage

- **C**reate
- **R**ead
- **U**ppdate
- **D**eleate

# Query methods

CRUD	HTTP	Collection	Element
Create	POST		Create a new element
	PUT	Replace the entire collection	Create (or replace if exists) a specific element
Read	GET	List all elements	Retrieve a specific element
Update	PATCH	Update some attributes of some elements	Update some attributes of a specific element
Delete	DELETE	Delete the entire collection	Delete a specific element



# Status codes

- <http://www.restapitutorial.com/lessons/httpmethods.html>

# Use of attributes

Query a subset of a collection : filter, page, range ...

```
GET /messages/?from=67&to=99
```