Web services; week1

Summary :

1/ What is a web service

2/ Why use it

3/ How does it work?

4/

5/

### 1/ What is a web service?

w3c : “a web service is a software system designed to support **interoperable** machine-to-machine interaction over a network”.

web service :

* it’s about all the devices that can access data : phone, server, computer…
* Request data : to whom?
* web service is 3rd party to which we ask data to be processed in a particular way
* make an interface (graphical or not)

### 2/ Why use it ?

* **interoperability :** any software system with a network access can be interfaced.
  + interface = what we use to communicate with each other (sound, english language…)
    - in our case: any web protocol
* modularity : the interface does not depend on data and their processing
  + we don’t care from where the data cames, as long as the interface doesn’t change
* evolutive :
  + you can add new features as long as it doesn’t change its main purpose

### 3/ How does it work?

Constraint 1 : Representational State Transfer (REST)

* it’s like a standart. it’s an architectural style that defines a set of constraints to be used for creating web services.

serialization : converting pure data to something that is readable by every computing system (XML, JSON..)

Who uses web services?

* developers (not any users)
* humans have to understand it

**CRUD**(create, read, update, delete) : methodology

* 4 basis functions of persistent storage
  + persistent storage = data which are theoretically always stored (unless, we ask the sys to delete them)
  + POST : create
  + GET : read
  + PUT : update
  + DELETE

### 4/ Conception

**Object :** data characterized by. Symbolizes data.

* data
* relations : when objects have related data/when objects are included in each other.
* properties

Object is what we work on in the case of web service. We need a tool to access it, and it’s called route.

**Route :**

* version : to keep track on the changes on the features that compose our software
* selection : to target a particular object
* name : human tool to target a specific data/object
* relations :
* options

**Action :**

* it’s a process that will either return a response or an error

How to make a route?

* we need a **version** : it will help the developer to know which interface he should use to contact the webservice.
  + In the url you have numbers that will help the developer to know where to look to know which routes are available.
* there always **only** should **names** : clear documentation.
  + not verbs.
* They should be **plural names** (if you forget if it’s single or plural…)
* Use options for selection. It helps organizing the data.

**Security :**

* API key : make API private
  + never in the url, need to pass through SSL (in HTTPS)
* OAuth2 token : a website A wants to access from facebook, if user agree, facebook create token and give it to website A
  + very strong
* HMAC : used by Amazon
  + public & private key