

MYCITE2 API 1.0

Verzió: 1.0 v 7

Tartalom

Bevezető	3
REST alapelvek	3
Erőforrások	3
Műveletek	4
Kérés és válasz formátumok	4
Autentikáció és autorizáció	5
Erőforrások hivatkozása	5
MyCite2 Adatformátum	6
Snippetek, adatgenerálási mélység	7
0-ás, snippet mélység	8
1-es, alapértelmezett mélység	9
2-es mélység	9
Az adattartalom szabályozása	9
MyCite2 adatlekérdezési és keresési lehetőségek	12
Kereshető mezők	12
MyCite2 API leíró „api-docs”	13
Általános műveletek erőforrásokkal	15
Új létrehozása	16
Hivatkozás	16
Lista lekérdezése	16
Módosítás	18
Törlés	19
A rendszerben általánosan kezelt erőforrások	19
A rendszer fő erőforrásai	19
Közlemény	19
Szerző	19
Folyóirat	19
Szervezet	20

Konferencia	20
Pályázat	20
Kiadó	20
Mentett lekérdezések	20
Mycite kereső operátorok és speciális keresőmezők	21
Általános kereső operátorok mezők típusa szerint	21
Speciális keresőmezők és operátoraik minden rekordtípusnál	27
Speciális keresőmezők és operátoraik egyes rekordtípusoknál	29

1 Bevezető

Az MyCite2 architektúrájának két fő eleme van:

- az adatkezelő backend szerver
- a megjelenítési frontend felhasználói felület

A backend és frontend között a kommunikáció egy HTTP protokoll alapú REST API-n keresztül történik.

Az API az összes backend funkció tekintetében ugyanazokat az elveket, megoldásokat és adatformátumokat követi, ilyen módon adva uniformizált megoldást a backend és frontend közötti kommunikációra.

A backend által szolgáltatott API külső kliensek számára is hozzáférhető, amennyiben rendelkeznek a megfelelő jogosultságokkal.

2 REST alapelvek

A REST működése kapcsán két alapfogalmat szükséges tisztázni, úgymint maga a REST és a JSON.

A REST (REpresentational State Transfer) egy skálázható, generikus, gyártófüggetlen, bővíthető, megfelelő szakszerű tervezés esetén egyszerű architektúra típus, ami a standard HTTP protokollra épül. Mivel a manapság legelterjedtebb hálózati applikációs protokollra (HTTP) épül, így a használata lényegében bármilyen platformról elérhető, ezáltal felhasználhatósága semmilyen módon nem korlátozott.

A JSON (JavaScript Object Notation) egy egyszerű, könnyen olvasható, szöveges, skálázható, flexibilis adatcsere formátum, melyet a REST műveletek bemeneti és kimeneti adatainak kódolására használunk egységesen. A JSON formátum elfogadottsága és rendszerbe való beágyazottsága is teljesen általános, így ennek használata és felhasználása bármilyen rendszerben könnyen megoldható.

2.1 Erőforrások

A REST alapfogalma az erőforrás: minden műveletet erőforrásokon végzünk, az erőforrások pedig URL-lel egyértelműen azonosíthatók. (E leírásban az erőforrásokra gyakran „rekord” néven is hivatkozunk, a két fogalmat egyenértékűnek tekintjük e dokumentáció szempontjából)

Egy REST API esetén az erőforrások két fő csoportra oszthatók: gyűjtemények és egyedek. A gyűjtemények egy adott csoportosítás szerint kigyűjtött (pl. adott típushoz tartozó) egyedek kollektívája. Egy egyed elérése a típusának és azonosítójának megadásával valósul meg, pl. `GET /api/author/1234`, míg egy gyűjtemény lekérése a kigyűjtés típusának megadásával kezdeményezhető, pl. `GET /api/author`.

Az erőforrásoknak lehetnek al-erőforrásai is. Ezek olyan erőforrások, amik – az adott lekérdezés kontextusában – a fő erőforráshoz kapcsolódóan értelmezendők, önállóan nem használандók. Például egy közlemény szerzőségei csak a közleményhez kapcsolódóan nyernek értelmet, így a szerzőségek a közlemény al-erőforrásaként kerülhetnek megvalósításra.

2.2 Műveletek

Az erőforrásokon végezhető műveletek hasonlóak a megszokott adatbázis CRUD (Create, Read, Update, Delete) műveletekhez, azonban a REST esetében ezeket a HTTP szerint megadott (GET, DELETE, POST, PUT, HEAD) műveletekkel valósítjuk meg. Például, a lekérdezés a GET paranccsal, törlés a DELETE paranccsal, és fejléc lekérdezés a HEAD művelet segítségével végezhető.

A PUT és a POST műveletek egyaránt használhatók létrehozásra és felülírásra, de míg a PUT viselkedése - definíciójának megfelelően - idempotens, a POST nem. Éppen ezért a PUT használata az adott erőforrás (azonosítójának) pontos ismeretében javasolt, a POST pedig ellenkező esetben.

A MyCite2 API-ban általában nagyméretű, sok mezővel rendelkező JSON-ok mozognak a backend és a kliens között. Szerkesztési – PUT – műveletek esetében, azért hogy ne kelljen minden adatot mindig átküldeni a backendnek, elég mindig csak azokat a mezőket berakni a JSON kérésbe, amiket a hívó módosítani szeretne. Ez lényegében azzal jár, hogy a PUT nem írja felül a teljes erőforrást a kapott JSON-nal, hanem HTTP PATCH műveletként viselkedik, ami definíció szerint az erőforrások részleges módosítására szolgál.

A REST alapelve szerint a szerver minden esetben állapotmentes - legalábbis ideális esetben. Az állapotmentesség teszi lehetővé, hogy a REST alapú szolgáltatások horizontálisan, egy új REST kiszolgáló példány beállításával skálázhatók legyenek, s így a REST API szerinti műveletek az egyenértékű horizontális kiszolgálók bármelyikén végrehajthatók. A MyCite2 REST API is állapotmentes, az autentikációt leszámítva, melynek adatait a sessionben tároljuk és a JSESSIONID-val azonosítjuk, lásd 2.4 Autentikáció és autorizáció.

2.3 Kérés és válasz formátumok

A REST filozófiának megfelelően egy HTTP kérés URL-jével csak az adott erőforrást azonosítjuk. A kérés más jellemzőit, például a válasza vonatkozó speciális megköteket és elvárásokat az URL-től szeparáltan, de a kérés részeként továbbított HTTP fejlécben definiáljuk. Ilyen jellemző többek közt a kérés típusa és az API verziója. Minden kérés esetén az Accept headerben prioritási sorrendben adjuk meg a kérésre adott válasz elvárt formátumát. Pl.

```
GET /api/publication/1234 HTTP/1.1
  Accept: application/vnd.mtmt2-1.0+json
```

A verziók és típusok egyeztetése tehát az Accept header alapján történik, melynek értéke a következő sablon alapján áll elő:

```
application/vnd.mtmt2[-verzió][+formátum]
```

A verzió egy fő- és egy alverziószámból áll. Pl. 1.0, 2.3. A formátum megadja a válasz ténylegesen elvárt ábrázolási formátumát. A MyCite2 API 1.0-ban ez json vagy xml lehet.

Az MYCITE2 API 1.0-ban válaszok típusa a fentieknek megfelelően az alábbi lehet:

- application/vnd.mtmt2-1.0+json
- application/vnd.mtmt2-1.0+xml

A visszatérés törzsében alapesetben a hivatkozott erőforrás található (egy egyed mezői vagy egy lista egyedekről a mezőkkel) a kérés Accept header-ében megadott formátumban.

A MyCite2 által támogatott válasz formátumok teljes listája az alábbi:

- JSON: `application/vnd.mtmt2-1.0+json;charset=UTF-8`
- XML: `application/vnd.mtmt2-1.0+xml;charset=UTF-8`
- CSV: `text/csv`
- HTML: `text/html`
- MyCite2 kivonat HTML: `text/template+html`

Az `Accept` headeren kívül a válaszok elvárt formátuma megadható a `format` URL paraméter használatával is. Ennek lehetséges értékei:

- JSON: `format=json`
- XML: `format=xml`
- CSV: `format=csv`
- HTML: `format=html`
- MyCite2 kivonat HTML: `format=template`

2.4 Autentikáció és autorizáció

Az API használatát csak autentikált, azonosított és felhatalmazott kliensek végezhetik. Az autentikáció és autorizáció munkamenet-vezérelt, cookie-ban tárolt session id (JSESSIONID) és írási műveletek esetén CSRF token (X-CSRF-TOKEN HTTP fejléc) küldendő minden HTTP lekéréshez. (Megjegyzés: az MyCite2 API a felhasználói jogosultságkezeléshez szükséges JSESSIONID cookiet leszámítva állapotmentes)

A CSRF token beszerezhető a `/login/csrf` címen, a token az adott munkamenetre érvényes.

Az autentikáció a `/login` URL-re POST módszerrel küldött `username` és `password` paraméterek megfelelő kitöltésével lehetséges (`application/x-www-form-urlencoded` formátumban). Sikeres esetben a szerver új munkamenet-azonosítót ad vissza a `Set-cookie` fejlécben és átirányít a CSRF token lekérő oldalra, hiba esetén átirányít a hibát megjelenítő oldalra (`/login/error`).

A munkamenet egy óra után lejár, azután újra be kell jelentkezni. A munkamenet korábbi megszüntetése a `/logout` URL-re küldött POST lekéréssel lehetséges.

A kommunikáció a kliens és a backend szerver között biztonságos HTTPS protokollon keresztül történik.

2.5 Erőforrások hivatkozása

A MyCite2 REST API esetében minden hozzáférhető erőforrás (egyed és gyűjtemény egyaránt) egyedi URL-en érhető el, ez megteremti a lehetőséget a hivatkozásra. Azonban a JSON szabvány nem foglalkozik a linkeléssel, vagyis, hogy milyen módon kapcsoljunk össze két JSON adatot egymással, milyen módon navigálhatunk közöttük. Erre külön mechanizmusok és ajánlások készültek.

Az MyCite2 API 1.0-ban jelenleg a JSON-ba generált „link” mező tartalmazza az erőforrás URI-ját, amin keresztül a REST műveletek végezhetők rajta. Ez a `link` mező minden kapcsolt rekordhoz is generálódik, így módon a kapcsolt rekordok is elérhetők, lekérdezhetők és bejárhatók.

3 MyCite2 Adatformátum

A MyCite2 API elsődleges adatformátuma a JSON, illetve az ezzel szemantikusan teljesen megegyező XML.

A továbbiakban, amikor JSON-ról beszélünk, ezzel egyenértékűként mindig beleértjük az XML formátumot is.

A JSON és XML formátum használható mind bemeneti (PUT, POST műveleteknél), mind kimeneti formátumként (GET műveleteknél), ezen kívül lekérdezésekben (GET műveletekben) kérhető még HTML, CSV, stb. kimeneti formátum is az 2.3 Kérés és válasz fejezetben leírt módon.

A MyCite2 által generált JSON válasz alapvetően 2 részből áll:

- Fejléc adat(ok)
- Válasz adat(ok)

A tényleges válasz adat mindig a JSON content mezőjében található, az ezen kívüli mezőket pedig mind fejléc mezőknek tekintjük. A content mező értéke listás lekérdezés esetében (pl. /api/publication) egy tömb, egyéb esetekben (pl. adott specifikus erőforrás lekérdezésekor, /api/publication/123) egy rekord.

A fejléc mezők az alábbiak lehetnek nem hibával végződő lekérés esetén:

- labelLang: a JSON-ban található feliratok, címkék nyelve
- paging: listás lekérdezés esetében az aktuális lekérdezés lapozási eredménye. Listás lekérdezés esetében az eredménylista a size paraméternek megfelelően lapozódik. Pl. ha a lekérdezés 100 elemet ad vissza, és a size=20, akkor 5 oldalon, oldalanként 20 elemet tudunk lekérdezni a size=20 és a megfelelő page paraméter beállításával. A paging-ben található adatok:
 - o totalPages: összesen ennyi oldalnyi eredményt szolgáltat a lekérdezés
 - o number: az aktuális oldalszám, aminek a tartalmát éppen megkaptuk
 - o size: az egy oldalon maximálisan listázott elemszám. Megegyezik a lekérdezéskor megadott size paraméter értékével.
 - o numberOfElements: az oldalon található elemek konkrét száma, ami kevesebb is lehet a size-nál
 - o first: true, ha ez az első oldal
 - o last: true, ha ez az utolsó oldal
 - o totalElements: ebben a listában ennyi elem lesz lapozhatóan lekérdezhető (ez általában max. 5000)
 - o totalEstimatedElements: ez becsült eredmény, amennyi elemet a lekérés szolgáltathat. Mivel nagy adathalmaz esetén a konkrét elemszám kiszámolása sokáig tarthat, ezért az esetek többségében csak becslést ad a rendszer, azonban a konkrét elemszám kiszámítása is lehetséges, lásd a következő pontot
 - o totalUncutElements: ennyi a konkrét darabszáma, amit a lekérés szolgáltathat (de ebből csak max 5000-et nézhet meg a felhasználó). totalUncutElements értéke csak akkor lesz kitöltve, ha azt a lekérdezés során ténylegesen kiszámoltuk. Ennek kikényszerítésére a ignoreCachedCount=true&ignoreEstimatedCount=true lekérzési paramétereket lehet használni. Ezek biztosítják, hogy ne egy

cachelt eredményt kapjunk vissza, illetve, hogy ne csak becslést adjunk az elemszámmra, hanem konkrétan ki is számoljuk a lekérdezés során. Az `ignoreCachedCount=true&ignoreEstimatedCount=true` beállítást csak indokolt esetben használjuk, mert ezzel a lekérdezés lassabb lesz és jobban is terheli a rendszert.

- `duplums`: duplumok listája, amenyiben a lekérdezés tartalmaz duplumellenőrzést
- `validationErrors`: validációs hibák

A fejléc mezők az alábbiak lehetnek hibával végződő lekérés esetén:

- `message`: hibaüzenet
- `status`: hiba státusza
- `error`: hiba kód
- `path`: a lekérés útvonala
- `timestamp`: a lekérés időbélyege
- `labelLang`: a hibaüzenet nyelve

A MyCite2 által kezelt erőforrások, pl. Közlemények, Szerzők, Intézmények általában sok (akár 30-50) adatmezővel és sok kapcsolódó erőforrással (akár 20) rendelkezhetnek. Ilyen mezőszám és adattartalom mellett nem célszerű a fő- és kapcsolódó erőforrásokat mindig külön kezelni. Ennek egyrészt teljesítmény okai vannak, hiszen ha lekérdezzük egy főerőforrást, majd annak 20 kapcsolódó erőforrását mind külön HTTP kérésként, akkor ez jelentős terhelést okozna a backendnek, és növelné a válaszidőket a kliensnek.

Ehelyett a MyCite2 -ben, némileg hasonlóan a GraphQL-hez erőforrások és azok kapcsolódó erőforrásai egyben lekérdezhetőek, illetve konfigurálható az adattartalmuk. Például, ha egy Közlemény lists lekérésnél csak a szerzőségekre vagyunk kíváncsiak, akkor az alábbi URL-lel leszűkíthetjük a visszatartott JSON mezők körét:

```
/api/publication?incFields=authorships
```

Ebben az esetben a Közlemények egyes állandó mezőin kívül (`mtid`, `otype`, `link`, `label`) csupán az `authorships` mezőben lesznek adatok.

3.1 Snippetek, adatgenerálási mélység

A MyCite2 adatformátumában két fontos fogalom az *adatgenerálási mélység* és a *snippetek*.

Ahogy az előző fejeztben kifejtésre került, a MyCite2-ben nagyméretű, sok mezővel rendelkező JSON adatok közlekednek a backend és a kliens között, és az adatok általában sok kapcsolt adatot is tartalmaznak. Mivel nem lenne célszerű mindig minden adatot átküldeni a backend és a kliens között, ezért szabályozni kell, hogy a főerőforrásból és a kapcsolódó erőforrásokból mennyit tartalmazzon a kérés vagy válasz JSON-ja.

Válasz JSON esetében az adatgenerálási mélységet alapvetően a `depth` URL paraméter határozza meg. Ennek 3 értéke lehet:

- 0: ez az úgynevezett „snippet” (kivonat, darabka) ábrázolás
- 1: alap mélységi érték
- 2: mint az 1-es, plusz a kapcsolt rekordok is 1-es mélységben generálva.

Az alábbiakban látható egy példa egy 1-es mélységben generált rekord leíróra. A 123997 azonosítójú *Thesis* típusú közlemény 1-es mélységben került kigenerálásra, míg a kapcsolódó `authorships` mező snippetként, 0-ás mélységben.

```
{
  "content": [
```

```

{
  "otype": "Thesis",
  "mtid": 123997,
  "status": "APPROVED",
  "published": true,
  "lastModified": "2015-02-08T17:07:48.746+0000",
  "created": "2014-10-21T14:03:00.000+0000",
  "type": "Thesis",
  "title": "A gazdaképzési rendszerek összehasonlító vizsgálata
a XIX. századtól a XX. század második feléig.",
  "publishedYear": 1993,
  "link": "/api/publication/123997",
  "label": "A gazdaképzési rendszerek összehasonlító
vizsgálata a XIX. századtól a XX. század második feléig.",
  "authorships": [
    {
      "otype": "Authorship",
      "mtid": 124022,
      "link": "/api/authorship/124022",
      "label": "Szávai Ferenc"
      "snippet": true
    }
  ]
}
]
}

```

A következő fejezetekben részletesen ismertetjük az egyes mélységi paraméterek működését és használatát.

3.1.1 0-ás, snippet mélység

A 0-s, vagy „snippet” mélység esetén csak az adott rekord leglényegesebb adatait tartalmazza a JSON. Ezek azok az adatok, amik minimálisan azonosítják a rekordot (pl. azonosító, típus, link), illetve azok az adatok, amiket leggyakrabban használunk és jelenítünk meg a frontenden. A snippet ábrázolás azért is kiemelt jelentőségű, mert minden rekord snippetjét eltároljuk egy cacheben, így a snippet mélység kigenerálása, amennyiben már cachelésre került, igen gyors.

Minden snippet tartalmazza az alábbi adatokat:

- `mtid`: a rekord azonosítója
- `otype`: a rekord típusa
- `link`: az erőforrás elérése
- `label`: a rekord címkéje, ember számára érthető tartalommal
- `template`: rekord kivonata HTML formátumban
- `template2`: rekord bővített kivonata HTML formátumban, ha rendelkezésre áll
- `snippet`: értéke mindig „true” jelezve, hogy ez a JSON leíró egy snippet. Ha egy JSON leíró nem tartalmaz `snippet:true`-t, akkor arra úgy tekintünk, mint egy teljes JSON leíróra.

Amikor a kliens küld snippetként adatot, akkor ezek közül csak az `mtid` és az `otype` és a `snippet: true` megadása szükséges, mivel ezek azonosítják programatikusan a rekordot. A backend által generált válaszban, viszont az összes fenti mező megtalálható.

A `label`, `template`, `template2` a felhasználó számára értelmezhető módon azonosítja a rekordot különböző részletességig. A `label` csupán egy egysoros címke, míg

a `template` és `template2` strukturált, HTML formátumú leírás, Közlemények esetében, például, bibliográfiai jellegű adatokkal és részletességgel. (A `format=template` kimeneti formátum ez alapján a `template` vagy `template2` adat alapján generálja a kimenetet)

A fenti snippet adatok alapján az egyes rekord típusok különböző egyéb adatot is tartalmazhatnak a snippetjükben, ezeknek köre és adattartalma típusonként eltérő, és a MyCite2 frontend szempontjából fontos adatokat tartalmazza.

A kliensek által küldött `snippet:true` jelzetű JSON-ok működése rekordok szerkesztésekor válik fontossá. Például, amikor egy szerzőséghez szerzőt akarunk csatolni, akkor nem szükséges a Szerző rekord minden mezőjével ellátott JSON-t megadni, elég csak a csatolni kívánt Szerző snippet-jét elküldeni, amit referenciaként használ fel a backend és a Szerző snippet adatai alapján megkereresi az adott szerzőt és csatolja – a Szerző mindennemű módosítása nélkül – a szerzőséghez. Ha ellenben nem kerülne `snippet:true` jelzet a szerzőre, akkor a JSON-t mint teljes rekordleíró értelmelné a backend és a csatolás mellett a Szerző adatit is megpróbálná szerkeszteni a kéréssel.

3.1.2 1-es, alapértelmezett mélység

Az 1-es adatgenerálási mélység az alapértelmezett. Ebben a mélységben a legfelsőbb szintű erőforráshoz tartozó rekord minden egyszerű típusú (szám, szöveg, dátum, boolean, stb.) mezője szerepelni fog a JSON-ban, míg a kapcsolódó rekordokról 0-ás mélységű snippet ábrázolást generál a backend. Az 1-es mélységben generált JSON tartalmaz minden olyan adatot, amit a frontendnek és a legtöbb kliensnek ismernie kell ahhoz, hogy egy rekordot értelmezni és kezelni tudjon. Mivel az 1-es mélységben a kapcsolódó rekordok snippet ábrázolását használjuk, és ez az előzőekben említett módon cachelve van, ezért az 1-es mélységű adatábrázolás is igen gyors, amennyiben a kapcsolt rekordok snippetje már cachelésre került.

Általában véve, ha 1-es mélységű adatgenerálást kérünk, akkor ez kívánja a legkevesebb adatbázis műveletet, ezért minden klienstől ennek használata az elvárt és javasolt.

3.1.3 2-es mélység

A 2-es mélységű adatgenerálás esetén a kapcsolt rekordok nem snippetként, hanem 1-es mélységben generált JSON-ként kerülnek az eredmény JSON-ba.

2-es mélységű adatgenerálásra csak igen kivételes esetben lehet szükség, mivel az mindig kiváltható egy 1-es mélységben kigenerált főrekorddal, aminek a kapcsolt (snippet) rekordjait egy külön 1-es mélységű kéréssel lekérheti a kliens.

A 2-es mélységű adatgenerálás nem javasolt, mivel ez értelemszerűen nem tudja használni a kapcsolt rekordok snippet ábrázolását (tekintve azok 1-es mélységét), így egy minden egyes kapcsolt rekordra/rekord listára külön adatbázislekérdezést igényel.

1.1 Az adattartalom szabályozása

Az adatgenerálási mélység imlicit módon szabályozza, hogy a kimeneti JSON-ba milyen adatok kerülnek. Emellett azonban egyéb módokon, egészen finoman, mező szinten is szabályozható, hogy a főrekord vagy kapcsolt rekordok mely mezői kerüljenek az eredmény JSON-ba.

Általános szabályozó URL paraméterek:

- `labelLang`: a JSON-ba generált ember számára olvasható feliratok nyelve. Lehetséges értéke: `hun` (magyar), `eng` (angol)
- `format`: az 2.3 fejezetben leírtak szerint a generált eredmény formátumát szabályozza. Lehetséges értékei: `json`, `xml`, `csv`, `html`, `template`
- `depth`: az adattartalom generálás általános mélysége. Lehetséges értékei:
 - 0: snippet ábrázolás
 - 1: alap mélységi érték, kapcsolt rekordok snippetként
 - 2: mint az 1-es, plusz a kapcsolt rekordok is 1-es mélységben.
 - `e`: speciális „export” mélység, mely egyes kapcsolt rekordokat annak megfelelően 0-ás vagy 1-es mélységben generálja, hogy a visszaimportáláshoz milyen mélységre van szükség.
 - `eo`: mint az „e” mélység, de opcionális, hogy visszaimportnál az adott rekordot beimportáljuk-e újként, vagy sem.
- `maxConnectedCount`: listás kapcsolt rekordok esetén csak az itt megadott számú elemet generáljuk az eredménybe. Pl. ha egy publikáció 2000 szerzővel rendelkezik, de nem kívánjuk mindent megmutatni, akkor `maxConnectedCount=5` beállítással csak az első 5 szerző fog bekerülni a JSON-ba.
- `ignoreBackRefs`: egyes kapcsolt rekordok rendelkeznek visszamutató kapcsolattal, melyek generálása az adatok értelmezése szempontjából redundáns, ezért `ignoreBackRefs=true` alkalmazásával ezek a visszamutató kapcsolatok elhagyhatók, ezzel is csökkentve az eredmény JSON méretét.
- `writeNulls`: a null értékű mezőket alapesetben nem generálja bele a backend a JSON válaszbba. `writeNulls=true` esetén ezek is belekerülnek.
- `ignoreUnhandled`: a kliens által küldött, de a MyCite2 backend által nem kezelt JSON mezőket a backend visszaküldi, így a kliensnek módja van bizonyos számára fontos állapotokat úgy elküldeni a backendnek, hogy a válaszban azokat vissza is kapja. Ez a működés letiltható az `ignoreUnhandled=true` paraméter megadásával.
- `ignorePermissionRelated`: ha értéke `true`, akkor a JSON-ba nem generál ops mezőt, ami az adott objektumra vonatkozó jogosultságokat tartalmazza. Ezt célszerű olyankor használni, ha a jogosultsági paraméterekre egyébként sem lesz szükség, ezzel a JSON generálás gyorsítható.
- `export`: ha értéke `true`, akkor `Content-Disposition: attachment` headerrel adja vissza generált tartalmat, ami böngészőben meghívva fájl letöltésként viselkedik, vagyis az adott URL tartalmát egy fájlba menti el.
- `exportFormat`: a rendszer által előre megfogalmazott `depth`, `format`, `fields`, `incFields`, `exFields`, stb. paraméterek, amelyek előre meghatározott kénező JSON/XML/CSV/HTML generálására szolgálnak, tipikusan export célokból. Az export formátumok rekordtípusonként eltérőek lehetnek, azok teljes listáját a központi adminok szabályozhatják az `ExportFormat` adattípus alatt.

E publikáció elkészültekor érvényben lévő beépített export formátumok (az `exportFormat` lehetséges paraméterei):

- Minden adattípushoz alkalmazható export formátumok:
 - `genericJson`: Alap JSON formátum
 - `genericXml`: Alap XML formátum
 - `genericTransferJsonExport`: JSON formátum MTMT szerverek közötti átvitelre
 - `genericTransferXmlExport`: XML formátum MTMT szerverek közötti átvitelre
 - `templateHtml`: Egyszerű rekordszerű HTML formátum
 - `csvExcel`: Excelbe importálható CSV formátum
- Publikáció típushoz használható export formátumok:
 - `basicPublicationJson`: Alap Közlemény JSON formátum
 - `transferPublicationJson`: JSON formátum MTMT szerverek közötti átvitelre
 - `basicPublicationXml`: Alap Közlemény XML formátum
 - `transferPublicationXml`: XML formátum MTMT szerverek közötti átvitelre
- Idézőkapcsolat típushoz használható export formátumok:
 - `citationTransferJsonExport`: JSON formátum MTMT szerverek közötti átvitelre
 - `citationTransferXmlExport`: XML formátum MTMT szerverek közötti átvitelre
- Szerző típushoz használható export formátumok:
 - `basicAuthorJson`: Alap Szerző JSON formátum
 - `basicAuthorXml`: Alap Szerző XML formátum
 - `transferAuthorJson`: JSON formátum MTMT szerverek közötti szerző átvitelre
 - `transferAuthorXML`: XML formátum MTMT szerverek közötti szerző átvitelre

Mezőszintű szabályozó paraméterek:

- `fields`: egyes mezők generálási mélységének szabályozására szolgál. Pl. `fields=authorships:1,authorship.author:1,citations:0` esetén az `authorships` 1-es mélységgel generálódik, azok belül az `author` mező, ami alapvetően már 0-ás mélységű lenne, mégis 1-es mélységgel generálódik, míg a `citations` mező elemei mind 0-ás mélységűek lesznek.
- `incFields`: a kimeneti JSON-ban, csak az itt felsorolt mezők és az itt megadott mélységben lesznek a JSON részei. Pl. `incFields=authorships:1,citations:0`. Esetén csak az `authorships` és `citations` mezők kerülnek a JSON-ba, az `authorships` elemei 1-es mélységben, míg a `citations` elemei 0-ás mélységben. Ha `incFields` megadásra kerül, akkor az `exFields`-ben megadottak nem kerülnek értelmezésre.
- `exFields`: egyes mezők kihagyása a JSON-ból. Pl. `exFields=citations,ratings,book.cites` esetén az adott közlemény minden mezője bekerül kivéve a `citations`, `ratings`, illetve a `book` mezőn belül a `cites` mező, míg a `book` kapcsolt rekord többi eleme nem.

- `ignoreImplicitSnippetTrueForExFields`: `exFields` esetén bizonyos kardinális mezők nem kerülnek be az `excludálás` miatt a JSON-ba, ezért azt implicit módon `snippet:true` jelzéssel látja el a backend. `ignoreImplicitSnippetTrueForExFields=true` megadással nem kerül a JSON leíróba ez az implicit `snippet:true` jelzés.

A mezőszintű szabályozó paramétereknél az `incFields` és `fields` paraméterek körültekintő használata szükséges, hogy a végeredményül várt JSON ne legyen túlságosan nagy a kliens számára.

Alapvetően a megfelelő `depth` megadását (`depth=1`) és `exFields`-szel, vagy `incFields`-szel történő szűkítését javasoljuk.

Egyéb paraméterek:

- `ignoreSnippets`: ha értéke `true`, akkor a JSON generáláshoz nem használja a snippeteket. Ez jelentősen lelassítja a JSON generálás működését, ezért csak abban az esetben használjuk, ha azt gyanítjuk, hogy a JSON valamilyen hibája miatt van, hogy régi vagy rosszul generált snippetek vannak benne. A `ignoreSnippets=true`-val a JSON eredmények mindig garantáltan megfelelnek az adatbázis táblákban tárolt értékekkel.
- `debugSnippets`: ha értéke `true`, akkor a JSON-ba belegenerálja egy külön mezőbe azokat a snippet JSON-okat, amiből az eredmény JSON-t generáljuk. Ez a kaócsoló jelentősen lelassítja a JSON generálás működését, és csak hibakeresésre használandó.
- `denyIneditableFieldSetting`: ha értéke `true`, és a hívó olyan mező értékét próbálja meg állítani, ami (jelenleg) nem szerkeszthető, akkor 403-as (access deny) hibát ad vissza a backend. Alap esetben a nem szerkeszthető mezők szerkesztését csendben ignorálja a backend, mint ha nem is próbálták volna meg beállítani azt a mezőt, a `denyIneditableFieldSetting=true`-vel ez felülbírálahtó. Alkalmazása csak ritkán lehet szükséges.

2 MyCite2 adatlekérdezési és keresési lehetőségek

A MyCite2 egy kiterjedt adatlekérdezési mechanizmust nyújt, aminek segítségével komplex keresések készíthetők, így pontosan szabályozható, hogy mely adatokat kívánja valaki megjeleníteni, lementeni, stb.

Kereshető mezők

A MyCite2 objektumai az `api-docs`-ban megadott rekord típusokra és azok mezőire vonatkozóan engedélyezi lekérdezési műveleteket végrehajtását. A rekordtípusokban az egyes mezők `x-mycite` részében megadott `searchable` paraméter értéke határozza meg. Ennek értéke az alábbi lehet:

- ALL: bárki számára kereshető
- NONE: senki számára nem kereshető
- `<Role>`: a mező az adott szerepkörben kereshető. Ennek értéke:
 - Admin1

- o Admin2
- o Admin3
- o Admin4
- o Admin5
- o Admin6
- o Author
- o Owner
- o Handler
- o User
- o Importer
- o Visitor

jelenleg ALL vagy NONE lehet. Az ALL jelenti, hogy a mező kereshető, a NONE pedig, hogy nem.

```
"PublicationSource": {  
  "type": "object",  
  "discriminator": "",  
  "properties": {  
    "repoType": {  
      "type": "string",  
      "description": "Repozitórium típusa",  
      "enum": [  
        "Eprints",  
        "Dspace",  
        "Atom"  
      ],  
      "x-mycite": {  
        "validFor": [  
          "SwordSource"  
        ],  
        "gridOrder": 26,  
        "searchable": "ALL",  
        ...  
      }  
    }  
  },  
}
```

3 MyCite2 API leíró „api-docs”

Az előző fejezetekben bemutattuk a MyCite2 API általános tudnivalóit. Az API konkrét végpontjait, elvárt paramétereit és válasz formátumait az [OpenAPI Specification](#)-nek (ismertebb nevén [Swagger](#)-nek) megfelelő leíró adja meg, ami a MyCite2 backend

```
/api-docs
```

URL-jén érhető el. A MyCite2 api-docs tartalmaz pár módosítást az OpenAPI Specification-höz képest. Az OpenAPI Specification-nek teljes egészében megfelelő api-docs, az alábbi módon kérhető le:

```
/api-docs?ignoreInheritance=true&ignoreMyciteValues=true
```

MyCite2 api-docsban a leírók kiegészítésre kerültek MyCite2 specifikus adatokkal, amik a frontend számára nyújtanak segítséget az API használatában és a megjelenés szabályozásában.

A MyCite2 kiegészítések az api-docsban egységesen az

```
x-mycite
```

mezőkben kerültek tárolásra. Más, külső kliensek számára ezek az adatok valószínűleg nem fontosak, ezért ezek a `ignoreMyciteValues=true` paraméterrel letilthatók.

A MyCite2 api-docsban a modellek leírói nem tartalmazzák az adott osztályok minden mezőjének leírását, csak az adott osztályban közvetlenül definiáltakét, és minden modellben az `x-mycite.extends` mezővel hivatkozik a szülő modell osztályra. Ezáltal a modell leírók sokkal kompaktabbak tudnak lenni, viszont a kliensnek kell utána felépíteni a tényleges modelleket a leszármazási hierarchia bejárásával. Ez a hierarchikus megadási mód letiltható az `ignoreInheritance=true` paraméterrel.

Mivel az adatmodell sok hasonló módon kezelhető entitást tartalmaz, nem célunk minden egyes entitásra külön megfogalmazni az API hívásokat, inkább általánosíthatóan próbáljuk a rendszer főbb erőforrásait és azok API-n keresztüli elérését és működését bemutatni az alábbiakban, a részleteket tekintetében, pedig az api-docs-ban megadottak a mérvadóak.

Az API-ban minden erőforráshoz megadjuk az elérési URL-eket. Az API-ban használt URL-kel kapcsolatos általánosan alkalmazott alapelvek:

- Minden erőforrás URL-je egységesen `/api-`val kezdődik
- Minden erőforrás URL második paramétere az erőforrásként kezelt rekord típusa, pl. `/api/publication`, `/api/author`
- Minden erőforrás (rekord) egyetlen URL-en keresztül érhető csak el. Pl. a Szerző rekordokat mindig `/api/author/123` URL-lel érjük el

Az URL-ek, bár kiismerhető és előre definiált logika alapján generálódnak, átlátszatlanok tekintendők, vagyis az API használójának soha nem szabad “kézzel” összeállítania URL-eket, mindig csupán a rendszer által generált, előző kérésből megismert URL-eket szabad használnia. Ez azt jelenti, hogy bár egy Szerzőre vonatkozó erőforrás URL-je a szerző azonosítójának ismeretében “kigenerálható”, pl. `/api/author/9978`, ezt az API használójának nem szabad és általában nem is szükséges megtennie, mivel az URL formátumáról a kiszolgáló szerver gondoskodik, annak formátuma előzetes jelzés nélkül is változhat. Az URL-t, mint egyedi, önálló jelentés nélküli, “átlátszatlan” azonosítót szükséges használni.

Megjegyzés: mivel a MyCite2 api-docs meglehetősen nagy méretű, ezért egyes Swagger eszközök, így maga a Swagger UI sem tudja kezelni, viszont a [swagger-codegen](#) használható az api-docs alapján generált kliensek, dokumentációk, és egyéb kimenetek generálására. A swagger-codegen-nel a legkezelhetőbb dokumentációt pl. ezzel a paranccsal lehet elérni:

```
java -jar swagger-codegen-cli.jar generate -i
'https://playground.mtmt.hu/api-docs?ignoreMyciteValues=true&ignoreInheritance=true' -l html -o ./html
```

3.1 Általános műveletek erőforrásokkal

Az MyCite2 által kezelt rekordok, illetve az ezeket menedzselő REST erőforrások (resource-ok) működése az alpműveletek tekintetében megegyezik, ezért ezeket nem erőforrásonként külön-külön ismertetjük, és ismételjük mindannyiszor, hanem generikus módon mutatjuk be ebben a fejezetben, majd a későbbiekben, az erőforrások ismertetésénél erre a generikus működésre hivatkozunk.

Az MyCite2 szolgáltatás által kezelt rekordokat az API-n keresztül általános módon kezeljük, vagyis minden erőforrással, ha csak külön nem jelezzük, az alábbi műveletek végezhetőek:

Művelet	HTTP metódus és URL séma
Új létrehozása	POST /<resource>
Hivatkozás	GET /<resource>/{id}
Lista lekérdezése	GET /<resource>? cond=...&negeated=...&join=...&sort=...& size=...&page=...&fields=...&embed=...
Módosítás	PUT/PATCH /<resource>/{id}
Törlés	DELETE /<resource>/{id}

Az erőforrásokon végzett műveletek bemenete minden alkalommal JSON vagy XML formátumú, kimenete pedig a 2.3 Kérés és válasz formátumok fejezetben leírtak szerint adható meg. Általánosságban a bemeneti JSON-ok mindig elegendő információt tartalmaznak a művelet elvégzéséhez, a kimeneti JSON-ok pedig tartalmazzák a művelet eredményeként létrejövő rekordok leírását, vagy ilyen rekordok listáját.

Az egyes műveletekhez tartozó HTTP kérésekben az alábbi fejléceket (header) alkalmazzuk általánosan:

- **Accept.** Ez a header szolgál az eredményként elvárt adattípusának (Pl. JSON) és verziójának (pl. v2) megadására. Bővebben lásd a “Verzió és típus egyeztetés” fejezetet.
- **Authorization.** A kérést végző entitás azonosítására szolgál, ennek helyes és érvényes megadása esetén végezhető el a kívánt művelet. Bővebben lásd az “Autentikáció és autorizáció” fejezetet.

Az egyes műveletek végrehajthatóságát a műveletet végző felhasználó (ahogy az Authorization headerben megadásra került) jogosultságai határozzák meg. Ha egy művelet végrehajtásához nincsen joga, akkor a művelet nem hajtódik végre és hibaüzenetet kap a hívó.

Az eredmény feldolgozásánál figyelembe kell venni a HTTP visszatérési kódot is. Amennyiben ez nem 2xx, akkor a kérés valamilyen okból nem sikerült, és meg kell ismételni. Ilyen hibás esetekben a válasz tartalmazhat a hibára vonatkozó magyarázatot vagy hibakódot is.

Új létrehozása

Egy HTTP POST művelet, amivel a POST során megadott JSON-nak megfelelő rekordot hozhatunk létre. A művelet eredményeként visszakapjuk az új resource URL-jét, amivel a továbbiakban hivatkozhatunk rá.

Lehetséges HTTP válaszkódok:

- 200 OK: az rekord létrehozása sikeres volt.
- 400 Bad Request: hibás input JSON
- 401 Unauthorized: a hívó fél nem azonosította magát (nincs belépve)
- 403 Forbidden: a hívó félnek nincs jogosultsága a művelethez
- 404 Not Found: nem létező resource-ra vonatkozó kérés
- 405 Method Not Allowed: a resource-on az adott HTTP művelet nem engedélyezett (kivételes esetben használt kód)
- 422 Unprocessable Entity: a bemeneti adatok nem érvényesek (validációs hiba)
- 500 Internal Server Error: nem várt hiba lépett fel a kérés teljesítése közben

Hivatkozás

Egy adott rekordra HTTP GET metódussal és az rekordhoz társított URL-lel hivatkozhatunk. Az rekord URL-jét az új rekord létrehozásának, vagy egy lista lekérdezés művelet eredményeként ismerhetjük. A visszaadott JSON a hivatkozott rekord teljes leírását tartalmazza.

Lehetséges HTTP válaszkódok:

- 200 OK: a hivatkozás lekérése sikeres volt
- 401 Unauthorized: a hívó fél nem azonosította magát (nincs belépve)
- 403 Forbidden: a hívó félnek nincs jogosultsága a művelethez
- 404 Not Found: nem létező resource-ra vonatkozó kérés
- 405 Method Not Allowed: a resource-on az adott HTTP művelet nem engedélyezett (kivételes esetben használt kód)
- 500 Internal Server Error: nem várt hiba lépett fel a kérés teljesítése közben

Lista lekérdezése

Adott típusú rekordok keresésének művelete HTTP GET metódussal.

Általánosságban minden erőforrás az alábbi paraméterekkel kérdezhető le:

```
GET /<resource>?cond=...&join=...&negated=...&sort=...&size=...&page=...
```

Állandó paraméterek:

- cond, join, negated: a lekérdezési feltételeket definiálják.
 - cond: egy lekérdezési feltétel. Egy feltétel 3 részből áll: a <mező>;<operátor>;<operandus>. A mező az adott erőforrás egy mezője, vagy egy ponttal (.) elválasztott mező sorozat, ami az

erőforrás kapcsolódó rekordjainak mezőire mutat. Az operator definiálja a feltétel műveletét. Az operandus a feltétel művelet tárgya, ami lehet egyetlen elem, vagy elemek sorozata vesszővel (,) elválasztva. Például:

```
cond=title;prefix;Web
```

```
cond=author.employments.worksFor;ini;100022
```

Az első esetben az rekord egy saját mezőjének a title-nek az értékét vizsgáljuk, mégpedig, hogy a Web karakter sorozattal kezdődik-e. A második esetben egy mező sorozat van megadva, vagyis az adott rekord author mezője által mutatott Author rekord employments mezője által mutatott Employment rekordban a worksFor mező értékét vizsgáljuk, mégpedig, hogy az adott mtid-vel megadott Institute értékének megfelel-e.

Feltételből tetszőleges számú megadható több cond egymás utáni felsorolásával. Pl:

```
cond=title;prefix;Web&cond=publishedYear;gt;2014
```

Ha a cond operandus részében zárójelre akarunk keresni, akkor a zárójelet escapeelni kell ‘\’ karakterrel, ami URL paraméterként %5C kóddal írandó. Pl:

```
cond=identifiers.idValue;eq;3975%5C(01%5C)00103-7
```

- join: a join megadja, hogy a feltételeket, amennyiben több van megadva belőlük, milyen módon kell összekapcsolni. A két lehetséges értéke az AND és az OR, vagyis, hogy a feltételek akkor vezetnek igaz eredményre, ha minden feltétele teljesül (AND/ÉS kapcsolat), vagy ha a feltételek közül legalább az egyik eredménye igaz (OR/VAGY kapcsolat)
- negated: ha értéke true, a feltételeket negálja (mindegyiket külön) és így értékeli ki őket
- sort: rendezési feltétel. Mely mező alapján milyen irányú rendezés. Sortból több is megadható, és a rendezést a sort megadási sorrendjében végzi a rendszer, Pl. `sort=title,asc&sort=publishedYear,desc` Ebben az esetben az eredményt először cím szerint növekvően rendezi, majd az azonos nevéken belül publikálási dátum szerint csökkenően.
- size, page: lapozási feltételek, melyekkel megadható, hogy válaszként maximum hány elemet (size) kapjunk vissza, és hogy hányadik oldaltól (page).
- maxCount: Egy keresésre a maximálisan lekérhető elemek száma, lapozással együtt. (Alapesetben ez - jogosultságoktól függően - max. 5000.) Csak központi adminok használhatják. A maxcount megnövelése esetén a későbbi oldalak lekérdezése egyre lassabb lehet, ezért nagy oldalhalmazok lekérése esetén lehetőleg minél nagyobb oldalméretet kell használni (size=1000 az ideális).
- ignoreCachedCount: optimalizációs okokból, ha a lekérdezés eredményeinek száma meghalad egy bizonyos értéket (jelenleg 5000-et), akkor az eredményszámot cacheljük, és ugyanezen paraméterekkel végzett lekérdezésekhez a jövőben ezt az

elmentett értéket adjuk vissza. A cachelt értékeket 24 óráig tároljuk, ezután újraszámoljuk. Az `ignoreCachedCount=true` beállítással ez a cachelés letiltható, és az eredmények számát mindenképpen újraszámoljuk, de ez hosszú ideig is eltarthat százezres vagy milliós elemszámmal, ezért használata csak indokolt esetben javasolt.

- `ignoreEstimatedCount`: optimalizációs okokból, ha a lekérdezés eredményeinek száma várhatóan igen nagy lesz (>100.000), akkor nem számoljuk meg a list tényleges hosszát, hanem csak egy becslést adunk rá. Az `ignoreEstimatedCount=true` beállítással ez a becslés számolás letiltható, és az eredmények számát mindenképpen újraszámoljuk, de ez hosszú ideig is eltarthat százezres vagy milliós elemszámmal, ezért használata csak indokolt esetben javasolt.
- `ignoreTotal`: Az `ignoreTotal=true` beállítással letiltható a lekérdezés eredménylista elemszámának kiszámolása.
- `previousTotal`: az `ignoreTotal=true` -val együtt használható, és egy előzőleg kiszámolt total count értéket lehet megadni a `previousTotal` paraméterben. Erre ekkor lehet csak szükség, ha pl. előzőleg kiszámoltattunk egy egzakt totál értéket
(`ignoreCachedCount=true&ignoreEstimatedCount=true`), majd ezzel a total count értékkel kívánjuk a lapozást végezni. Erre csak indokolt és ritka esetben lehet szükség, használata nem javasolt.
- `countOnly`: `countOnly=true` esetében a lekérés csak az eredmény lista darabszámát adja vissza (a paging mezőben), de magát az eredménylistát nem. Ez a `ignoreCachedCount=true&ignoreEstimatedCount=true` paraméterekkel együtt lehet hasznos, ha csak egy adott lekérdezés eredményének tényleges elemszámára vagyunk kíváncsiak, de maga az eredménylista nem szükséges (pl. ha csak statisztikákat akarunk készíteni).

A fentiekén kívül egyéb paramétereket is definiálhatnak és értelmezhetnek az egyes erőforrások.

Lehetséges HTTP válaszkódok:

- 200 OK: a keresés sikeres volt
- 401 Unauthorized: a hívó fél nem azonosította magát (nincs belépve)
- 404 Not Found: nem létező resource-ra vonatkozó kérés
- 405 Method Not Allowed: a resource-on az adott HTTP művelet nem engedélyezett (kivételes esetben használt kód)

Módosítás

Adott rekord módosítása HTTP PUT vagy PATCH metódussal történhet. A MyCite2 -ben a PUT művelet lényegében PATCH-ként működik a kezelt rekordtípusokra, vagyis csupán azokat a mezőit módosítja, amiket a bemeneti JSON-ban megadtak, lásd 2.2 Műveletek.

A visszaadott JSON a módosított rekord teljes leírását tartalmazza.

Lehetséges HTTP válaszkódok:

- 200 OK: a módosítás megtörtént és sikeres volt
- 400 Bad Request: hibás input JSON
- 401 Unauthorized: a hívó fél nem azonosította magát (nincs belépve)
- 403 Forbidden: a hívó félnek nincs jogosultsága a művelethez
- 404 Not Found: nem létező resource-ra vonatkozó kérés

- 405 Method Not Allowed: a resource-on az adott HTTP művelet nem engedélyezett (kivételes esetben használt kód)
- 409 Conflict: a módosítás a kért paraméterekkel nem végezhető el, például, mert közben más már a kéréssel nem kompatibilis módon változtatta az rekordot.
- 500 Internal Server Error: nem várt hiba lépett fel a kérés teljesítése közben

Törlés

Egyes rekordok HTTP DELETE művelettel törölhetők. Törlés után az erőforrás URL-je a továbbiakban nem lesz elérhető, arra a szerver 404-es hibakódot ad.

Lehetséges HTTP válaszkódok:

- 204 No Content: a törlés sikeres volt, a válasz nem tartalmaz semmilyen tartalmat
- 401 Unauthorized: a hívó fél nem azonosította magát (nincs belépve)
- 403 Forbidden: a hívó félnek nincs jogosultsága a művelethez
- 404 Not Found: nem létező resource-ra vonatkozó művelet
- 405 Method Not Allowed: a resource-on az adott HTTP művelet nem engedélyezett (kivételes esetben használt kód)
- 500 Internal Server Error: nem várt hiba lépett fel a kérés teljesítése közben

3.2 A rendszerben általánosan kezelt erőforrások

Az ebben a fejezetben ismertetett erőforrásokra az előző két fejezetben definiált műveletek mindegyike azonos módon értelmezett, ugyanolyan módon hajtható végre.

A MyCite2 rendszerben kezelt erőforrások a rendszer rekordjai. Az egyes API végpontok egy-egy rekord típus (plusz esteleg annak altípusainak) kezelésére szolgálnak. Pl. a

`/api/publication`

végpont a Publication és annak altípusainak (Book, BookChapter, Thesis, Patent, stb.) kezelését szolgálja. Általános szabályként elmondható, hogy minden a rendszerben tárolt és felhasználók által megtekinteni vagy kezelni engedett rekordtípushoz tartozik egy ilyen API végpont, aminek útvonala az rekord típusa nevéből származtatott a típusnév kisbetűsítésével. Pl.

Publication	<code>/api/publication</code>
Author	<code>/api/author</code>
Journal	<code>/api/journal</code>

A rendszerben található rekordokról, azok adattartamáról és működéséről a MyCite2 api-docs dokumentáció szolgáltató bővebb információt.

3.2.1 A rendszer fő erőforrásai

Közlemény

URL: `/api/publication`

A közlemény erőforrás kezeli a közlemények összességét függetlenül azok típusától. Tehát a `/publication` alatt kezelhetők a folyóiratcikkek, könyvek, könyvfejezetek, disszertációk, stb.

Szerző

URL: `/api/author`

A szerző az MyCite2 által nyilvántartott, közleményekkel rendelkező személy, aki az MyCite2 használatára jogosult.

Folyóirat

URL: `/api/journal`

Folyóirat cikk típusú közlemény folyóiratainak kezelője.

Szervezet

URL: `/api/institute`

Az MyCite2-ben nyilvántartott különféle típusú szervezetek kezelője.

Konferencia

URL: `/api/conference`

A közlemények konferencia megjelenéseit kezelő erőforrás. Egy közlemény egy konferencián jelenhet meg, tehát egy közleményhez kapcsolódóan egy konferencia hivatkozás létezhet.

Pályázat

URL: `/api/funding`

A közlemény megjelenését támogató pályázatot kezelő erőforrás.

Kiadó

URL: `/api/publisher`

Közlemények kiadóinak kezelője.

Mentett lekérdezések

URL: `/api/smartquery`

Az MyCite2 egyes rekordaira vonatkozó lekérdezések (query-k) tárolására és futtatására szolgáló erőforrás.

4 Mycite kereső operátorok és speciális keresőmezők

Az alább Mycite2 keresési célú URL-jeiben a `cond=...` mezőben megadható, adattípusonkénti kereső operátorokat, illetve egyes típusok speciális kereső mezőit ismertetjük. Ezeknek a táblázatoknak az aktuális változata mindig elérhető a

<https://m2.mtmt.hu/api-docs/query-ops>

címen. Az itt ismertetett információk mind megtalálhatók az <https://m2.mtmt.hu/api-docs> leírásban is JSON foarmátumban.

Általános kereső operátorok mezők típusa szerint

Operátor	Címke	Leírás	Paraméterek
Szöveg			
prefix	Szöveg eleji egyezés	Igaz, ha a megadott szöveg egyezik a keresett szöveg elejével	Az illesztendő szöveg
postfix	Szöveg végi egyezés	Igaz, ha a megadott szöveg egyezik a keresett szöveg végével	Az illesztendő szöveg
any	Bármilyen egyezés	Igaz, ha a megadott szöveg bárhol megtalálható a keresett szövegben	Az illesztendő szöveg
anyw	Bármilyen egyezés szavanként	Igaz, ha a megadott szöveg bárhol megtalálható a keresett szöveg bármelyik szavában	Az illesztendő szöveg
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték
eqw	Teljes egyezés szavanként	Igaz, ha a megadott szó teljes egyezéssel megtalálható a szövegben, vagyis ha a szó pont így szerepel a szövegben	Az illesztendő szöveg
ne	Nem teljes egyezés	Igaz, ha a szöveg nem egyezik megadottal	A keresett érték
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-

notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-
Szám			
eq	=	Igaz, ha az érték megegyezik a megadottal, vagy többes értékű mező esetén tartalmazza a megadottat	A keresett érték
ne	!=	Igaz, ha az érték nem egyezik meg a megadottal	A keresett érték
lt	<	Igaz, ha az érték kisebb a megadottnál	A keresett érték
le	<=	Igaz, ha az érték kisebb, vagy egyenlő a megadottal	A keresett érték
gt	>	Igaz, ha az érték nagyobb a megadottnál	A keresett érték
ge	>=	Igaz, ha az érték nagyobb, vagy egyenlő a megadottal	A keresett érték
range	-tól, -ig	Igaz, ha a numerikus érték a megadott alsó (-tól) és felső (-ig) értékek közé esik	Kezdő érték, Vég érték
in	Tartalmazza	Igaz, ha a vesszővel felsorolt numerikus értékek bármelyikét tartalmazza	A keresett értékek
nin	Nem tartalmazza	Igaz, ha a vesszővel felsorolt numerikus értékek bármelyikét nem tartalmazza	A keresett értékek
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-
notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-
Dátum/idő			
eq	=	Igaz, ha a dátum megegyezik a megadottal	A keresett érték
ne	!=	Igaz, ha a dátum nem egyezik meg a megadottal	A keresett érték
lt	<	Igaz, ha a dátum korábbi a megadottnál	A keresett érték

le	<=	Igaz, ha a dátum korábbi, vagy egyenlő a megadottal	A keresett érték
gt	>	Igaz, ha az dátum későbbi a megadottnál	A keresett érték
ge	>=	Igaz, ha a dátum későbbi, vagy egyenlő a megadottal	A keresett érték
range	-tól, -ig	Igaz, ha a dátum a megadott alsó (-tól) és felső (-ig) értékek közé esik	Kezdő dátum, Vég dátum
in	Tartalmazza	Igaz, ha a vesszővel felsorolt dátumok bármelyikét tartalmazza	A keresett értékek
nin	Nem tartalmazza	Igaz, ha a vesszővel felsorolt numerikus értékek bármelyikét nem tartalmazza	A keresett értékek
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-
notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-
Logikai érték (igaz/hamis)			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
ne	!=	Igaz, ha az érték nem egyezik meg a megadottal	A keresett érték
Érték halmaz			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
ne	!=	Igaz, ha az érték nem egyezik meg a megadottal	A keresett érték
lt	<	Igaz, ha az érték kisebb a megadottnál	A keresett érték
le	<=	Igaz, ha az érték kisebb, vagy egyenlő a megadottal	A keresett érték
gt	>	Igaz, ha az érték nagyobb a megadottnál	A keresett érték
ge	>=	Igaz, ha az érték nagyobb, vagy egyenlő a megadottal	A keresett érték

range	-tól, -ig	Igaz, ha a numerikus érték a megadott alsó (-tól) és felső (-ig) értékek közé esik	Kezdő érték, Vég érték
in	Tartalmazza	Igaz, ha a vesszővel felsorolt értékek bármelyikét tartalmazza	A keresett értékek
nin	Nem tartalmazza	Igaz, ha a vesszővel felsorolt numerikus értékek bármelyikét nem tartalmazza	A keresett értékek
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-
notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-
Rekord típus			
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-
notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-
in	Tartalmazza	Igaz, ha a vesszővel felsorolt objektumok bármelyikét tartalmazza	A keresett MTMT azonosítók
nin	Van kapcsolt rekord, ami nem tartalmazza	Igaz, ha a vesszővel felsorolt MTMT azonosítók bármelyikét nem tartalmazza a kapcsolt rekordok valamelyike. Figyelem: ez a "Nem tartalmazza" operátortól eltérő működést eredményez. Az esetek többségében a "Nem tartalmazza" operátort szükséges használni.	A keresett MTMT azonosítók
ninatall	Nem tartalmazza	Igaz, ha van olyan elem, ami egyiket sem tartalmazza, de nincs másik olyan elem, ami valamelyiket mégis tartalmazná.	A keresett MTMT azonosítók
inall	Mindent tartalmazza	Igaz, ha minden felsorolt elemet tartalmaz	A keresett MTMT azonosítók
eq	=	Igaz, ha az objektum megegyezik a megadottal, vagy többes értékű mező esetén tartalmazza a megadottat	A keresett érték

ne	!=	Igaz, ha az objektum nem egyezik meg a megadottal	A keresett érték
prefix	Címke eleji egyezés	Igaz, ha a megadott szöveg egyezik a rekord címke szövegének elejével	Az illesztendő szöveg
postfix	Címke végi egyezés	Igaz, ha a megadott szöveg egyezik a rekord címke szövegének végével	Az illesztendő szöveg
any	Címke bármilyen egyezés	Igaz, ha a megadott szöveg bárhol megtalálható a rekord címke szövegében	Az illesztendő szöveg
anyw	Címke bármilyen egyezés szavanként	Igaz, ha a megadott szöveg bárhol megtalálható a rekord címke szövegének bármelyik szavában	Az illesztendő szöveg
eqw	Címke teljes egyezés szavanként	Igaz, ha a megadott szó teljes egyezéssel megtalálható a rekord címke szövegében, vagyis ha a szó pont így szerepel a rekord címke szövegében	Az illesztendő szöveg
Rekord lista			
only	Kizárólag	Igaz, ha a mezőhöz csupán egyetlen MTMT azonosító érték van beállítva, és ez az MTMT azonosító megegyezik a feltételben beállított értékkel	A keresett érték
Intézmény			
inia	Intézmény vagy előd	Igaz, ha az adott intézmény vagy annak valamelyik előd intézménye szerepel a megadott intézmény ID listában	A keresett MTMT azonosítók
inic	Intézmény vagy alintézménye	Igaz, ha az adott intézmény vagy annak gyerek intézménye szerepel a megadott intézmény ID listában	A keresett MTMT azonosítók
inica	Intézmény vagy elődje/alintézménye	Igaz, ha az adott intézmény vagy annak gyerek intézménye, illetve ezek elődintézményei szerepelnek a megadott intézmény ID listában	A keresett MTMT azonosítók

ninia	Nem intézmény vagy előd	Igaz, ha az adott intézmény vagy annak valamelyik előd intézménye nem szerepel a megadott intézmény ID listában	A keresett MTMT azonosítók
ninic	Nem intézmény vagy alintézménye	Igaz, ha az adott intézmény vagy annak gyerek intézménye nem szerepel a megadott intézmény ID listában	A keresett MTMT azonosítók
ninica	Nem intézmény vagy elődje/alintézménye	Igaz, ha az adott intézmény vagy annak gyerek intézménye, illetve ezek elődintézményei nem szerepelnek a megadott intézmény ID listában	A keresett MTMT azonosítók
Szerző			
isme	Én vagyok	A szerző én vagyok (a jelenleg belépett szerző felhasználó)	-
isnotme	Nem én vagyok	A szerző nem én vagyok (a jelenleg belépett szerző felhasználó)	-
coAuthorsOf	Társszerzői	Adott szerző társszerzői	A keresett MTMT azonosítók
coAuthorsOfMe	Társszerzőim	Jelenleg belépett szerző társszerzői	-
Felhasználó			
isme	Én vagyok	A szerző én vagyok (a jelenleg belépett szerző felhasználó)	-
Szerzőség			
authorNameIn	Bármelyik nevet tartalmazza	Igaz, ha a szerző neve megegyezik valamelyik megadott családnév-keresztnev párral	A keresett érték
authorNameAll	Minden nevet tartalmaz	Igaz, ha a szerzők között mindegyik felsorolt családnév-keresztnev pár megtalálható	A keresett érték
Tudományosztályozás - Frascati			
incc	Tudományterülethez tartozik	Igaz, ha a keresett tudományterület az adott tudományterülethez vagy alá	A keresett MTMT azonosítók

		tartozik	
DuplumKeresés			
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-
notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-

Speciális keresőmezők és operátoraik minden rekordtípusnál

Operátor	Címke	Leírás	Paraméterek
Adatgazda (Érték halmaz)			
eq	=	Igaz, ha értéke ME és az Adatgazda a jelenleg belépett felhasználó, vagy ha értéke egy MTMT azonosító és ez megegyezik egy beállított Adatgazda értékével (adott adatgazdára keresés)	A keresett érték
Forrásközlemény adatgazda (Érték halmaz)			
eq	=	Adatgazda	A keresett érték
Címke vagy azonosító (Szöveg)			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték
Érvényességi idő (Egész szám)			
eq	=	Igaz, ha az érték megegyezik a megadottal, vagy többes értékű mező esetén tartalmazza a megadottat	A keresett érték
range	-tól, -ig	Igaz, ha a numerikus érték a megadott alsó (-tól) és felső (-ig) értékek közé esik	Kezdő érték, Vég érték
in	Tartalmazza a	Igaz, ha a vesszővel felsorolt numerikus értékek bármelyikét tartalmazza	A keresett érték
"Nem duplum"-nak jelölt (Logikai érték (igaz/hamis))			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Önmaga (Az aktuális rekordtípus)			
isnull	Üres	Igaz, ha a mező értéke üres (nincs beállítva)	-

notnull	Nem üres	Igaz, ha a mező értéke nem üres (be van állítva)	-
in	Tartalmazza	Igaz, ha a vesszővel felsorolt objektumok bármelyikét tartalmazza	A keresett MTMT azonosítók
nin	Van kapcsolt rekord, ami nem tartalmazza	Igaz, ha a vesszővel felsorolt MTMT azonosítók bármelyikét nem tartalmazza a kapcsolt rekordok valamelyike. Figyelem: ez a "Nem tartalmazza" operátortól eltérő működést eredményez. Az esetek többségében a "Nem tartalmazza" operátort szükséges használni.	A keresett MTMT azonosítók
ninatal	Nem tartalmazza	Igaz, ha van olyan elem, ami egyiket sem tartalmazza, de nincs másik olyan elem, ami valamelyiket mégis tartalmazná.	A keresett MTMT azonosítók
inall	Mindent tartalmazza	Igaz, ha minden felsorolt elemet tartalmaz	A keresett MTMT azonosítók
eq	=	Igaz, ha az objektum megegyezik a megadottal, vagy többes értékű mező esetén tartalmazza a megadottat	A keresett érték
ne	!=	Igaz, ha az objektum nem egyezik meg a megadottal	A keresett érték
prefix	Címke eleji egyezés	Igaz, ha a megadott szöveg egyezik a rekord címke szövegének elejével	Az illesztendő szöveg
postfix	Címke végi egyezés	Igaz, ha a megadott szöveg egyezik a rekord címke szövegének végével	Az illesztendő szöveg
any	Címke bármilyen egyezés	Igaz, ha a megadott szöveg bárhol megtalálható a rekord címke szövegében	Az illesztendő szöveg
anyw	Címke bármilyen egyezés szavanként	Igaz, ha a megadott szöveg bárhol megtalálható a rekord címke szövegének bármelyik szavában	Az illesztendő szöveg
eqw	Címke teljes egyezés szavanként	Igaz, ha a megadott szó teljes egyezéssel megtalálható a rekord címke szövegében, vagyis ha a szó pont így szerepel a rekord címke szövegében	Az illesztendő szöveg

Speciális keresőmezők és operátoraik egyes rekordtípusoknál

Operátor	Címke	Leírás	Paraméterek
Közlemény: authorCandidate - Lehetséges szerző (Szöveg)			
eq	=	Igaz, ha értéke ME és az Adatgazda a jelenleg belépett felhasználó, vagy ha értéke egy MTMT azonosító és ez megegyezik egy beállított Adatgazda értékével (adott adatgazdára keresés)	A keresett érték
Közlemény: publicationRole - Szerepe (Érték halmaz)			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Szerzőség: firstOrLast - Első vagy utolsó szerző (Logikai érték (igaz/hamis))			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Szerző: active - Aktív (Logikai érték (igaz/hamis))			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték
Intézmény: mine - Hozzám rendelt intézmény (Érték halmaz)			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Intézmény: mineInica - Hozzám rendelt intézmény vagy elődje/alintézménye (Érték halmaz)			
eq	=	Igaz, ha az érték megegyezik a megadottal	-
Intézmény: mineInia - Hozzám rendelt intézmény vagy elődje (Érték halmaz)			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Intézmény: mineInic - Hozzám rendelt intézmény vagy alintézménye (Érték halmaz)			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Intézmény: affiliations - Affiliációk ()			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték
Felhasználó: active - Aktív (Logikai érték (igaz/hamis))			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték

Intézményi szerzőség: firstOrLast - Első vagy utolsó szerző (Logikai érték (igaz/hamis))			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Személyi szerzőség: firstOrLast - Első vagy utolsó szerző (Logikai érték (igaz/hamis))			
eq	=	Igaz, ha az érték megegyezik a megadottal	A keresett érték
Admin: mine - Saját (Logikai érték (igaz/hamis))			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték
Admin: adminOf - Szerző adminja ()			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték
Admin: active - Aktív (Logikai érték (igaz/hamis))			
eq	Teljes egyezés	Igaz, ha a szöveg teljesen megegyezik a megadottal	érték