Bionano VIA™ KB API Guide

DOCUMENT NUMBER:
CG-00046

DOCUMENT REVISION:
A

Effective Date:
06/27/2023

bionano

Table of Contents

Legal Notice

For Research Use Only. Not for use in diagnostic procedures.

Patents

Products of Bionano Genomics® may be covered by one or more U.S. or foreign patents.

Trademarks

The Bionano logo and names of Bionano products or services are registered trademarks or trademarks owned by Bionano Genomics, Inc. (“Bionano”) in the United States and certain other countries.

Bionano™, Bionano Genomics®, Saphyr®, Saphyr Chip®, Bionano Access™, and Bionano EnFocus™ are trademarks of Bionano Genomics, Inc. All other trademarks are the sole property of their respective owners.

No license to use any trademarks of Bionano is given or implied. Users are not permitted to use these trademarks without the prior written consent of Bionano. The use of these trademarks or any other materials, except as permitted herein, is expressly prohibited and may be in violation of federal or other applicable laws.

bionano

Revision History

REVISION	NOTES
A	Initial release.

bionano

Introduction

Adding and approving Knowledge Base (KB) events is generally performed one-by-one by a user via the VIA Client UI. For users that want to enter events (e.g., legacy data or batch loading) without using the UI, a REST API may be used to add events to the KB as well as move events from **Pending** status to **Approved** status.

bionano

Summary

The steps to add approved events to the KB are as follows:

1. Use a REST API to make a `POST HTTP` request with the defined JSON payload to create a KB event with a **Pending** status.

2. Note the unique id and a revision number of the created KB event.

3. Use a REST API to make a `POST HTTP` request with the defined form parameters along with the unique ID and a revision number of the KB event to approve the pending KB event.

Connection

Connection to the VIA Server must be made using the same hostname, port, and protocol (HTTP or HTTPS) as is used for the VIA Client UI. Each of the POST/GET REST APIs mentioned in this document should be prefixed with this connection information.

```
POST protocol://server:port/…
```

E.g.,

```
POST https://localhost:8443/kb/constitutionalevent
```

bionano

Authentication

All requests are to the VIA Server over HTTP or HTTPS. Use HTTP Basic access authentication.

Username and password must match a VIA user with the following permissions:

- Ability to submit to the KB
- Ability to approve KB submissions

bionano

Adding an event

A single event can be added using a POST request. The body of the request must be JSON (see JSON Field Descriptions and examples below).

Currently, events can ONLY be added within **Pending** status. A subsequent call is required to move the events into **Approved** status.

Once an event is added, a response is returned, which will later be used to move events into **Approved** status. Please review the section **Response Data** for more information.

bionano

Constitutional Event

HTTP POST request format

```
POST protocol://server:port/kb/constitutionalevent
```

EXAMPLE REQUEST BODY - JSON

```
{
  "build": "NCBI Build 37",
  "chr": 7,
  "start": 5000000,
  "end": 6000000,
  "eventKind": "CNV",
  "event": "loss",
  "data": {
    "classification": "benign",
    "note": "example note",
    "label": "loss on chr7",
    "interpretation": "example interpretation",
    "evidenceRating": 5,
    "pubMedReferences": [{
            "pubMedId": 123456,
            "note": "example reference"
    }],
    "exampleCases": [{
            "name": "example sample",
            "note": "notes about sample"
    }],
    "phenotypes": [{
            "id": "HP00000",
            "label": "example phenotype"
    }],
    "relevantGenes": [{
            "name": "example gene name",
            "note": "note about gene"
    }],
    "inheritanceMode": ["UNKNOWN"],
    "acmgEvidenceCategories": {
            "1A": {
                    "score": 0.1,
                    "notes": "notes about acmg category"
```

```
35            }
36    },
37    "seqVarData": {
38            "ref": "A",
39            "alt": "T",
40            "transcriptId": "NM_0000000",
41            "proteinId": "",
42            "hgvs_c": "example hgvsc",
43            "hgvs_p": "example hgvsc",
44            "consequences": ["example", "consequences"]
45    }
46 }
47}
```

bionano

Oncology Event

HTTP POST request format

```
POST protocol://server:port/kb/oncologyevent
```

EXAMPLE REQUEST BODY - JSON

```
{
  "build": "NCBI Build 37",
  "chr": 1,
  "start": 5000000,
  "end": 6000000,
  "eventKind": "CNV",
  "event": "loss",
  "data": {
    "classification": "benign",
    "note": "example note",
    "label": "loss on chr1",
    "interpretation": "example interpretation",
    "pubMedInfos": [{
            "pubMedId": 123456,
            "note": "example reference",
            "diagnostic": false,
            "therapeutic": false,
            "hasGoodPrognosticOutcome": false,
            "hasBadPrognosticOutcome": false
    }],
    "exampleCases": [{
            "name": "example sample",
            "note": "notes about sample"
    }],
    "relevantGenes": [{
            "name": "example gene name",
            "note": "note about gene",
            "oncoGene": true,
            "tumorSuppressor": false
    }],
    "seqVarData": {
            "ref": "A",
            "alt": "T",
```

```
            "transcriptId": "NM_0000000",
            "proteinId": "",
            "hgvs_c": "example hgvsc",
            "hgvs_p": "example hgvsc",
            "consequences": ["example", "consequences"]
    }
  },
  "cancerTypes": {
    "WHO": ["who", "cancer"],
    "ONCOTREE": ["oncotree", "cancer"]
  }
}
```

RESPONSE DATA

Each time a new KB entry is created, it is given a unique id and a revision number. These are returned in the response body and are required to move the event into the **Approved** status:

```
{
  "uuid": "fea068ac-081d-44e5-beba-6089f4ac175a",
  "revision": 1
}
```

bionano

Moving Events to Approved Status

OBTAINING THE REVISION TOKEN

To make a change to the KB, a revision token must be obtained; the event will have to be queried from the KB to extract this information. The unique ID (Kbid.uuid) and a revision number (Kbid.revision) for the event that was added are needed for the query (see Response Data).

HTTP GET REQUEST FORMAT - CONSTITUTIONAL

```
GET protocol://server:port/kb/constitutionalevent/TO_BE_REVIEWED/Kbid.uuid/Kbid.revision
```

EXAMPLE:

```
GET http://127.0.0.1:8081/kb/constitutionalevent/TO_BE_REVIEWED/fea068ac-081d-44e5-beba-6089f4ac175a/1
```

HTTP GET REQUEST FORMAT - ONCOLOGY

```
GET protocol://server:port/kb/oncologyevent/TO_BE_REVIEWED/Kbid.uuid/Kbid.revision
```

EXAMPLE:

```
GET http://127.0.0.1:8081/kb/oncologyevent/TO_BE_REVIEWED/fea068ac-081d-44e5-beba-6089f4ac175a/1
```

Response Body Content

Once the query is made, the response body contains the following JSON:

```
{
  "regionId": {
    "uuid": "fea068ac-081d-44e5-beba-6089f4ac175a",
    "revision": 1
  },
  "build": "NCBI Build 37",
  "chr": "chr1",
  "start": 5000000,
  "end": 6000000,
  "eventKind": "CNV",
  "event": "loss",
  "data": {
    ...
  },
  "cancerTypes": {
    ...
  },
  "audit": {
```

```
19    "status": "TO_BE_REVIEWED",
20    "revisionDate": 1632271183511,
21    "revisionUser": "admin",
22    "approvalUser": null,
23    "revToken": {
24      "uuid": "851b6ea6-578a-43b4-bcc6-4d070eda9c12"
25    }
26  }
27}
```

The audit.revToken.uuid value, 851b6ea6-578a-43b4-bcc6-4d070eda9c12, will need to be extracted.

Reviewing the Event

The event can be moved to “Approved” status using a POST request with application/x-www-form-urlencoded form.

HTTP POST REQUEST FORMAT - CONSTITUTIONAL

```
POST protocol://server:port/kb/constitutionalevent/review
```

HTTP POST REQUEST FORMAT - ONCOLOGY

```
POST protocol://server:port/kb/oncologyevent/review
```

FORM PARAMETERS:

```
reviewOperation: APPROVE
id: fea068ac-081d-44e5-beba-6089f4ac175a
rev: 1
token: 851b6ea6-578a-43b4-bcc6-4d070eda9c12
```

Verifying Changes

To verify that the event was created and moved to the “Approved” status, one must search for it again via the uuid:

HTTP GET request format - Constitutional

```
GET protocol://server:port/kb/constitutionalevent/APPROVED/id/rev
```

```
Example:
```

```
GET http://127.0.0.1:8081/kb/constitutionalevent/APPROVED/fea068ac-081d-44e5-beba-6089f4ac175a/1
```

HTTP GET request format - Oncology

```
GET protocol://server:port/kb/oncologyevent/APPROVED/id/rev
```

EXAMPLE:

```
GET http://127.0.0.1:8081/kb/oncologyevent/APPROVED/fea068ac-081d-44e5-beba-6089f4ac175a/1
```

EXAMPLE RESPONSE BODY

The returned JSON data should reflect the event that was just added. If the addition/approval wasn’t successful, a 404 error will be returned.

```
{
  "regionId": {
    "uuid": "0fb927f0-82d3-44f0-afb9-628eff0733fa",
    "revision": 1
  },

…

  "audit": {
    "status": "APPROVED",
    "revisionDate": 1634217250801,
    "revisionUser": "admin",
    "approvalUser": "admin",
    "revToken": {
      "uuid": "ef177b89-45aa-4e02-b5d0-b9514fb8113f"
    }
  }
}
```

bionano

JSON Field Descriptions

Table 1. Constitutional Event JSON Fields

Field	Type	Required	Description
build	string	yes	Human genome build. Must be consistent with VIA's definition: "NCBI Build 37", etc.
chr	string	yes	
start	int	yes	
end	int	yes	
eventKind	string	yes	must be **CNV**, **ZYGOSITY**, **SEQVAR**
event	string	yes	
data	object	yes	See **ConstitutionalData** Object

***Table 2**. Oncology Event JSON Fields*

Field	Type	Required	Description
build	string	yes	Human genome build. Must be consistent with VIA's definition: "NCBI Build 37", etc.
chr	string	yes	
start	int	yes	
end	int	yes	
eventKind	string	yes	must be **CNV**, **ZYGOSITY**, or **SEQVAR**
event	string	yes	
data	object	yes	See **OncologyData** Object
cancerTypes	object	no	See **CancerTypes** Object

bionano

Sub-Object JSON Fields

Table 3. ConstitutionalData

Field	Type	Required	Description
classification	string	no	typically benign, likely benign, likely pathogenic, pathogenic, artifact, or VUS.
note	string	no	
label	string	yes	
interpretation	string	no	
evidenceRating	int	no	1 - 5
pubMedReferences	object array	no	See **PubMedInfo** Object
exampleCases	object array	no	See **ExampleCaseInfo** Object
phenotypes	object array	no	See **Phenotype** Object
relevantGenes	object array	no	See **GeneInfo** Object
inheritanceMode	string array	no	typically: “De Novo”, “Dominant”, “Recessive”, or “X-Linked”
acmgEvidenceCategories	object map	no	keys are according to ACMG guidelines: 1A, 1B, 2A, 2B, 2C-1, 2C-2, etc. See **AcmgEvidenceCategory** Object Note: this field is not displayed in the UI currently (will be added in a future version) but this info. can be added to the KB via the API.
seqVarData	object	no	Included if the event is seqvar. See **SeqVarData** Object

Table 4. OncologyData

Field	Type	Required	Description
classification	string	no	typically benign, likely benign, likely pathogenic, pathogenic, artifact, or VUS.
note	string	no	
label	string	yes	
interpretation	string	no	
pubMedInfos	object array	no	please note: the oncology pubmed info. has extra fields as compared to constitutional. See **OncoPubMedInfo** Object
exampleCases	object array	no	See **ExampleCaseInfo** Object
relevantGenes	object array	no	See **OncoGeneInfo** Object
seqVarData	object	no	See **SeqVarData** Object

Table 5. ExampleCaseInfo

Field	Type	Required	Description
name	string	yes	
note	string	yes	

Table 6. PubMedInfo

Field	Type	Required	Description
pubMedId	int	yes	
note	string	yes	

bionano

Table 7. OncoPubMedInfo

Field	Type	Required	Description
pubMedId	int	yes	
note	string	yes	
diagnostic	boolean	yes	
therapeutic	boolean	yes	
hasGoodPrognosticOutcome	boolean	yes	
hasBadPrognosticOutcome	boolean	yes	

Table 8. GeneInfo

Field	Type	Required	Description
name	string	yes	
note	string	yes	

Table 9. OncoGeneInfo

Field	Type	Required	Description
name	string	yes	
note	string	yes	
oncoGene	boolean	yes	
tumorSuppressor	boolean	yes	

Table 10. Phenotype

Field	Type	Required	Description
id	string	yes	
label	string	yes	

Table 11. AcmgEvidenceCategory

Field	Type	Required	Description
score	double	yes	
notes	string	yes	

Table 12. SeqVarData

Field	Type	Required	Description
ref	string	yes	
alt	string	yes	
transcriptId	string	no	Most refseq and ensembl transcripts are supported. VIA will not add an additional link otherwise.
proteinId	string	no	
hgvs_c	string	no	
hgvs_p	string	no	

Table 13. CancerTypes

Field	Type	Required	Description
WHO	string array	no	
ONCOTREE	string array	no	

Bulk import

A bulk import of events into the KB will need to be performed programmatically. Regardless of the format of the data that needs to be imported, it first needs to be converted to the JSON format we have outlined. i.e. if a customer has the data in a TSV format it needs to be converted to the JSON format we have outlined. Hopefully, the data will be programmatically converted to the JSON format. The REST calls can be performed programmatically as well, using Java, Scala, Python, etc.

Curl example

It is also possible to create a script that uses Curl as follows,

```
curl -u admin:pwd-goes-here -H "Content-Type: application/json" -d @example.json -X POST
   http://localhost:8081/kb/constitutionalevent
```

where `example.json` contains the JSON text.

bionano

Bulk export / search

There is a way to search the KB database for all events and get its basic information.

```
URL - protocol://server:port/kb/oncologyevents/search

URL - protocol://server:port/kb/constitutionalevents/search


// form parameters:

approvalStatus: APPROVED, TO_BE_REVIEWED, or ARCHIVED

eventKind: CNV, ZYGOSITY, SEQVAR, or SV

build: NCBI Build 37 (the value of related events being searched)
```

Curl example

```
curl --user admin:pwd-goes-here -X POST localhost:8081/kb/oncologyevents/search -F "approvalStatus=TO_BE_REVIEWED" -
   F "eventKind=CNV" -F "build=NCBI Build 37"
```

SEARCH RESPONSE

The response will be a json file containing basic info of all events that match the criteria.

Example:

```
1 [
2    {
3    "regionId": {
4            "uuid": "977bdc13-d0b1-4136-9d61-afef70c37b7d",
5            "revision": 1
6    },
7    "build": "NCBI Build 37",
8    "chr": "chr1",
9    "start": 2000000,
10   "end": 3000000,
11   "event": "loss",
12   "classification": "benign",
13   "label": "loss on chr1",
14   "exampleCases": [
15           {
16                   "name": "example sample",
17                   "note": "notes about sample"
18           }
19   ]
20 },
```

```
{
  "regionId": {
          "uuid": "83510572-3cdb-46b0-b45b-40812ed252fb",
          "revision": 1
  },
  "build": "NCBI Build 37",
  "chr": "chr1",
  "start": 3000000,
  "end": 4000000,
  "event": "loss",
  "classification": "benign",
  "label": "loss on chr1",
  "exampleCases": [
          {
                  "name": "example sample",
                  "note": "notes about sample"
          }
  ]
}
]
```

Technical Assistance

For technical assistance, contact Bionano Genomics Technical Support.

You can retrieve documentation on Bionano products, SDS's, certificates of analysis, frequently asked questions, and other related documents from the Support website or by request through e-mail and telephone.

TYPE	CONTACT
Email	support@bionano.com
Phone	Hours of Operation: Monday through Friday, 9:00 a.m. to 5:00 p.m., PST US: +1 (858) 888-7663
Website	www.bionano.com/support
Address	Bionano Genomics, Inc. 9540 Towne Centre Drive, Suite 100 San Diego, CA 92121