



Guia de disseny API REST

30 de març de 2021

Serveis d'Administració Electrònica en el Govern de les Illes Balears

Lot2 (Serveis de tramitació d'expedients electrònics)

Control de versions del document

Data	Autor	Versió	Canvis
30/03/2021	Límit technologies	1.0	Versió inicial

ÍNDEX

Introducció	4
Normes bàsiques	4
Utilitzar JSON	4
Utilitzar noms	4
Utilitzar plurals	4
Controlar errors i retornar HTTP Status Code adequats	5
Utilitzar CRUD	7
CRUD:	7
Anidar recursos	9
Afegir Hipermedia	9
Afegir filtres, ordenació i paginació	9
Cache	9
Versionat	10
Documentació	10
Exemple de API	12

1. INTRODUCCIÓ

Aquest document pretén ser una guia bàsica sobre les normes que s'han de seguir en la definició i implementació de les API REST de l'aplicació HELIUM, i concretament en el pas a microserveis.

2. NORMES BÀSIQUES

2.1. UTILITZAR JSON

Utilitzarem el Content-Type **application/json** com a format per defecte en la transmissió de dades de la API

2.2. UTILITZAR NOMS

Utilitzar noms enlloc de verbs en las rutes dels endpoints, per tal de desacoblar els recursos dels mètodes o accions a dur a terme.

Els verbs s'ajusten a les operacions CRUD:

- **GET:** Utilitzat per a obtenir un o més recursos
- **POST:** Utilitzat per a crear un nou recurs
- **PUT:** Utilitzat per a modificar un recurs existent
- **PATCH:** Utilitzat per a modificar parcialment un recurs existent (modifica les propietats informada, deixant la resta sense modificar.
- **DELETE:** Utilitzat per eliminar un recurs

ex.

Acció a realitzar	Operació	Correcte	Incorrecte
Consulta del recurs articles	GET	/articles	/getArticles
Creació d'un article	POST	/articles	/createArticle

2.3. UTILITZAR PLURALS

Utilitzarem el nom del recurs en plural en les rutes dels endpoints.

2.4. CONTROLAR ERRORS I RETORNAR HTTP STATUS CODE ADEQUATS

Sempre es retornaran els HTTP Status Code definits per defecte en cada aplicació. En cas de no saber quins codis retornar, es consultarà a l'arquitecte de l'aplicació.

Els codis HTTP més comuns són els següents:

Codi	Text	Descripció
1XX	Informació	
101	Switching Protocol	Si es mou el client d'HTTP a un altre protocol menys habitual
2XX	Éxit	
200	Ok	Resposta estàndard per peticions correctes.
201	Created	La petició ha estat completada creant un nou recurs.
202	Accepted	La petició ha estat acceptada però encara no s'ha completat (p. ex. creacions retardades)
204	No Content	El servidor ha processat correctament la petició, però no retorna cap contingut.
3XX	Redirecció	
301	Moved Permanently	Aquesta i totes les peticions futures haurien de ser dirigides a la URL donada
304	Not Modified	Indica que la petició a la URL no ha estat modificada des que va ser requerida per última vegada. La sol·licitud no ha modificar res, p. ex. PUT / PATCH
307	Temporary Redirect	Es tracta d'una redirecció que hauria d'haver estat aquí amb una altra URI, però encara es pot processar amb la URI proporcionada
4XX	Error de client	
400	Bad Request	La petició no pot ser completada degut a un error de sintaxi.
401	Unauthorized	Similar a 403, però usat específicament quan la autenticació és necessària i ha fallat. Token incorrecte.
403	Forbidden	La petició era legal però el servidor refusa respondre.

404	Not Found	La petició no pot trobar el recurs sol·licitat però pot estar disponible en un futur.
405	Method Not Allowed	Petició realitzada a una URI usant un mètode de sol·licitud no suportat per l'anomenada URI.
408	Request Timeout	El temps d'espera màxim del servidor s'ha esgotat esperant la petició.
409	Conflict	No es pot realitzar l'acció, generalment en un PUT a causa de conjunts de dades incompatibles (penseu en un conflicte GIT que s'ha de corregir manualment abans de combinar)
410	Gone	Un recurs o element que existia anteriorment s'ha suprimit permanentment i ja no està disponible. Si es desconeix si la supressió és permanent o no, o si l'element existia, s'hauria d'utilitzar el codi 404
413	Request entity too long	El contingut que es demana és massa gran, i s'ha de fragmentar o reenviar en un format més curt / compacte
414	Request URI too long	L'URI proporcionada és massa llarga i no és acceptable pel servidor
415	Unsupported Media Type	El media type sol·licitat no s'admet. Per exemple, si l'usuari intenta enviar una sol·licitud mitjançant XML, però només es dona suport a JSON
429	Too many requests	El client ha enviat massa sol·licituds al servidor en el període de temps donat
5XX	Error de servidor	
500	Internal server error	El servidor ha tingut un error inesperat i no ho ha estat capaç de realitzar l'acció sol·licitada
501	Not Implemented	S'utilitza quan el client envia una sol·licitud que el servidor és incapaç de respondre degut a la manca de funcionalitat
503	Service Unavailable	El servidor no pot respondre temporalment a la sol·licitud (per exemple, a causa de manteniment) i el client hauria de tornar-ho a provar aviat.

2.5. UTILITZAR CRUD

CRUD:

- **GET:** Utilitzat per a obtenir un o més ítems, sense manipular-los

ex. /items?queryString

Retorna:

- 200: OK. Petició correcte
- 204: No content. Petició correcte, però no retorna cap resultat
- 400: Bad input parameter. Paràmetres de la consulta incorrectes

- **POST:** Utilitzat per a crear un nou ítem

ex. /items

Retorna:

- 201: Created. Petició correcte
- 400: Bad input parameter. Paràmetres de la consulta incorrectes

Retornarà el link a l'element en la capçalera Location.

- **PUT:** Utilitzat per a modificar un ítem existent

ex. /items/1

Retorna:

- 204: No content. Ítem modificat, però no retorna informació
- 304: Not modified. No s'han realitzat canvis
- 404: Not found. No s'ha trobat l'ítem a modificar
- 409: Conflict. Conflict amb les dades existents i les passades per modificar

El retornarà el link a l'element en la capçalera Location.

- **PATCH:** Utilitzat per a modificar parcialment un item existent (modifica les propietats informades, deixant la resta sense modificar.

ex. /items/1

Retorna:

- 204: No content. Item modificat, però no retorna informació
- 304: Not modified. No s'han realitzat canvis
- 404: Not found. No s'ha trobat l'item a modificar
- 409: Conflict. Conflict amb les dades existents i les passades per modificar

El retornarà el link a l'element en la capçalera Location.

- **DELETE:** Utilitzat per eliminar un item

ex. /items/1

Retorna:

- 200: Ok. Item eliminat
- 404: Not found

El retornarà el link a l'element en la capçalera Location.

- **OPTIONS:** Utilitzat per consultar quins mètodes (verbs) hi ha disponibles per el recurs.

ex. /items
 /items/1

Retorna:

- 200: Ok. Retorna informació addicional al body
- 204: No content. No retorna informació addicional al body
- 405: Method not allowed. Si no s'implementa el mètode Options

Exemple de resposta:

```
HTTP/1.1 204 No Content
Allow: OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE
Date: Tue, 30 Mar 2021 02:54:04 GMT
Connection: keep-alive
```


2.6. ANIDAR RECURSOS

Quan es volen obtenir recursos que pertanyen a un altre recurs, la ruta es formarà de forma anidada.

ex. /categoria/{categoriald}/articles

2.7. AFEGIR HIPERMEDIA

S'utilitzarà HATOAS per a afegir links (hipervincles) d'accés als diferents recursos relacionats amb el recurs que s'estigui servint.

ex.

```

{
  "links": [{
    "rel": "self",
    "href": "http://exemple.es/app/api/v1/articles/12"
  }, {
    "rel": "categoria",
    "href": "http://exemple.es/app/api/v1/categories/5"
  }],
  "id": "12",
  "categoriald": "5",
  "nom": "article",
  "import": 25
}
```

2.8. AFEGIR FILTRES, ORDENACIÓ I PAGINACIÓ

En les peticions on es puguin obtenir llistes, intentarem afegir filtres amb RSQL, i paràmetres de paginació i ordenació.

ex. <http://exemple.es/articles?nom=article&pagina=0&mida=20&ordre=nom,desc>

2.9. CACHE

Sempre que sigui possible utilitzarem la caché per evitar càlculs innecessaris.

Es pot usar caché en servidor, i també utilitzar les capçaleres de la petició per a utilitzar caché en client.

ex. Cache-Control: public, max--age=3600 (expira en 1 hora)

2.10. VERSIONAT

Mantindrem versions de les APIs.

Per aquest motiu afegirem /v1/, /v2/, ... a l'inici de la ruta de la API.

Ex. /v1/articles

2.11. DOCUMENTACIÓ

S'utilitzarà OpenAPI 3.0 per a definir les APIs, i la seva corresponent documentació.

Concretament es farà ús de l'eina web SwaggerHub (<https://app.swaggerhub.com>) per a definir les APIs.

Per a cada una dels mètodes de la API s'especificaran amb el màxim detall:

- **Tags:** Apartats en els que es dividirà la documentació
- **OperationId:** Identificador únic de cada un dels mètodes
- **Summary:** Descripció curta del mètode
- **Description:** Descripció llarga del mètode. Per a aquesta descripció es pot utilitzar el llenguatge Markdown.
- **Parameters:** Els paràmetres que pot rebre el mètode. Hi ha 4 tipus de paràmetres (path, query, header i cookie), però n'utilitzarem bàsicament 2:
 - path: són els paràmetres que estan inclosos dins la pròpia ruta de la petició.
P. ex. en /items/{id}, id és el paràmetre.
 - query: són els query paràmetres que s'afegeixen a peticions get.
P. ex en /items?nom=valor, nom és el paràmetre.

Sempre que els paràmetres s'utilitzin en més d'un mètode es definiran en l'apartat components/parameter, i d'aquesta manera es podran reaprofitar en els diferents mètodes.

- **RequestBody:** L'especificació del contingut que s'envia en el body de la petició (se s'envia).
- **Responses:** Les respostes que pot retornar el mètode.
S'indica el HTTP status code, seguit de tota la informació rellevant de la resposta.

Sempre que sigui possible es definiran els recursos en l'apartat components/schemas, i d'aquesta manera es podran reaprofitar en els diferents mètodes.

Sempre que sigui possible es definiran validacions sobre els paràmetres o esquemes de dades definides i exemples adequat que facilitin la entenibilitat a l'usuari.

Generalment per a definir l'exemple, s'inclourà un atribut exemple en cada propietat de l'objecte que s'està definint.

Ex.

schema:

Item:

type: object

properties:

jndiDataSource:

type: string

maxLength: 255

example: java:/es.caib.example.db

Per més informació de com especificar cada un dels elements es pot consultar la documentació a: <https://swagger.io/specification/>

3. EXEMPLE DE API

<https://app.swaggerhub.com/apis-docs/josepg-limit/heliumms-domini-service/1.0.0#/>

Disseny Operacions securitzades disponibles per usuaris dissenyadors		▼
GET	/v1/dominis	Cerca de dominis
POST	/v1/dominis	Crea un nou domini
GET	/v1/dominis/{dominiId}	
PUT	/v1/dominis/{dominiId}	
PATCH	/v1/dominis/{dominiId}	
DELETE	/v1/dominis/{dominiId}	
Consulta Operacions disponibles per tots els usuaris		▼
GET	/v1/dominis	Cerca de dominis
GET	/v1/dominis/{dominiId}	
GET	/v1/dominis/{dominiId}/resultats	Consulta un domini

Fitxer YML:

```
openapi: 3.0.3
servers:
  - url: http://pendent.de.definir.es
    description: Servidor de desenvolupament
info:
  description: Microservei de Dominis de HeliumMS
  version: "1.0.0"
  title: HeliumMS Dominis API
  contact:
    name: Limit Technologies
    email: limit@limit.es
tags:
  - name: Disseny
    description: Operacions securitzades disponibles per usuaris dissenyadors
  - name: Consulta
    description: Operacions disponibles per tots els usuaris
paths:
  /v1/dominis:
    get:
      tags:
        - Disseny
        - Consulta
      operationId: listDominisV1
      summary: Cerca de dominis
      # Descripció amb markdown :)
```

```
description: |
  Obté una pàgina de la cerca dels **dominis** del sistema.

  La cerca pot rebre paràmetres per
  - ordenar
  - paginar
  - filtrar (utilitzant sintaxi rsq1)
parameters:
  - $ref: '#/components/parameters/FiltreParam'
  - $ref: '#/components/parameters/TipusExpedientParam'
  - $ref: '#/components/parameters/TipusExpedientPareParam'
  - $ref: '#/components/parameters/PageNumberParam'
  - $ref: '#/components/parameters/PageSizeParam'
  - $ref: '#/components/parameters/SortParam'
responses:
  '200':
    description: Ok
    content:
      application/json:
        schema:
          $ref:
            '#/components/schemas/DominiPagedList'
  '204':
    description: No content
  '400':
    description: Bad input parameter
post:
  tags:
    - Disseny
  operationId: createDominiV1
  summary: Crea un nou domini
  description: Adds an item to the system
  requestBody:
    required: true
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/Domini'
  description: Domini a crear
  responses:
    '201':
      description: Domini created
      headers:
        Location:
          description: Adressa del domini creat
          schema:
            type: string
            format: uri
            example:
              http://example.com/v1/dominis/{assignedIdValue}
```

```

    '400':
      description: Bad request
    '409':
      description: Conflicte. Ja existeix el domini
/v1/dominis/{dominiId}:
  get:
    tags:
      - Disseny
      - Consulta
    operationId: getDominiV1
    parameters:
      - $ref: '#/components/parameters/DominiIdPathParam'
    responses:
      '200':
        description: Ok
      '404':
        description: Not found
  put:
    tags:
      - Disseny
    operationId: updateDominiV1
    parameters:
      - $ref: '#/components/parameters/DominiIdPathParam'
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/Domini'
    responses:
      '204':
        description: Domini actualitzat
      '404':
        description: Not found
      '409':
        description: Conflicte
      '#304':
        # description: 'Domini no modificat. Els atributs passats
coinciden amb els actuals del domini'
  patch:
    tags:
      - Disseny
    operationId: patchDominiV1
    parameters:
      - $ref: '#/components/parameters/DominiIdPathParam'
    requestBody:
      required: true
      content:
        application/json:
          schema:

```

```

    $ref: '#/components/schemas/Domini'
  responses:
    '204':
      description: Domini actualitzat
    '404':
      description: Not found
    '409':
      description: Conflicte
    '#304':
      # description: 'Domini no modificat. Els atributs passats
coincidien amb els actuals del domini'
  delete:
    tags:
      - Disseny
    operationId: deleteDominiV1
    parameters:
      - $ref: '#/components/parameters/DominiIdPathParam'
    responses:
      '200':
        description: Ok
      '404':
        description: Not found
/v1/dominis/{dominiId}/resultats:
  get:
    tags:
      - Consulta
    summary: Consulta un domini
    operationId: consultaDominiV1
    parameters:
      - $ref: '#/components/parameters/DominiIdPathParam'
      - name: identificador
        in: query
        description: identificador del domini al sistema remot
        required: false
        schema:
          type: string
      - name: parametres
        in: query
        schema:
          type: string
    responses:
      '200':
        description: 'Ok'
        content:
          application/json:
            schema:
              $ref:
                '#/components/schemas/ResultatDomini'
components:
  openIdConnectUrl: https://example.com/.well-known/openid-configuration
```

schemas:

Domini:

type: object

required:

- codi
- nom

properties:

id:

type: integer
format: int64
readOnly: true
example: 1234

codi:

type: string
maxLength: 64
example: codi_domini

nom:

type: string
maxLength: 255
example: nom_domini

tipus:

type: string
enum:

- CONSULTA_SQL
- CONSULTA_WS
- CONSULTA_REST

example: CONSULTA_REST

url:

type: string
format: uri
maxLength: 255
example: http://example.com/nom_domini

tipusAuth:

type: string
enum:

- NONE
- HTTP_BASIC
- USERNAME_TOKEN

example: HTTP_BASIC

origenCredencials:

type: string
enum:

- ATRIBUTS
- PROPERTIES

example: PROPERTIES

usuari:

type: string
maxLength: 255
example: codi_usuari

contrasenya:


```

    type: string
    format: password
    maxLength: 255
    example: password
  sql:
    type: string
    maxLength: 1024
    example: select camp_1, camp_2, camp_3 from taula where
camp_4 = :parametre
  jndiDatasource:
    type: string
    maxLength: 255
    example: java:/es.caib.example.db
  descripcio:
    type: string
    maxLength: 255
    example: Text descriptiu del domini
  cacheSegons:
    type: integer
    description: En cas de posar el valor 0 ???
  timeout:
    type: integer
    description: En cas de posar el valor 0 ???
  ordreParams:
    type: string
    maxLength: 255
  entorn:
    type: integer
    format: int64
    description: Identificador de l'entorn al que pertany
el domini
  expedientTipus:
    type: integer
    format: int64
    description: Identificador del tipus d'expedient al que
pertany el domini. Si el domini és global aquest identificador serà null.
  ResultatDomini:
    type: array
    items:
      $ref: '#/components/schemas/ParellaCodiValor'
  ParellaCodiValor:
    properties:
      codi:
        type: string
      valor:
        type: string
    type: object
  DominiList:
    type: array
    items:

```

```

    $ref: '#/components/schemas/Domini'
  DominiPagedList:
    type: object
    properties:
      content:
        $ref: '#/components/schemas/DominiList'
      allof:
        - $ref: '#/components/schemas/PagedResponse'
  PagedResponse:
    type: object
    properties:
      pageable:
        $ref: '#/components/schemas/PagedResponse_pageable'
      totalPages:
        type: integer
        format: int32
      last:
        type: boolean
      totalElements:
        type: integer
        format: int32
      size:
        type: integer
        format: int32
      number:
        type: integer
        format: int32
      numberOfElements:
        type: integer
        format: int32
      sort:
        $ref:
          '#/components/schemas/PagedResponse_pageable_sort'
      first:
        type: boolean
  PagedResponse_pageable:
    type: object
    properties:
      sort:
        $ref:
          '#/components/schemas/PagedResponse_pageable_sort'
      offset:
        type: integer
        format: int32
      pageNumber:
        type: integer
        format: int32
      pageSize:
        type: integer
        format: int32

```

```

    paged:
      type: boolean
    unpaged:
      type: boolean
  PagedResponse_pageable_sort:
    type: object
    properties:
      sorted:
        type: boolean
      unsorted:
        type: boolean
  parameters:
    DominiIdPathParam:
      name: dominiId
      in: path
      description: 'Identificador del domini'
      required: true
      style: simple
      schema:
        type: integer
        format: int64
    FiltreParam:
      name: filtre
      in: query
      description: cadena amb format rsq1 per a definir el filtre a
aplicar a la consulta
      required: false
      schema:
        type: string
    TipusExpedientParam:
      name: expedientTipusId
      in: query
      description: Identificador del tipus d'expedient al que pertany el
domini
      required: false
      schema:
        type: integer
        format: int64
    TipusExpedientPareParam:
      name: expedientTipusPareId
      in: query
      description: Identificador del tipus d'expedient pare al que pertany
el domini (en cas d'herència)
      required: false
      schema:
        type: integer
        format: int64
    PageNumberParam:
      name: page
      in: query

```

```
description: número de pagina a retornar en cas de desitjar
paginació
required: false
schema:
  type: integer
  format: int32
  minimum: 0
  default: 0
PageSizeParam:
  name: size
  in: query
  description: mida de la pàgina a retornat en cas de desitjar
paginació
required: false
schema:
  type: integer
  format: int32
  minimum: 0
  # maximum: 100
  default: 25
SortParam:
  name: sort
  in: query
  description: ordre a aplicar a la consulta
  required: false
  schema:
    type: string
```