



Generalitat de Catalunya
**Centre de Telecomunicacions
i Tecnologies de la Informació**

Estandards de Nomenclatura v1.1

Juliol 2026

Índex de Continguts

1	Introducció.....	3
1.1	APIs	3
1.1.1	Dades generals	4
1.1.2	Vies d'accés (Path)	5
1.1.3	Definicions/Components	8
1.1.4	Propietats	9
1.2	Productes.....	11

1 Introducció

En aquest document, es defineixen els estàndards de nomenclatura definits en CTTI per a diverses seccions de la definició d'un API i un producte.

Moltes d'aquestes convencions es validen de manera automàtica al pipeline corporatiu del SIC. La compuerta de qualitat del pipeline executa dos rulesets corporatius contra els artefactes del proveïdor abans de la publicació al catàleg:

- **sic-corporate-api** (17 regles) sobre el document OpenAPI 2.0/3.0 de l'API.
- **sic-corporate-product** (26 regles) sobre el descriptor de Producte d'IBM API Connect.

Cada convenció validada d'aquest document té associada una **severitat funcional**:

- **error** — Incompliment bloquejant: el pipeline atura el desplegament fins que el proveïdor ho corregeixi. En SIC+ això impedeix el merge de la Pull Request.
- **warn** — Recomanació no bloquejant: el pipeline registra l'avís, però l'execució continua.

La descripció funcional completa de cada regla, amb els seus criteris d'acceptació i exemples, està al **Disseny funcional (ERQ) corporatiu dels rulesets sic-corporate-api i sic-corporate-product**. Aquest document se centra en les convencions transversals; el ERQ conté el catàleg exhaustiu de regles.

1.1 APIs

Per començar, es tindran les següents regles de nomenclatura a l'hora de dissenyar noves APIs:

- **Especificació de les APIs:** Les APIs s'especificaran i documentaran d'acord amb el format OpenAPI, sent la recomanació l'ús d'OpenAPI 3.0 sobre OpenAPI 2.0/Swagger.
- **Esquemes JSON:** S'han de tenir en compte les consideracions següents a l'hora de generar els esquemes d'entrada i sortida de les APIs:
 - Les variables es generaran en format lowerCamelCase.
 - Les dades han de ser llegibles per humans.
 - La definició de dades s'ha de mostrar com a entitats o subentitats funcionals.
- **Verbs:** S'ha de seguir l'estàndard HTTP per a l'ús de verbs sempre que sigui possible per complir amb l'estàndard RESTful. Tots els verbs HTTP són permesos si són usats de manera correcta.
- **URI:** Les URIs es definiran en minúscules seguint la notació kebab-case. En CTTI, s'aplica l'estructura següent:

Estandards de Nomenclatura

https://{Hostname}/{Nom d' Organització}/{Catàleg}/{via d' accés base}/{via-de-accés} (la via d'accés base inclou el codi d'aplicació i **és validada per les pipelines del SIC**).

- **Codis d' estat HTTPS:** Totes les trucades a l' API retornaran un codi d' estat seguint l' especificació de codis marcada pel protocol HTTP.

1.1.1 Dades generals

Les regles de nomenclatura per a les dades generals i bàsiques d' una API són:

- **Títol:** Nom de l' API. Ha de ser prou descriptiu. Ha de començar pel codi d' aplicació, cada paraula començar en majúscula i ha de contenir espais entre cada paraula en cas d' existir. **Exemple:** 3292 Gestor CSV.

És validat per les pipelines del SIC. Regles validadores: api-nam-01-title-starts-with-codi (error) i api-nam-01-bis-title-titlecase (warn).

- **Nom:** Nom intern de l' API. Serà igual que el Títol, amb el codi d' aplicació i totes les lletres en minúscules i amb guions en lloc d' espais. **Exemple:** 3292-gestor-csv.

És validat per les pipelines del SIC. Regles validadores: api-nam-02-xibmname-starts-with-codi (error) i api-nam-02-bis-xibmname-kebabcase (warn).

- **Versió:** Número de versió de l' API a implementar. Ha d' estar en format **Semantic Versioning 2.0** estricte: tres components numèrics **MAJOR.MINOR.PATCH**, sense etiquetes de pre-release (-alpha, -rc1, etc.) ni metadades (+build.123).

És validada per les pipelines del SIC. Regla validadora: api-ver-01-semver-strict (error).

- **Descripció:** Descripció de la funcionalitat que ofereix l' API amb suficient detall perquè sigui fàcilment entensible.

És validada per les pipelines del SIC. Regla validadora: api-doc-01-description-min-length (warn).

- **Host (només en OpenAPI 2.0/Swagger):** Indicarà el host on es troba l' API per ser executada. Es deixarà el valor per defecte **\$(catalog.host)**.

- **Via d'accés base (Base Path)/Servidors (Servers):** URL base que es concatenarà al host i al path de cada recurs de l'API. Es recomana que sigui un nom descriptiu general que englobi tota la funcionalitat d'aquesta API, hauria de incloure el codi de diàleg primer, ser unívoc de manera que no pugui generar cap tipus de solapament amb les URLs de la resta d'APIs desplegades i seguir les següents regles de nomenat.

És validada per les pipelines del SIC. Regles validadores del Base Path: api-url-01-basepath-starts-with-codi (error) i api-url-02-basepath-no-trailing-slash (error).

Estandards de Nomenclatura

Regla	Exemple	Observacions
Usar substantius per representar recursos. Usar verbs només si no és possible usar substantius.	/3292/clients	Un recurs representa una entitat, i aquesta és la raó per la qual convé fer servir substantius. Els verbs es refereixen a accions, que s'utilitzen per factoritzar accions HTTP.
Usar guió mitjà en utilitzar paraules compostes (com a màxim 3 paraules).	/3292/clients-interns	
No usar guions baixos.	<u>Correcte:</u> /3292/clients-interns <u>Incorrecte:</u> /3292/clients_interns	
Usar lletres minúscules únicament.	/3292/clients	D'acord amb RFC3986, les URL distingeixen entre majúscules i minúscules. L'ús de lletres majúscules per definir recursos fa que la clau d'URL sigui sensible, la qual cosa pot generar confusió en situacions particulars.

- **Llista d'esquemes (només en OpenAPI 2.0/Swagger):** Es defineix el protocol de transferència que utilitzarà l'API. S'utilitzarà **https**.
- **Consumeix (només en OpenAPI 2.0/Swagger):** Indica el tipus d'intercanvi d'informació que rebrà l'API. Es recomana que l'intercanvi entre els nous consumidors sigui sempre usant l'estàndard de JSON, per la qual cosa hauria de ser sempre **application/json**.
- **Produeix (només en OpenAPI 2.0/Swagger):** Indica el tipus d'intercanvi d'informació que retornarà l'API. En el nostre cas, es recomana que sigui sempre **application/json**, cosa que ens permetrà rebre les dades en format JSON.

1.1.2 Vies d'accés (Path)

Aquí es defineixen tots les regles de nomenclatura per a els paths que tindrà l'API. Per a cada path, s'afegiran els verbs que tindrà cada operació, així com els paràmetres d'entrada/sortida. La ruta completa de l'API estarà formada per:

https://{hostname}/{nom-organització}/{catàleg}/{via-de-accés-base}/{via-de-accés}

- Les dades de la URI s'han de trobar en notació **kebab-case**

Estandards de Nomenclatura

- Les dades de negoci en header o body s'hauran d'escriure en **lowerCamelCase**

Les vies d'accés han de seguir les regles de nomenclatura següents:

Regla	Exemple	Observacions
Usar substantius per representar recursos. Usar verbs només si no és possible usar substantius.	/gestionar/proveïdor	Un recurs representa una entitat, i aquesta és la raó per la qual convé fer servir substantius. Els verbs es refereixen a accions, que s'utilitzen per factoritzar accions HTTP.
Usar guió mitjà en utilitzar paraules compostes (com a màxim 3 paraules).	/gestionar/client-extern	
No usar guions baixos.	Correcte: /gestionar/client-extern/alta-expedient Incorrecte: /gestionar/client-extern/alta_expedient	
Usar lletres minúscules únicament.	/gestionar/client-intern	D'acord amb RFC3986, les URL distingeixen entre majúscules i minúscules. L'ús de lletres majúscules per definir recursos fa que la clau d'URL sigui sensible, la qual cosa pot generar confusió en situacions particulars.
El caràcter alfanumèric s'ha de fer servir amb moderació.	/consultar/informes/2023	Alguns caràcters no poden formar part d'una URL (per exemple, l'espai) i altres caràcters tenen un significat especial en una URL.
El nom del recurs sempre ha de començar amb un caràcter alfabètic.	/consultar/informes	Els caràcters alfanumèrics són segurs, cosa que significa que no es requereix codificació d'URL (convertir una cadena en un format d'URL vàlid).
Evitar l'ús d'extensions d'arxiu.	Correcte: /consultar/informes/2023/març Incorrecte: /consultar/informes/2023/març.pdf	

Dins de cada Via d' Accés, trobem aquests components:

- **Paràmetres**

Es definiran tots els paràmetres necessaris a l' API. Permet crear paràmetres que podran ser usats en diferents operacions. Es recomana que segueixin les següents regles de nomenclatura:

- Usar substantiu com a nom de paràmetre.
- Usar lowerCamelCase.
- Limitar els query params a 7. D'aquesta manera, evitem que la sol·licitud assoleixi el límit d'URI de 2048 caràcters. Si la sol·licitud té més de 7 que 100 euros, utilitzar POST al seu lloc.

- **Operacions**

Poden ser de diferents tipus (Verbs).

- **Verbs**
 - **GET:** Demana una representació d'un recurs específic. Les peticions que fan servir el mètode GET només han de recuperar dades.
 - **HEAD:** Demana una resposta idèntica a la d'una petició GET, però sense el cos de resposta.
 - **POST:** S' utilitza per trametre una entitat a un recurs en específic, causant sovint un canvi en l' estat o efectes secundaris en el servidor.
 - **PUT:** Reemplaça totes les representacions actuals del recurs de destinació amb la càrrega útil de la petició.
 - **DELETE:** Esborra un recurs específic.
 - **PATCH:** És utilitzat per aplicar modificacions parcials a un recurs.

Les operacions inclouen la informació següent:

- **Resum:** S' inclourà un breu resum de l' operació.
- **Descripció:** S' inclourà una descripció detallada de l' operació.
- **Documentació externa:** Documentació externa de l' operació a la qual s' accedeix mitjançant URL.
- **Paràmetres:** S' afegiran tots i cadascun dels paràmetres utilitzats en l' operació. Es recomana que segueixin les següents regles de nomenclatura:
 - Usar substantiu com a nom de paràmetre.
 - Usar lowerCamelCase.
 - Limitar els query params a 7. D'aquesta manera, evitem que la sol·licitud assoleixi el límit d'URI de **2048** caràcters. Si la sol·licitud té més de 7 que 100 euros, utilitzar POST al seu lloc.
- **requestBody (només en OpenAPI 3.0):** s' inclou la definició del cos d' entrada de la petició.
- **Respostes:** Codis de resposta HTTP que retornarà l' operació.

Estandards de Nomenclatura

- **Codis 1xx** – Associats a accions que s'informen a un navegador.
- **Codis 2xx** – Peticions rebudes, processades i respostes correctament.
És validat per les pipelines del SIC. Regles validadores (OAS 2 i OAS 3): api-oas-03-oas2-2xx-has-example (warn), aplicable a OAS2, i api-oas-03-oas3-2xx-has-example (warn), aplicable a OAS 3.
- **Codis 3xx** – Indica que es necessita realitzar una acció addicional com pot ser una redirecció.
- **Codis 4xx** – S'ha produït algun error del costat del navegador.
És validat per les pipelines del SIC. Regla validadora: api-oas-04-4xx-documented (warn).
- **Codis 5xx** – Error generat al costat del servidor.

1.1.3 Definicions/Components

Ens permet definir els models dels objectes que s'intercanviaran a les entrades i sortides de les APIs.

A l'hora de crear una nova Definició, tindrem els següents camps:

- **Nom:** Se debe utilizar lowerCamelCase.
- **Títol:** Títol de l' esquema.
- **Tipus:** Tipus de l'objecte a crear (string, booleà, array...).
- **Descripció:** Tot i que és opcional, és important que s'inclogui una descripció clara de l' objecte creat que permeti la seva ràpida identificació.
- **Propietats:** En aquest camp, s'afegeix l' estructura de camps que conforme aquesta definició. Ha de seguir les regles comentades anteriorment, que són:
 - Les variables es generaran en format lowerCamelCase.
 - Les dades han de ser llegibles per humans.
 - La definició de dades s' ha de mostrar com a entitats o subentitats funcionals.



×

Add object

Esquema

Definición

Referencia

Nombre de esquema (clave)

untitled

Título (opcional)

Tipo (opcional)

Seleccionar una opción

Descripción (opcional)

Escritura

Vista previa

TT

B

I

↺

🔗

“”

</>

📎

≡

≡

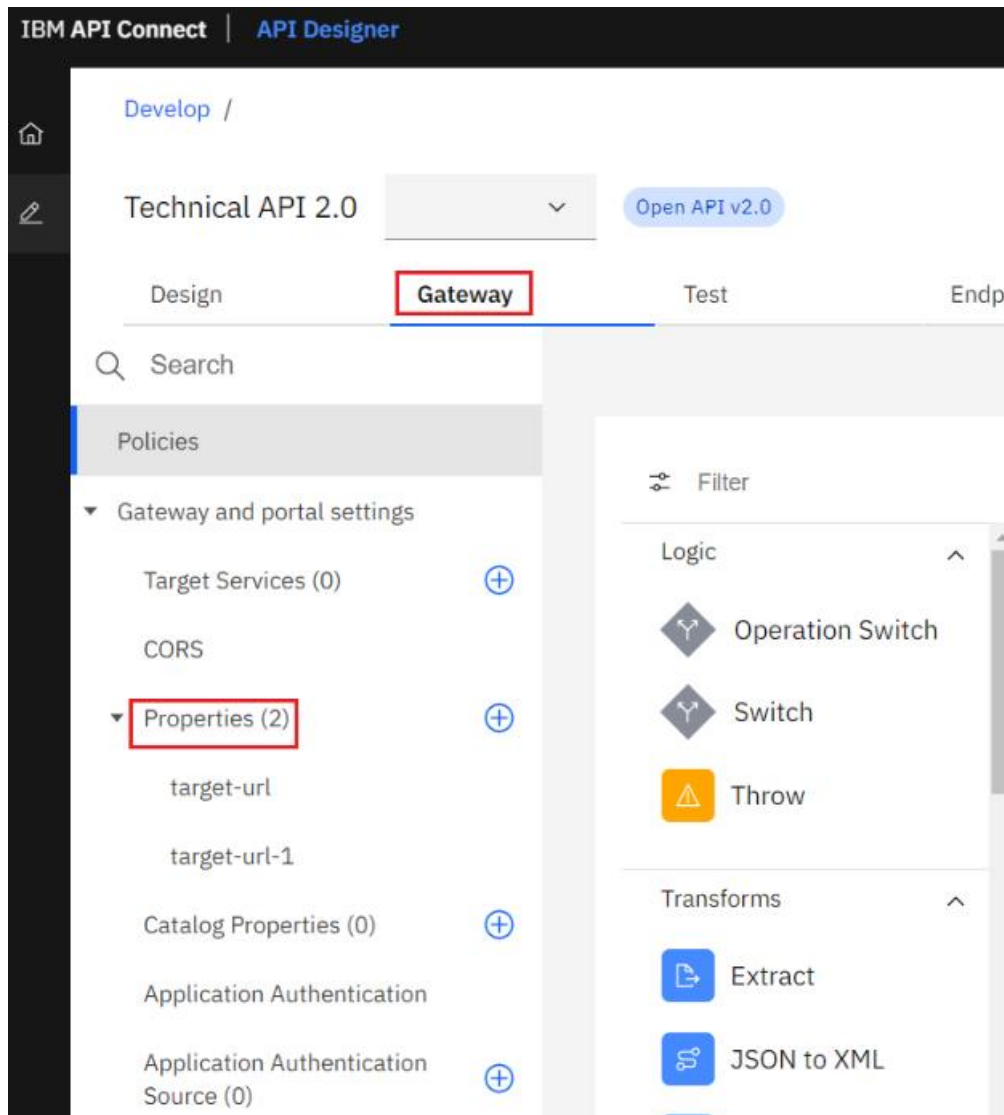
≡

Cancelar

Crear

1.1.4 Propietats

En la vista Gateway d'API Connect, hi ha les **Propietats**.



Les propietats ens permetran definir variables en funció del catàleg en el qual es trobi desplegada la nostra API. Posteriorment, es podran utilitzar aquests valors tant en el disseny com en l'acoblament de l'API utilitzant **\$(nom-variable)**.

Són molt útils i indispensables per poder definir variables que canviïn en cada entorn, com hosts, ports, usuaris, contrasenyes, etc.

Haurien de seguir les següents regles de nomenclatura:

- Usar lletres minúscules únicament.
- Usar guió mitjà en utilitzar paraules compostes (com a màxim 3 paraules).
- No usar guions baixos.
- Usar caràcters numèrics amb moderació, essent la recomanació la seva inclusió només com a enumeració al final del nom de la propietat per distingir propietats similars.

Estandards de Nomenclatura

En el cas de l'API de exemple de Technical API, tindríem les propietats **target-url** i **target-url-1** que indiquen URLs a les quals es vulgui redirigir mitjançant la política d'Invoke.

Per a més informació en relació amb el disseny d'URLs, el disseny de recursos, gestió d'errors, paginació..., consultar el document de *Bones pràctiques*.

1.2 Productes

Els productes han de seguir les regles de nomenclatura següents:

- **Títol:** Nom del producte. Ha de ser prou descriptiu. Ha de començar pel codi d'aplicació, cada paraula començar en majúscula i ha de contenir espais entre cada paraula en cas d'existir. **Exemple:** 3292 Gestor CSV.

És validat per les pipelines del SIC. Regles validadores: prd-nam-02-title-starts-with-codi (error) i prd-nam-02-bis-title-titlecase (warn).

- **Nom:** Nom intern del producte. Es recomana que sigui igual que el Títol, amb el codi d'aplicació i totes les lletres en minúscules i amb guions en lloc d'espais. **Exemple:** 3292-gestor-csv.

És validat per les pipelines del SIC. Regles validadores: prd-nam-01-name-starts-with-codi (error) i prd-nam-01-bis-name-kebabcase (warn).

- **Categories:** s'han de crear dues categories, com a mínim, que tinguin l'estructura:
 - '[codi-diàleg]'
 - [codi-diàleg]/[nom-producte-sense-codi]

```
info:
  version: 1.0.0
  title: 3292 Technical Product
  name: 3292-technical-product
  categories:
    - '3292'
    - 3292/technical-product
```

És validat per les pipelines del SIC. Regles validadores: prd-tag-01-categories-exists (error) i prd-tag-02-categories-has-codi-project (error).

- **Plans:** de cara a estandaritzar els plans de subscripció dels productes que poden usar els desenvolupadors dins de CTTI, s'han definit i acordat els següents tiers: **Default Plan, Bronze, Silver, Gold** (és validat per les pipelines del SIC). Els plans, per tant, hauran de seguir la següent nomenclatura:
 - **Nom:** el nom del pla haurà de començar per un dels quatre tiers i estar en minúscules, separant cada paraula per guions. En cas que es vulgui afegir alguna cosa més que el nom del tier, s'haurà d'introduir a continuació,

Estandards de Nomenclatura

després d'un guió. **Exemples:** *gold-auth (Gold Auth)*, *default-plan-users (Default Plan Users)*, *silver-external-clients (Silver External Clients)*, etc.

És validat per les pipelines del SIC. Regla validadora: `prd-pla-02-name-tier-period-pattern (warn)`. El nom del pla ha de seguir el patró `{tier}` o `{tier}-{funcional}` amb tier a `default-plan`, `bronze`, `silver` o `gold`.

- **Títol:** títol haurà de ser igual que el nom, cada paraula començar en majúscula i ha de contenir espais entre cada paraula en cas d' existir.
- **Descripció:** descripció del pla amb prou detall per ser fàcilment entendible, si és necessària. També es pot deixar amb un valor igual al títol.

```
plans:
  gold:
    title: Gold
    description: Gold
```