

Web API

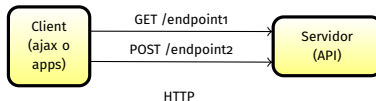
Interfícies d'usuari

Aleix Llusà Serra

Enginyeria de Sistemes TIC
Universitat Politècnica de Catalunya
<http://epsem.upc.edu>

28 d'abril de 2021

- Interfícies de programació d'aplicació (API) per a serveis web.
- Comunicacions HTTP de màquina a màquina. El client pot ser tant un navegador (AJAX, SSE...) com altres aplicacions o servidors web.
- Formats d'intercanvi típics: Json o XML.



- Quins són els **recursos** que té el servidor, és a dir la informació que es vol consultar.
- Com s'identifiquen i s'accedeix als recursos, s'anomenen **endpoints**. A HTTP directament són les URL.
- Quines **accions** es poden fer a cada recurs
- Quines **respostes** retornen les accions
- **Formats** d'intercanvi de dades, generalment Json o XML
- Alguns estàndards d'arquitectura:
 - XML-RPC
 - SOAP
 - Webdav
 - REST

Representational State Transfer (REST)

- Arquitectura d'estil de programació de Web API a sobre del protocol HTTP
- Serveis web amb REST API: **RESTful Web Services** (RWS)
- Especificacions de recursos, endpoints, accions, respostes i formats
- Cada RWS web defineix la seva pròpia REST API
- No protocol estàndard, no interoperables
- A diferència p.ex. de Webdav: protocol d'accés a continguts web

Requeriments REST per a implementar un RWS

Roy Fielding, PhD 2000

- Arquitectura client-servidor: Separar l'emmagatzematge de dades de la interfície d'usuari
- Comunicacions sense estat (stateless): No emmagatzemar estat al servidor entre diferents peticions. No crear sessió (excepcions com l'autenticació).
- Catxejable: Les respostes han d'indicar si es poden emmagatzemar temporalment en cache.
- Sistema de capes: Servidors estructurats de forma transparent. Arquitectura flexible i modular on hi poden haver intermediaris, endpoints que es processin en diferents servidors, etc. però client no ho percep.
- Codi a demanda (opcional): transferir lògica que el client pugui executar (p.ex. scripts JS)
- Interfície uniforme: basada en recursos, manipulacions, missatges autodescriptius i hipermèdia (HATEOAS).

- URL semàntica: endpoints seguint estructura conceptual dels recursos:
 - La jerarquia d'URL implica estructura.
`/app?est=pompeu&curs=iu` `/ests/pompeu/cursos/iu`
 - Els plurals indiquen col·lecció.
`/estudiant/pompeu` `/estudiants/pompeu`
 - Query string per a filtres en col·leccions.
`/estudiants?pag=30&ordre=alfabetic`
 - Els recursos s'especifiquen amb noms, sense verbs
`/recurs/edita` `/recurs/afegeix`
- Verbs HTTP: les accions són mètodes HTTP. Mapa CRUD:

POST	Create
GET	Read
PUT/PATCH	Update replace/modify
DELETE	Delete

- Codis de resposta HTTP: èxit de les peticions

200	Ok
201	Created
400	Bad Request
401	Unauthorized
500	Server error

- Json I XML: formats habituals d'intercanvi de dades. Permetre escollir:
Enviament: body segons capçalera **HTTP Content-Type**
Resposta: demanar al servidor mitjançant capçalera **HTTP Accept**
- Enllaços hipermedia: tipus relacions implementats amb URL.
Millor un enllaç i fer petició a REST API que retornar excés de dades
 - Llistat de continguts
`[{id:1,url:http://domini.test/continguts/1},{id:2,url:http://domini.test/continguts/2}]`
 - Amb paginació `{resultats:[...],page:1,self:http://domini.test/continguts?page=1,next:http://domini.test/continguts?page=2}`

Test:

- cURL – trobareu el paquet a Debian i Ubuntu
- Postman

Documentació:

- <https://sphinxcontrib-httpdomain.readthedocs.io>
- <https://openapis.org> <https://swagger.io>

Generalitat de Catalunya: Dades Obertes

EX: Eleccions al Parlament de Catalunya 2021. Per primera vegada s'ofereixen les dades de recompte a la plataforma de dades obertes

<http://governobert.gencat.cat/ca/detalls/noticia/noticia-publicacio-dades-electorals>.

Aquesta plataforma ofereix una API de dades d'accés obert de Socrata. Podeu veure'n la documentació per al recompte definitiu a <https://dev.socrata.com/foundry/analisi.transparenciacatalunya.cat/7qqj-4kvi>

Tots els resultats per municipis, Json o CSV:

GET <https://analisi.transparenciacatalunya.cat/resource/7qqj-4kvi.json>

GET <https://analisi.transparenciacatalunya.cat/resource/7qqj-4kvi.csv>

Exemple: resultats totals de Manresa

GET https://analisi.transparenciacatalunya.cat/resource/7qqj-4kvi.json?nivell=MU&nom_municipi=Manresa

```
[{
  "nivell": "MU", "codi_circumscripcio": "8",
  "codi_municipi": "113", "nom_municipi": "Manresa",
  "codi_comarca": "7", "nom_comarca": "Bages",
  "cens_electoral": "51872",
  "votants": "27397", "votants_1": "52.82",
  "abstenci": "24475", "abstenci_1": "47.18",
  "vots_nuls": "311", "vots_nuls_1": "1.14",
  "vots_en_blanc": "210", "vots_en_blanc_1": "0.77",
  "vots_a_candidatures": "26876", "vots_a_candidatures_1": "98.1",
  "vots_v_lids": "27086", "vots_v_lids_1": "98.86",
  "vots_psc": "4248", "psc": "15.68",
  "vots_erc": "6424", "erc": "23.72",
  "vots_jxcat": "8126", "jxcat": "30",
  "vots_vox": "1499", "vox": "5.53",
  "vots_cup_g": "2322", "cup_g": "8.57",
  "vots_ecp_pec": "1268", "ecp_pec": "4.68",
  "vots_cs": "884", "cs": "3.26",
  "vots_pp": "623", "pp": "2.3",
  "vots_pdecat": "1112", "pdecat": "4.11",
  "vots_recortes_cero_gv_m": "97", "recortes_cero_gv_m": "0.36",
  ...
}]
```

The Movie Database **TMDB**

<https://themoviedb.org> és una interfície web a una base de dades comunitària de pel·lícules i sèries de TV. A més, la mateixa informació també s'ofereix com a REST API a <https://api.themoviedb.org/3>, podeu llegir-ne la documentació a <https://developers.themoviedb.org> (en cas que la vulgueu utilitzar heu de registrar un usuari).

A continuació podeu veure alguns exemples de com es documenta aquesta REST API (sense la part d'autenticació).

GET movie details

Obtenir informació d'una pel·lícula

Sol·licitud

```
GET /movie/{movie_id:int}  
Accept: application/json
```

Paràmetres URL:

- movie_id (integer) L'identificador de la pel·lícula

Codis estat:

- 200 OK
- 401 Unauthorized: Invalid API Key
- 404 NotFound: The movie could not be found

Exemple de resposta

```
HTTP/1.1 200 OK  
Content-Type: application/json  
{  
  "id": 550,  
  "title": "Peli",  
  "backdrop_path": "/fCayJr.jpg",  
  ...  
}
```

POST rate movie

Valorar una pel·lícula
Sol·licitud

```
POST /movie/{movie_id}/rating
Content-Type: application/json
Accept: application/json
{
  "value": 8.5
}
```

Paràmetres URL:

- movie_id (integer) L'identificador de la pel·lícula

Paràmetres JSON:

- value (number) La valoració sobre 10

Codis estat:

- 201 Created
- 401 Unauthorized: Invalid API Key
- 404 NotFound: The movie could not be found

Exemple de resposta

```
HTTP/1.1 201 Created
Content-Type: application/json
{
  "status_code": 1,
  "status_message": "Success."
}
```

GET search movies

```
GET /search/movie?query=text:string[&page=pagina:int]
```

```
Accept: application/json
```

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{"page": 1,  
  "results": [  
    {  
      "id": 1,  
      "title": "Peli",  
      ...  
    },  
    {  
      "id": 2,  
      "title": "Peli 2",  
      ...  
    }, ...],  
  "total_results": 14,  
  "total_pages": 5}
```

Paràmetres query string:

- query (string) El text de cerca (requerit)
- page (integer) Pàgina de batch

Resposta JSON:

- page (integer) Pàgina actual
- total_pages (integer)
- total_results (integer)
- results (array) La llista de resultats amb cada pel·lícula (id, title, etc.)