



Quansic ISNI API Guideline

Date: Feb. 2020

Version: 2.0

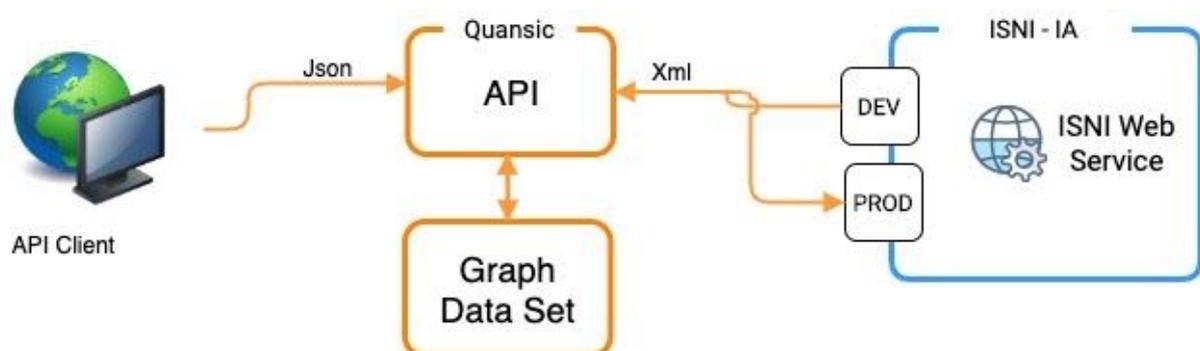
Status: **prod**

This is the guideline for the implementation of Quansic's ISNI APIs V 2.0 to request a new ISNI assignment to contributors.

The full reference is available at: <https://api.quansic.com/api-docs>

Architecture

The Quansic API offers an abstraction of the official ISNI assign API, tailored for the Music Industry. It is based on a familiar data model and terminology, making it easy to use in your environment.



End point

The ISNI search api allows you to request a new ISNI for a contributor., such as a Performer or a Songwriter.

The endpoint is:

```
https://api.quansic.com/api/v2/isni
```

The method is POST.

Encoding is UTF-8.

Authentication

The authentication is common to all API calls.

Client

The first step is to authenticate your client.

Quansic provides a unique string for each client application.

You can test your client and review its status by going to

```
https://api.quansic.com/api/v2/isni/clients/{clientId}
```

You will receive the following response

```
{
  "client": {
    "name": "Best Record label",
    "key": "W5awRig4t1B8fetx",
    "status": "valid",
    "validTo": "20210214T00:00:00",
    "vouchers": 9
  }
}
```

Where

- Name: The name of the company owning the key
- Key: The key itself
- Status: the status of the key. Values are "valid" or "pending".

- ValidTo : the expiry date of the key
- Vouchers: The number of vouchers valid on the key account.

Mode

You can call the API in 2 modes, “dev” or “prod”.

- “dev”: To familiarize yourself with data submission, use the “dev” mode. It’s free. The process is as follows
 - The API first checks for an obvious match against Quansic’s data set.
 - If there is a match, the resulting record is returned.
 - If there is no match, the query is sent to ISNI-IA in “dev” mode where a ISNI request is simulated. If successful, an ISNI is returned. **DO NOT USE those dev IDs.**
- “prod”: When you are comfortable with your setup, you can switch to “prod”. A voucher is required to call the API in “prod” mode. The process is as follows:
 - The API first checks for an obvious match against Quansic’s data set.
 - If there is a match, the resulting record is returned. The voucher is not deactivated.
 - If there is no match, the query is sent to ISNI-IA in “prod” mode where a ISNI request is generated. If successful, an ISNI is returned. The voucher is deactivated, regardless of the result.

Use the “prod” mode responsibly and professionally. You are populating live databases that are used by the rest of the Music Industry.

Voucher

The voucher is a unique code valid for one transaction.

They are only used in PROD mode when requesting a new ISNI assignment.

Vouchers must be purchased and active in your API account for a ISNIassign request to succeed.

When running in “prod” mode, the voucher is verified and deactivated after the transaction, if a call has been made to the ISNI-IA API.

Always set your voucher to “auto”. This means that the voucher will automatically be pulled from your account when making a query.

For more information on vouchers go to quansic.com/vouchers.html

```
{
  "client" : "W5awRig4t1B8fetx",
  "voucher" : "auto",
  "mode" : "dev",
  ...
}
```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
client	[1-1]	Mandatory	String identifying the client. Supplied by Quansic.
mode	[1-1]	Mandatory	Specifies whether the API call is in test mode ("dev") or in production mode ("prod"). In "dev" mode, the returned values are transient. Do not use them.
voucher	[0-1]	Mandatory	Set to auto

Encoding

It is recommended to provide all titles and names in Camel Case. The first letter of each word is Uppercase, the rest in lower case.

Any UTF-8 character is accepted but try to avoid technical characters such as slash "/", backslash "\", all parentheses "(){}[]", or quotes "", as they are not always properly processed by all systems.

Recommended	NOT recommended
Jeremy Docato	JEREMY DUCATO
Herbert Von Karajan	H. Von Karajan
The Red Brothers	The red BROTHERS
The Slash Band	The / band

Sections

The request is a json object with the authentication header and 6 sections: entity, name, nameVariants, contributions, identifiers, relationships.

```
{
  "client" : "myClientID",
  "voucher" : "1234-8743-2345",
  "mode" : "prod",
```

```

"entity": {...},

"name" : {...},

"nameVariants" : [...],

"identifiers" : [...],

"contributions" : [.. ],

"relationships" : [...]

```

Entity

The entity section provides details about the entity for which an ISNI is requested.

```

{
  "entity": {
    "type" : "person",
    "birthdate" : "1959-09-28",
    "deathdate" : "2018-12-17",
    "gender" : "male",
    "urls" : ["https://www.wikidata.org/wiki/Q2831"],
    "countries" : ["FR", "GB"]
  }
}
[..]

```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
type	[1]	Mandatory	Specifies whether the ISNI request is for a person or a band.
firstname, lastname	[1-1]	Mandatory	lastname is mandatory. For a band or a mononym (name with a single word) use the lastname element.
birthdate	[0-1]	Optional	Date may be provided as YYYY, YYYY-MM or YYYY-MM-DD

deathdate	[0-1]	Optional	Date may be provided as YYYY, YYYY-MM or YYYY-MM-DD
gender	[0-1]	Optional	Allowed values are [male, female, none]. For bands, use none.
countries	[0-n]	Optional	A list of 2-letter ISO 3166 country codes. Typical value is the nationality of the entity.
urls	[0-n]	Optional	A list of urls relevant to the entity. The urls that are checked by ISNI-IA are <ul style="list-style-type: none"> - discogs url - wikipedia url - Wikidata url - Artist official website

Note: the birthdate is highly recommended, as it is a key match criteria.

Name

The name section provides details about the primary name of the entity for which an ISNI is requested.

The name section is Mandatory.

```
{
  "name": {
    "firstname" : "Phil",
    "lastname"  : "Collins",
    "nametype"  : "realname"
  }
  [...]
}
```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
firstname	[0-1]	Optional	First name of the entity.
lastname	[1]	Mandatory	lastname is mandatory. For a band or a mononym (name with a single word) use the lastname element.

nametype	[1]	Mandatory	Allowed values are [realname, pseudonym]. For a band, use realname.
----------	-----	-----------	--

Name variants

The name variants section provides a list of name variations on the primary name of the entity, as specified in the `name` section.

Name variants can be different spellings of the name or transliterations in other character sets or other languages.

The name variants section is Optional.

```
{
  "namevariants": [
    {
      "firstname" : "Phillip",
      "lastname"  : "Collins",
      "nametype"  : "realname"
    },
    {
      "firstname" : "Phillip David Charles",
      "lastname"  : "Collins",
      "nametype"  : "realname"
    },
    {
      "lastname"  : "Фил Колинс",
      "nametype"  : "realname"
    }
  ]
  [...]
}
```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
firstname	[0-1]	Optional	First name of the entity.
lastname	[1]	Mandatory	lastname is mandatory. For a band or a mononym (name with a single word) use the lastname element.

nametype	[1]	Mandatory	The namevariant.nametype must be the same as the name.nametype entity from the name section. Allowed values are [realname, pseudonym]. For a band, use realname.
----------	-----	-----------	--

Note: If the entity nametype is set "realname", a pseudonym is NOT a name variant. In this case, the pseudonym must be provided in the relationship section. See "rules for ISNI request" below for more details.

Identifiers

The Identifiers section provides a list of known identifiers already assigned to the entity. The identifiers section is Optional.

```
{
  "identifiers": [
    {
      "key" : "IPI",
      "value" : "00121924204"
    },
    {
      "key" : "MBID",
      "value" : "401c3991-b76b-499d-8082-9f2df958ef78"
    }
  ]
}
[...]
```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
key	[1]	Mandatory	The allowed values are: [IPI, IPN, MBID]
value	[1]	Mandatory	The value of the identifier.

Note:

- IPI: Interested Party Identifier
- IPN: International Performer Number
- MBID: Musicbrainz artist identifier

Contributions

The Contributions section provides a list of Sound Recordings or Musical Works to which the entity has contributed.

The contributions section is Optional.

```
{
  "contributions": [
    {
      "title" : "Swing Low",
      "identifier" : {
        "type" : "ISRC",
        "value" : "USW110201016"
      },
      "role" : "performer"
    },
    {
      "title" : "Driving Me Crazy",
      "identifier" : {
        "type" : "ISWC",
        "value" : "T0101444669"
      },
      "role" : "author"
    },
    {
      "title" : "Can't Find My Way",
      "identifier" : {
        "type" : "ISRC",
        "value" : "USAT21503053"
      },
      "role" : "performer"
    }
  ]
}
[...]
```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
title	[1]	Mandatory	The title of the contribution.

role	[0-1]	Optional	The role of the entity. Allowed values are [author, composer, performer]. For bands, use performer.
identifier	[0-1]	Optional	The identifier of the contribution. Allowed values are [ISRC, ISWC]

Note: The number of contributions is not limited but it is good practise to provide at least 3 contributions for each request.

The identifier can be provided either in machine readable or human readable formats.
ie. "T0101444669" or "T-010.144.466-9" are both valid.

Relationships

The Relationships section provides a list of relationships between the entity for which an ISNI is requested and other entities.

Relationships can be used to indicate that a Person is member of a band, is a co-author with another person, or the entity is also known under a pseudonym

The Relationships section is Optional.

```
{
  "relationships": [
    {
      "name": {
        "lastname" : "Genesis",
        "nametype" : "realname"
      },
      "identifier" : {
        "type" : "ISNI",
        "value" : "000000007949869X"
      },
      "type" : "isMember"
    },
    {
      "name": {
        "lastname" : "Phil Collins Big Band",
        "nametype" : "realname"
      },
      "identifier" : {
        "type" : "ISNI",
        "value" : "0000000083377639"
      },
      "type" : "isMember"
    }
  ]
}
```

[...]

```
}
```

Specification for each element:

Element	Cardinality	Optional/Mandatory	Description
<code>name.firstname</code>	[0-1]	Mandatory	The first name of the related entity
<code>name.lastname</code>	[1]	Mandatory	<code>lastname</code> is mandatory. For a band or a mononym (name with a single word) use the <code>lastname</code> element.
<code>name.nametype</code>	[1]	Mandatory	The name type of the related entity Allowed values are [<code>realname</code> , <code>pseudonym</code>]. For a band, use <code>realname</code> .
<code>identifier.type</code>	[0-1]	Optional	The type of identifier provided for the related entity. Allowed values are [<code>ISNI</code> , <code>IPI</code> , <code>IPN</code>].
<code>identifier.value</code>	[0-1]	Optional	The value of the identifier.
<code>type</code>	[1]	Mandatory	The type of relationship between the entity and the related entity. Allowed values are [<code>co-author</code> , <code>co-performer</code> , <code>pseudonym</code> , <code>hasMember</code> , <code>isMemberOf</code> , <code>realname</code>]

Note: Providing an identifier in the relationship element considerably slows down the ISNI-IA matching algorithm. We recommend that you do NOT provide any identifier in this section. See “Rules for ISNI request” below for more details.

Typical usages are:

- Declare the pseudonym of a person and a co-author:

```
{
  "entity": {
    "type" : "person",
    "name" : {
      "firstName" : "Abel",
      "lastName" : "Tesfaye",
      "nameType" : "realName"
    },
    [...]
    "relationships" : [
```

```

    {
      "type" : "pseudonym",
      "name": {
        "lastname" : "The Weeknd",
        "nametype" : "pseudonym"
      },
    },
    {
      "type" : "co-author",
      "name": {
        "firstname" : "Edward",
        "lastname" : "Sheeran",
        "nametype" : "realname"
      }
    }
  ]
}

```

- Declare the members of a band:

```

{
  "entity": {
    "type" : "band",
    "name" : {
      "lastName" : "Genesis",
      "nameType" : "realName"
    },
  },
  [...]
  "relationships" : [
    {
      "type" : "hasMember",
      "name": {
        "firstname" : "Phil",
        "lastname" : "Collins",
        "nametype" : "realname"
      },
    },
    {
      "type" : "hasMember",
      "name": {
        "firstname" : "Peter",
        "lastname" : "Gabriel",
        "nametype" : "realname"
      },
    },
  ],
}
]

```

Rules for ISNI Request

To maximize your success rate, we **STRONGLY** recommend the following rules when creating your data set.

Minimum Data Set

Some artist names are considered as “common” when there are many different entities that carry the same name. To disambiguate amongst all the homonyms, it is important to provide additional data elements.

It is recommended to provide at least:

- A name
- A date of birth
- 3 or more contributions
- A relationship
- A url

Either a birthdate or a list of contributions is mandatory.

Identifiers

When you provide identifiers in the `relationship.identifier` section, it adds strong constraints on the matching logic.

We recommend that you do NOT supply any identifier in the `relationship.identifier` section.

Real names and pseudonyms

If an entity has both a real name and a pseudonym, we recommend that you request 2 ISNIs - one for the real name and one for the pseudonym - and that you link them.

Here is the process:

- Request an ISNI for the entity real name.
 - In the `name` section, enter the real name.
 - Set `name.nametype` to `realname`
 - Create a relationship, with the pseudonym in the `relationships.name` element
 - Set `relationships.type` to `pseudonym`
- Request a second ISNI for the entity pseudonym.
 - In the `name` section, enter the pseudonym.
 - Set `name.nametype` to `pseudonym`
 - Create a relationship, with the real name in the `relationships.name` element
 - Set `relationships.type` to `realname`

The two ISNIs will be linked in the ISNI database with a “same person, other identity” relation.

This is the result in the ISNI database.

ISNI:	0000 0004 6595 4622
Name:	Malice No Malice
Creation class:	Musical sound recording
Creation role:	creator performer
Related identities:	Malice (pseud) Thornton, Gene (real name)

Ephemeral duos

Some duos are considered ephemeral. They are usually represented as

artist A featuring artist B

Those duos are not eligible for an ISNI.

We suggest the following method:

- Request an ISNI for the first entity's name. (Artist A).
 - Create a relationship by setting `relationships.type` to `co-performer` with Artist B
- Request a second ISNI for the second entity (Artist B).
 - Create a relationship with `relationships.type` to `co-performer` with Artist A

Relationships

Relationships are not provided to describe the life of the Entity, but to help the matching algorithm disambiguate the name. Whenever possible, provide 2 or more relationships.

Examples

Example of a full `isniAssign` request payload for a person:

```
{
  "mode": "prod",
  "key": "W5awRig4t1B8fetx",
  "voucher": "auto",
  "entity": {
    "birthdate": "1959-09-28",
    "gender": "male",
    "type": "person",
    "countries": [
      "DE"
    ],
    "name": {
      "lastname": "Le Monde",
```

```

    "firstname": "Adrian",
    "nametype": "pseudonym"
  },
  "contributions": [
    {
      "identifier": {
        "type": "ISWC",
        "value": "T-803.044.027-9"
      },
      "role": "author",
      "title": "Drunk Again"
    },
    {
      "identifier": {
        "type": "ISWC",
        "value": "T-801.332.782-8"
      },
      "role": "author",
      "title": "Nostromo"
    },
    {
      "identifier": {
        "type": "ISWC",
        "value": "T-801.287.148-7"
      },
      "role": "author",
      "title": "Swanlake"
    },
    {
      "identifier": {
        "type": "ISWC",
        "value": "T-802.489.150-8"
      },
      "role": "author",
      "title": "My Monkey"
    },
    {
      "identifier": {
        "type": "ISWC",
        "value": "T-803.044.130-7"
      },
      "role": "author",
      "title": "Cartoonized"
    },
    {
      "role": "author",
      "title": "Drunk Again"
    },
    {
      "role": "author",
      "title": "Ambivalence"
    }
  ],

```

```

    "identifiers": [
      {
        "type": "IPI",
        "value": "00259024273"
      },
      {
        "type": "IPI",
        "value": "00259024371"
      },
      {
        "type": "MBID",
        "value": "0af8a451-ca01-41f7-b8af-aecb5f78e6ba"
      }
    ],
    "relationships": [
      {
        "type": "co-author",
        "name": {
          "lastname": "Moro",
          "firstname": "Brit",
          "nametype": "realname"
        }
      },
      {
        "type": "isMemberOf",
        "name": {
          "lastname": "Raven Parque",
          "nametype": "realname"
        }
      }
    ]
  }
}

```

A “rich” Json payload would look like this for a band:

```

{
  "mode": "prod",
  "key": "W5awRig4t1B8fetx",
  "voucher": "auto",
  "entity": {
    "birthdate": "1982",
    "gender": "none",
    "type": "band",
    "countries": [
      "DE"
    ],
    "urls":
["https://musicbrainz.org/artist/e3832020-6413-4fbf-9fea-98fe6362
ba6d"],

```

```

"name": {
  "lastname": "Raven Parque",
  "nametype": "realname"
},
"contributions": [
  {
    "identifier": {
      "type": "ISRC",
      "value": "QZV121900219"
    },
    "role": "performer",
    "title": "Drunk Again"
  },
  {
    "identifier": {
      "type": "ISRC",
      "value": "QZV121900220"
    },
    "role": "performer",
    "title": "Colors"
  },
  {
    "identifier": {
      "type": "ISRC",
      "value": "QZV121900232"
    },
    "role": "performer",
    "title": "My Vision Thing"
  },
  {
    "identifier": {
      "type": "ISRC",
      "value": "QZV121900237"
    },
    "role": "performer",
    "title": "My Monkey"
  }
],
"identifiers": [
  {
    "type": "MBID",
    "value": "e3832020-6413-4fbf-9fea-98fe6362ba6d"
  }
],
"relationships": [
  {
    "type": "hasMember",
    "name": {
      "firstname": "Adrian",
      "lastname": "Le Monde",
      "nametype": "pseudonym"
    }
  }
],

```

```

    {
      "type": "hasMember",
      "name": {
        "lastname": "Moro",
        "firstname": "Brit",
        "nametype": "realname"
      }
    }
  ]
}

```

Response

The response from the API to the client is as follows:

```

{
  "entity": {
    "request" : {},
    "response" : {}
  },
  "xml": {
    "request" : {},
    "response" : {}
  },
  "source": "ISNI ACC",
  "status": 200,
  "message": "an ISNI can be assigned to the provided entity"
}

```

Element	Description
entity	The message from the Quansic API
entity.request	Your original API request body
entity.response	The summary response from the Quansic API. This is where you will find the ISNI code
xml	The message from the ISNI-IA API
xml.request	The xml request that was sent to ISNI-IA
xml.response	The xml response that was received from ISNI-IA
source	The final source that processed your request

status	The final status of your request
message	The summary of your request