VNext Migration Guide

# Authentication

VNext supports multiple options for Authentication.

## JWT

User Passwords may be sent to the /api/2.0/authenticate endpoint. {login: “user\_name”, password: “password”}

The response will return {authenticated: true, jwt: “JWT TOKEN”} for a successful login.

The JWT token may then be used to authenticate API calls by setting it as the Authorization Header

Authorization: “Bearer ${jwt\_token}”

## Firebase

Firebase tokens obtained by the appropriate calls to the firebase authentication APIs can also be passed as Authorization Bearer Tokens similar to self signed JWT tokens above.

## Legacy V4

Legacy Basic Auth requests containing user id plus encrypted passwords “user:pass” base64 encoded and passed as an authorization header “Basic %{base64 encoded string}” may be used to authenticate calls.

## Legacy V5

Legacy Basic Auth requests containing user id plus session guid “user:session” base64 encoded and passed as an authorization header “Basic %{base64 encoded string}” may be used to authenticate calls.

# Search API

Search filters will generally work the same regardless of entity type although some filters are entity specific and will not be universally available. Search is driven by Manticore a fork of the Sphinx search engine. We use a combination of real time, and disk based main + delta indexes to provide search functionality.

## Search Enabled Entities

### Currently Available

* /api/2.0/query/crisis-posts
* /api/2.0/query/crisis-topics
* /api/2.0/query/users
* /api/2.0/query/comments

### Pending

* /api/2.0/query/moments
* /api/2.0/query/shout\_outs
* /api/2.0/query/images
* /api/2.0/query/documents
* /api/2.0/query/cms
* /api/2.0/query/schools
* /api/2.0/query/employers
* /api/2.0/query/vocations
* /api/2.0/query/majors

## Search Filters

### Constraints

#### Gender Constraint

Filter by Gender.

##### Available Options

* male
* female
* trans\_male
* trans\_female
* intersex
* gender\_fluid
* other

##### Example

|  |
| --- |
| /api/2.0/query/users?gender=male,trans\_male,intersex |

#### Origin Constraint

Filter by Origin

##### Available Options

* none
* jetzy
* crisis

##### Example

|  |
| --- |
| /api/2.0/query/users?origin=jetzy |

#### SCHOOL Constraint

Filter by School

##### Available Options

Schools are filtered by school identifiers.

|  |
| --- |
| /api/2.0/query/users?school=ref.school.1 |

#### Major Constraint

Filter by Education Major

##### Available Options

College Major Identifiers

|  |
| --- |
| /api/2.0/query/users?major=ref.college-major.1 |

#### Employer Constraint

Filter by Employer

##### Available Options

Employer Identifiers

|  |
| --- |
| /api/2.0/query/users?employer=ref.employer.1 |

#### Vocation Constraint

Filter by Job

##### Available Options

Vocation Identifiers

|  |
| --- |
| /api/2.0/query/users?vocation=ref.vocation.1 |

#### AGE Constraint

Filter by Job

##### Available Options

Age Min and Max.

|  |
| --- |
| /api/2.0/query/users?age-min=32&age-max=45 |
| [PUT] /api/2.0/query/users Request Body: {age:{min: 32, max: 34}} |

#### TOPIC Constraint

Filter by Crisis Topic

##### Available Options

Topic Identifiers

|  |
| --- |
| /api/2.0/query/users?topics=ref.crisis-topic.1,ref.crisis-topic.2 |

#### Interaction Constraint

Filter by Entity Comments, Likes, Dislikes, etc.

##### Available Options

You may specify interaction type and accepted ranges.   
Interactions| comments, likes, dislikes, hearts (pending).

|  |
| --- |
| /api/2.0/query/users?interactions=comments>=23,comments<=45,likes=5,dislikes=7,hearts<13 |
| [PUT] /api/2.0/query/users Body: (Support Pending current version looks for a map not a list.) {  interactions: [ %{comments: “>23”}, %{comments: “<45”}]  } |

#### GEO Constraint

Filter by GEO

##### Available Options

Filter by longitude/latitude or a location id. Current version does not provide max range support instead results may be ordered by geo. I’ll add a radius field in the near future.

|  |
| --- |
| /api/2.0/query/moments?lng=32.33&lat=23.34 |
| /api/2.0/query/moments?location=ref.location.1 |

#### Text Query

Perform text based query using Manticore/Sphinx MATCH syntax.

|  |
| --- |
| /api/2.0/query/moments?query=HelloWorld |
| [put] /api/2.0/query/moments  Body: {  query: “HelloWorld”  } |

### Order BY & Limit

The following ordery\_by params may be included (as query param or body param) to specify result ordering.

* GEO (Ascending)
* GEO DESC
* GEO ASC
* AGE (Ascending)
* AGE DESC
* AGE ASC
* TIME (Descending)
* TIME DESC
* TIME ASC
* WEIGHT (Query Match Weight)

Result Paging can additionally be specified as follows.

* page (0 to n)
* rpp (results per page

|  |
| --- |
| /api/2.0/query/crisis-posts?order\_by=GEO&rpp=50&page=4 |

# IMAGE UPLOAD API

The Image upload api is used to upload images to be associated with user profile images, posts, comments, etc. It returns image ids that then need to be included in update/create api calls for the other entity types. There are more actions available than listed below, such as ones for backend crop/resize operations, but these actions are still under development.

## Upload Image

The Upload API is used to upload images. Uploaded images are kept in a specific location and not promoted to saved images until included as part of an entity such as a crisis-post.

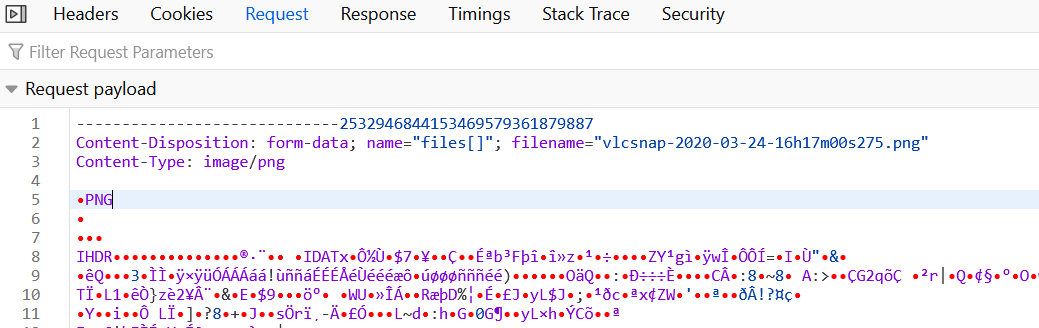
### Request

### [post] /documents/1.0/image-upload

[BODY]

Content-Type: multipart/form-data

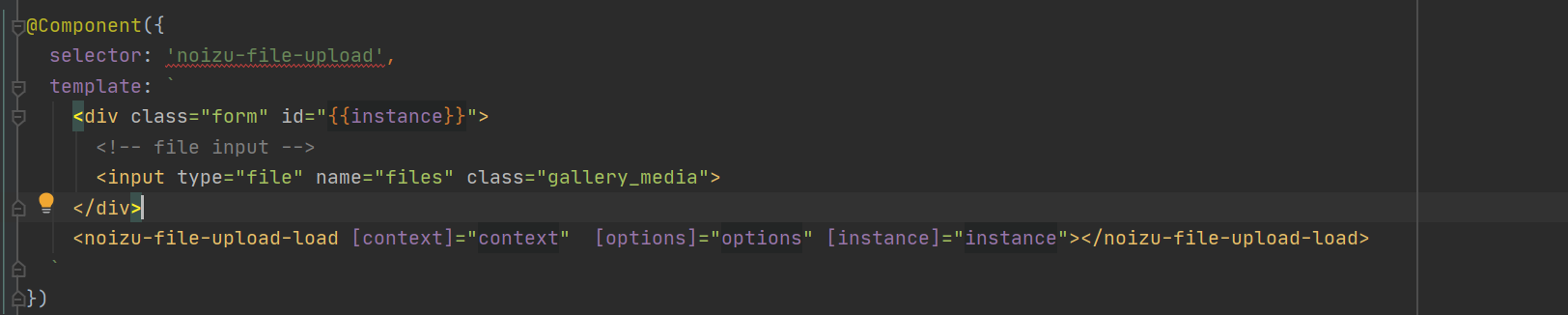
File with name=”files[]” and filename=”original file name”



For Example here is the upload method from crisis-help, where the s variable is the input field also shown below.



Input Element (where the name = “files[]” comes from.



### Response

The response breaks from the convention of some other api calls. For example is uses pascal case instead of snake case for hasWarnings, isSuccess. This is due to the need to match the interface expected by the third party Jquery FileUploader library used by the angular project.

|  |
| --- |
| {  **warnings**: [] // list of any warnings encountered.  **isSuccess**: bool, // Outcome of upload was successful.  **hasWarnings**: bool, // Warnings were raised.  **files**: [  **uploaded**: bool,  **type**: string, // “/image/png”  **titl**e: string, // Original File Name  **size2**: string, // 2.3 MB, 43.3 KB , etc. human readable sizes.  **size**: int, // size in bytes.  **replaced**: bool, // existing file overwritten.   **oldtitle**: string, // previous title/image file.  **oldname**: string, // previous name/image file.  **name**: string, // file name  **id**: string, // Uploaded file String Reference Id. E.g. “ref.image-upload.a948c9f6d37d47779a3578aba2cdc516”  **format**: string, // “image”  **file**: string, // path to uploaded file. E.g. “/mnt/upload/a948c9f6d37d47779a3578aba2cdc516/base”  **extension**: string, // e.g. “png”,  **dimensions**: {  **width**: integer,  **height**: integer,  },  **entity**: { // here is the actual upload file entity created by the backend, not in the Jquery Uploader format used above.  **\_\_struct\_\_**: “Elixir.Jetzy.ImageUploadEntity”,  **vsn**: float, // internal entity data structure version.  **source**: string, // e.g. “jetzy, none, crisis, etc. Where the image came from.  **size**: integer, // size in bytes.  **owner:** sref, // String Reference of entity who uploaded/owns this record. E.g. “ref.user.5”  **original\_file\_name**: string, // original file name  **name**: string, // Name for image, will be file name.  **modified\_on**: date, // iso8601 date time string.  **meta:** Internal Use,  **initial**: Internal Use,  **image\_type**: String | Null, “Profile Image, Post Image, etc. will be set to null here generally”  **identifier:** Guid String, // e.g. “a948c9f6d37d47779a3578aba2cdc516”  **hash:** MD5 String, // Hash of uploaded file (used to detect unexpected modifications, etc. example “F487506B2706BF125E1B44F8A4D2135E”  **file\_format**: String, // Format, “png”, “bmp”, etc.  **description**: null | string, // Description of image, will be null here.  **deleted\_on**: date | null, // ISO8601 Date Time String  **created\_on**: date, // ISO8601 Date Time String  **blur\_hash**: blur hash string | null, // will be null for now. Blur Hash is a library that allows colorful image approximations to be displayed while obtaining full images.  **base\_dimensions**: {  height: integer, width: integer,  }  **base**: String, // Base location for this image. If an image has different resolutions stored, translated versions, thumbs etc. they would be saved under this base dir.   // Example “/mnt/upload/a948c9f6d37d47779a3578aba2cdc516”  }  ]  } |

The most important piece here is the id (“ref.image-upload.a948c9f6d37d47779a3578aba2cdc516”) as this is what is needed to be added to other entities so when saved they can grab the upload file and convert it into a regular image entity.

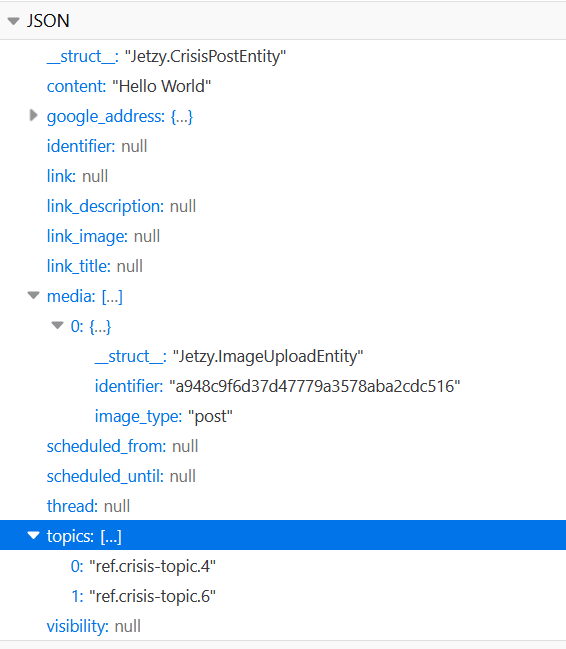
# Post API

Like other APIs we use standard Rest conventions [GET] to obtain an entity, [PUT] to update, [POST] to create, [DELETE] to delete. By convention we always use plural for the entity type. So [GET] /api/2.0/crisis-posts/ref.crisis-post.123 rather than [GET] /api/2.0/crisis-post/ref.crisis-post.123

## Create

[POST] /api/2.0/crisis-posts

**Body**: Json. Note the Crisis Post json sent to the server may differ slightly from the version returned. Note that Image Uploads are posted as a set of Entities.



**Response**

{   
 outcome: true/false,  
 entity: {.. the inserted entity}  
}

## UPDATE

[PUT] /api/2.0/crisis-posts/ref.crisis-post.5

Body: Same as above although media will contain Jetzy.ImageEntity records instead of Jetzy.ImageUploadEntity records for existing images.

## GET

<https://api-vnext.jetzyapp.com/api/2.0/crisis-posts/43>

**Response**

|  |
| --- |
| {  "vsn": 1.0,  "visibility": "public",  "universal\_identifier": 121,  "type": "media",  "topics": [],  "status": "active",  "sphinx\_index": null,  "scheduled\_until": null,  "scheduled\_from": null,  "radius": 1.0,  "owner": "ref.user.2",  "modified\_on": "2020-07-31T10:03:38.876221Z",  "moderation\_status": "none",  "meta": {},  "media": [  "ref.image.a948c9f6d37d47779a3578aba2cdc516"  ],  "location": "ref.location.9",  "interactions": "ref.entity-interactions.121",  "initial": null,  "identifier": 43,  "group": null,  "geo": {  "longitude": -91.25190169999999,  "latitude": 43.81377510000001  },  "deleted\_on": null,  "created\_on": "2020-07-31T10:03:38.876221Z",  "content\_flag": "none",  "content": "Hello World"  } |

**Note you can automatically expand embedded references Such as owner, images. Using ?expand-refs=list, or ?expand-all-refs=true**

**[GET]** [**https://api-vnext.jetzyapp.com/api/2.0/crisis-posts/43?expand-refs=user,image,entity-interactions**](https://api-vnext.jetzyapp.com/api/2.0/crisis-posts/43?expand-refs=user,image,entity-interactions)

* **note images Don’t currently correctly expand using this format.**
* **Some fields on expanded entries will be suppressed, such as user email address.**

|  |
| --- |
| {  "vsn": 1.0,  "visibility": "public",  "universal\_identifier": 121,  "type": "media",  "topics": [],  "status": "active",  "sphinx\_index": null,  "scheduled\_until": null,  "scheduled\_from": null,  "radius": 1.0,  "owner": {  "vsn": 1.0,  "verified": true,  "universal\_identifier": 22,  "sref": "ref.user.2",  "sphinx\_index": null,  "repo": "Elixir.Jetzy.UserEntity",  "ref\_kind": "Elixir.Jetzy.UserEntity",  "profile\_image": null,  "origin": "crisis",  "name": {  "last": "Brings",  "first": "Keith"  },  "modified\_on": "2020-06-19T15:41:52.058400Z",  "meta": {},  "interactions": {  "vsn": 1.0,  "synchronized\_on": "2020-06-19T15:41:52.058445Z",  "subject": "ref.user.2",  "sref": "ref.entity-interactions.22",  "repo": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "ref\_kind": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "reaction\_10": 0,  "reaction\_09": 0,  "reaction\_08": 0,  "reaction\_07": 0,  "reaction\_06": 0,  "reaction\_05": 0,  "reaction\_04": 0,  "reaction\_03": 0,  "reaction\_02": 0,  "reaction\_01": 0,  "modified\_on": "2020-06-19T15:41:52.058445Z",  "meta": {},  "initial": null,  "identifier": 22,  "comments": 0  },  "initial": null,  "info": "ref.user-info.2",  "identifier": 2,  "gender": "none",  "email": "keith.brings@noizu.com",  "deleted\_on": null,  "date\_of\_birth": null,  "created\_on": "2020-06-19T15:41:52.058400Z"  },  "modified\_on": "2020-07-31T10:03:38.876221Z",  "moderation\_status": "none",  "meta": {},  "media": [  "ref.image.a948c9f6d37d47779a3578aba2cdc516"  ],  "location": "ref.location.9",  "interactions": {  "vsn": 1.0,  "synchronized\_on": "2020-07-31T10:03:38.901278Z",  "subject": "ref.crisis-post.43",  "sref": "ref.entity-interactions.121",  "repo": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "ref\_kind": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "reaction\_10": 0,  "reaction\_09": 0,  "reaction\_08": 0,  "reaction\_07": 0,  "reaction\_06": 0,  "reaction\_05": 0,  "reaction\_04": 0,  "reaction\_03": 0,  "reaction\_02": 0,  "reaction\_01": 0,  "modified\_on": "2020-07-31T10:03:38.901278Z",  "meta": {},  "initial": null,  "identifier": 121,  "comments": 0  },  "initial": null,  "identifier": 43,  "group": null,  "geo": {  "longitude": -91.25190169999999,  "latitude": 43.81377510000001  },  "deleted\_on": null,  "created\_on": "2020-07-31T10:03:38.876221Z",  "content\_flag": "none",  "content": "Hello World"  } |

**[GET] https://api-vnext.jetzyapp.com/api/2.0/crisis-posts/43?expand-all-refs=true**

|  |
| --- |
| {  "vsn": 1.0,  "visibility": "public",  "universal\_identifier": 121,  "type": "media",  "topics": [],  "status": "active",  "sphinx\_index": null,  "scheduled\_until": null,  "scheduled\_from": null,  "radius": 1.0,  "owner": {  "vsn": 1.0,  "verified": true,  "universal\_identifier": 22,  "sref": "ref.user.2",  "sphinx\_index": null,  "repo": "Elixir.Jetzy.UserEntity",  "ref\_kind": "Elixir.Jetzy.UserEntity",  "profile\_image": null,  "origin": "crisis",  "name": {  "last": "Brings",  "first": "Keith"  },  "modified\_on": "2020-06-19T15:41:52.058400Z",  "meta": {},  "interactions": {  "vsn": 1.0,  "synchronized\_on": "2020-06-19T15:41:52.058445Z",  "subject": "ref.user.2",  "sref": "ref.entity-interactions.22",  "repo": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "ref\_kind": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "reaction\_10": 0,  "reaction\_09": 0,  "reaction\_08": 0,  "reaction\_07": 0,  "reaction\_06": 0,  "reaction\_05": 0,  "reaction\_04": 0,  "reaction\_03": 0,  "reaction\_02": 0,  "reaction\_01": 0,  "modified\_on": "2020-06-19T15:41:52.058445Z",  "meta": {},  "initial": null,  "identifier": 22,  "comments": 0  },  "initial": null,  "info": {  "vsn": 1.0,  "vocation": null,  "sref": "ref.user-info.2",  "school": null,  "repo": "Elixir.Jetzy.User.InfoEntity",  "ref\_kind": "Elixir.Jetzy.User.InfoEntity",  "panic\_cms": null,  "modified\_on": null,  "meta": {},  "major": null,  "initial": null,  "identifier": 2,  "employer": null,  "bio\_cms": null,  "about\_cms": null  },  "identifier": 2,  "gender": "none",  "email": "keith.brings@noizu.com",  "deleted\_on": null,  "date\_of\_birth": null,  "created\_on": "2020-06-19T15:41:52.058400Z"  },  "modified\_on": "2020-07-31T10:03:38.876221Z",  "moderation\_status": "none",  "meta": {},  "media": [  {  "\_\_struct\_\_": "Elixir.Jetzy.ImageEntity",  "vsn": 1.0,  "universal\_identifier": 122,  "sref": "ref.image.a948c9f6d37d47779a3578aba2cdc516",  "sphinx\_index": null,  "source": "none",  "repo": "Elixir.Jetzy.ImageEntity",  "ref\_kind": "Elixir.Jetzy.ImageEntity",  "name": "vlcsnap-2020-03-24-16h17m00s275.png",  "mysql\_identifier": 11,  "modified\_on": "2020-07-31T09:26:27.410291Z",  "moderation\_status": null,  "meta": {},  "localized": null,  "locale": null,  "interactions": null,  "initial": null,  "image\_type": "post",  "identifier": "a948c9f6d37d47779a3578aba2cdc516",  "hash": "F487506B2706BF125E1B44F8A4D2135E",  "file\_format": "png",  "external": false,  "description": null,  "deleted\_on": null,  "created\_on": "2020-07-31T09:26:27.410276Z",  "content\_flag": null,  "blur\_hash": null,  "base\_dimensions": {  "width": 1280,  "height": 534  },  "base": "/mnt/images/a948c9f6d37d47779a3578aba2cdc516"  }  ],  "location": {  "vsn": 1.0,  "universal\_identifier": 120,  "status": "active",  "sref": "ref.location.9",  "sphinx\_index": null,  "source": "google",  "repo": "Elixir.Jetzy.LocationEntity",  "ref\_kind": "Elixir.Jetzy.LocationEntity",  "radius\_kilometer": null,  "place\_url": "https://maps.google.com/?q=La+Crosse,+WI,+USA&ftid=0x87f954df6732b395:0x7fa98815193722b0",  "place\_id": "ChIJlbMyZ99U-YcRsCI3GRWIqX8",  "place\_icon": "https://maps.gstatic.com/mapfiles/place\_api/icons/geocode-71.png",  "place\_address": "<span class=\"locality\">La Crosse</span>, <span class=\"region\">WI</span>, <span class=\"country-name\">USA</span>",  "name": "La Crosse",  "modified\_on": "2020-07-31T10:03:38.876269Z",  "meta": {},  "location\_type": "none",  "initial": null,  "identifier": 9,  "geo": {  "longitude": -91.25190169999999,  "latitude": 43.81377510000001  },  "description\_cms": null,  "deleted\_on": null,  "created\_on": "2020-07-31T10:03:38.876269Z"  },  "interactions": {  "vsn": 1.0,  "synchronized\_on": "2020-07-31T10:03:38.901278Z",  "subject": {  "vsn": 1.0,  "visibility": "public",  "universal\_identifier": 121,  "type": "media",  "topics": [],  "status": "active",  "sref": "ref.crisis-post.43",  "sphinx\_index": null,  "scheduled\_until": null,  "scheduled\_from": null,  "repo": "Elixir.Jetzy.CrisisPostEntity",  "ref\_kind": "Elixir.Jetzy.CrisisPostEntity",  "radius": 1.0,  "owner": {  "vsn": 1.0,  "verified": true,  "universal\_identifier": 22,  "sref": "ref.user.2",  "sphinx\_index": null,  "repo": "Elixir.Jetzy.UserEntity",  "ref\_kind": "Elixir.Jetzy.UserEntity",  "profile\_image": null,  "origin": "crisis",  "name": {  "last": "Brings",  "first": "Keith"  },  "modified\_on": "2020-06-19T15:41:52.058400Z",  "meta": {},  "interactions": {  "vsn": 1.0,  "synchronized\_on": "2020-06-19T15:41:52.058445Z",  "subject": "ref.user.2",  "sref": "ref.entity-interactions.22",  "repo": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "ref\_kind": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "reaction\_10": 0,  "reaction\_09": 0,  "reaction\_08": 0,  "reaction\_07": 0,  "reaction\_06": 0,  "reaction\_05": 0,  "reaction\_04": 0,  "reaction\_03": 0,  "reaction\_02": 0,  "reaction\_01": 0,  "modified\_on": "2020-06-19T15:41:52.058445Z",  "meta": {},  "initial": null,  "identifier": 22,  "comments": 0  },  "initial": null,  "info": {  "vsn": 1.0,  "vocation": null,  "sref": "ref.user-info.2",  "school": null,  "repo": "Elixir.Jetzy.User.InfoEntity",  "ref\_kind": "Elixir.Jetzy.User.InfoEntity",  "panic\_cms": null,  "modified\_on": null,  "meta": {},  "major": null,  "initial": null,  "identifier": 2,  "employer": null,  "bio\_cms": null,  "about\_cms": null  },  "identifier": 2,  "gender": "none",  "email": "keith.brings@noizu.com",  "deleted\_on": null,  "date\_of\_birth": null,  "created\_on": "2020-06-19T15:41:52.058400Z"  },  "modified\_on": "2020-07-31T10:03:38.876221Z",  "moderation\_status": "none",  "meta": {},  "media": [  {  "\_\_struct\_\_": "Elixir.Jetzy.ImageEntity",  "vsn": 1.0,  "universal\_identifier": 122,  "sref": "ref.image.a948c9f6d37d47779a3578aba2cdc516",  "sphinx\_index": null,  "source": "none",  "repo": "Elixir.Jetzy.ImageEntity",  "ref\_kind": "Elixir.Jetzy.ImageEntity",  "name": "vlcsnap-2020-03-24-16h17m00s275.png",  "mysql\_identifier": 11,  "modified\_on": "2020-07-31T09:26:27.410291Z",  "moderation\_status": null,  "meta": {},  "localized": null,  "locale": null,  "interactions": null,  "initial": null,  "image\_type": "post",  "identifier": "a948c9f6d37d47779a3578aba2cdc516",  "hash": "F487506B2706BF125E1B44F8A4D2135E",  "file\_format": "png",  "external": false,  "description": null,  "deleted\_on": null,  "created\_on": "2020-07-31T09:26:27.410276Z",  "content\_flag": null,  "blur\_hash": null,  "base\_dimensions": {  "width": 1280,  "height": 534  },  "base": "/mnt/images/a948c9f6d37d47779a3578aba2cdc516"  }  ],  "location": {  "vsn": 1.0,  "universal\_identifier": 120,  "status": "active",  "sref": "ref.location.9",  "sphinx\_index": null,  "source": "google",  "repo": "Elixir.Jetzy.LocationEntity",  "ref\_kind": "Elixir.Jetzy.LocationEntity",  "radius\_kilometer": null,  "place\_url": "https://maps.google.com/?q=La+Crosse,+WI,+USA&ftid=0x87f954df6732b395:0x7fa98815193722b0",  "place\_id": "ChIJlbMyZ99U-YcRsCI3GRWIqX8",  "place\_icon": "https://maps.gstatic.com/mapfiles/place\_api/icons/geocode-71.png",  "place\_address": "<span class=\"locality\">La Crosse</span>, <span class=\"region\">WI</span>, <span class=\"country-name\">USA</span>",  "name": "La Crosse",  "modified\_on": "2020-07-31T10:03:38.876269Z",  "meta": {},  "location\_type": "none",  "initial": null,  "identifier": 9,  "geo": {  "longitude": -91.25190169999999,  "latitude": 43.81377510000001  },  "description\_cms": null,  "deleted\_on": null,  "created\_on": "2020-07-31T10:03:38.876269Z"  },  "interactions": "ref.entity-interactions.121",  "initial": null,  "identifier": 43,  "group": null,  "geo": {  "longitude": -91.25190169999999,  "latitude": 43.81377510000001  },  "deleted\_on": null,  "created\_on": "2020-07-31T10:03:38.876221Z",  "content\_flag": "none",  "content": "Hello World"  },  "sref": "ref.entity-interactions.121",  "repo": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "ref\_kind": "Elixir.Jetzy.Entity.InteractionsCacheEntity",  "reaction\_10": 0,  "reaction\_09": 0,  "reaction\_08": 0,  "reaction\_07": 0,  "reaction\_06": 0,  "reaction\_05": 0,  "reaction\_04": 0,  "reaction\_03": 0,  "reaction\_02": 0,  "reaction\_01": 0,  "modified\_on": "2020-07-31T10:03:38.901278Z",  "meta": {},  "initial": null,  "identifier": 121,  "comments": 0  },  "initial": null,  "identifier": 43,  "group": null,  "geo": {  "longitude": -91.25190169999999,  "latitude": 43.81377510000001  },  "deleted\_on": null,  "created\_on": "2020-07-31T10:03:38.876221Z",  "content\_flag": "none",  "content": "Hello World"  } |

## DELETE

[Delete] /api/2.0/crisis-posts/ref.crisis-post.5

# Moment API

## Create

## UPDATE

## GET

## UPDATE

# ShoutOut API

## Create

## UPDATE

## GET

## UPDATE

# Comment API

Comments can be appended to any entity type (provided you have the correct reference string.)

## Create top level comment

[POST] /api/2.0/entity/:subject\_ref/comments

**Example**

[POST] <https://api-vnext.jetzyapp.com/api/2.0/entity/ref.crisis-post.1/comments?expand-refs=user,entity-interactions>  
  
Body:

|  |
| --- |
| {  comment: “Hello World”,  } |

## Create NESTED comment

[POST] /api/2.0/entity/:subject\_ref/comments/:parent\_comment\_ref/replies

Example

[POST] /api/2.0/entity/ref.crisis-post.1/ref.comment.543/replies

|  |
| --- |
| {  comment: “Hello World 2”,  } |

## UPDATE

## GET

## UPDATE

# User Interaction API

API pending (I’ll post later this week), but it will look something like

[PUT] /api/2.0/entities/:ref/interaction/:interaction - Add an interaction to entity

[DELETE] /api/2.0/entities/:ref/interaction/:interaction – Remove an interaction from an entity.

Where :interaction may be like, dislike, heart, laugh, happy, angry, etc. Only one interaction will exist at a time, so you won’t be able to like and dislike a entity, etc.

## Create

## UPDATE

## GET

## UPDATE

# CMS API

API Details Pending. The Noizu KitchenSink CMS V2 library allows for tracking version trees of records (Version 1.5.3.2) plus multiple per version revisions.

# Data StructureS

Here are some of the common data structures you’ll be working with.

## Moment

This data structures may be modified slightly as I finish importing Moment records into vnext.

|  |
| --- |
| {  **identifier**: integer,  **universal\_identifier**: integer,  **owner**: ref string or entity,  **moment\_type**: (pending)  **post\_type**: text | url | image | video,  **geo**: {longitude: float, latitude: float} | null,  **cms\_entry**: ref string or entity (see CMS section)  **status**: active |pending | …,  **visibility**: visible | private,  **moderation\_status**: none |flagged |under\_review | accepted |rejected,  **content\_flag**: none, user\_Flagged, auto\_flagged, content, fact\_check,  **interactions**: See Interactions Section,  **sphinx\_index**: null | real\_time | delta | main,  **initial**: internal use  **meta**: internal use,  **created\_on**: ISO8601 date time string,  **modified\_on**: ISO8601 date time string,  **deleted\_on**: ISO8601 date time string | null,  **vsn**: internal use (version of record format).  } |

## ShoutOut

This data structures may be modified slightly as I finish importing ShoutOut records into vnext.

|  |
| --- |
| {  **identifier**: integer,  **universal\_identifier**: integer,  **cms\_entry**: ref string or entity (see CMS Section),  **moderation\_status**: none |flagged |under\_review | accepted |rejected,  **content\_flag**: none, user\_Flagged, auto\_flagged, content, fact\_check,  **interactions**: See Interactions Section,  **sphinx\_index**: null | real\_time | delta | main,  **initial**: internal use  **meta**: internal use,  **created\_on**: ISO8601 date time string,  **modified\_on**: ISO8601 date time string,  **deleted\_on**: ISO8601 date time string | null,  **vsn**: internal use (version of record format).  } |

## Crisis POST

|  |
| --- |
| {  **identifier**: integer,  **universal**\_identifier: integer | nil,  **type**: :text | :image | :video | atom,  **content**: String.t,  **media**: list,  **owner**: tuple | Jetzy.UserEntity.t,  **topics**: [tuple | Jetzy.CrisisTopicEntity.t],  **status**: :approved | :pending | :deleted | atom,  **moderation**\_status: :none | :flagged | :under\_review | :accepted | :rejected | atom,  **content\_flag**: :none | :user\_flagged | :auto\_flagged | :content | :fact\_check | atom,  **location**: tuple,  **radius**: float | nil,  **geo**: Geo.Point.t | nil,  **visibility**: :public | :private | :group | :public\_scheduled | :group\_scheduled | atom, # :  **group**: tuple,  **scheduled\_from**: nil,  **scheduled\_until**: nil,  **interactions**: tuple | any,  **sphinx\_index**: atom,  **created\_on**: DateTime.t,  **modified\_on**: DateTime.t,  **deleted\_on**: DateTime.t,  **initial**: any,  **meta**: Map.t,  **vsn**: float  } |

## User

|  |
| --- |
| {  **identifier**: integer,  **universal\_identifier**: integer,  **name**: Map.t,  **email**: String.t,  **gender**: atom,  **date\_of\_birth**: DateTime.t,  **profile\_image**: tuple | nil | Jetzy.ImageEntity.t,  **verified**: boolean,  **origin**: atom,  **info**: :not\_loaded | Jetzy.User.InfoEntity.t | tuple,  **interactions**: any,  **created\_on**: DateTime.t,  **modified\_on**: DateTime.t,  **deleted\_on**: DateTime.t,  **sphinx\_index**: atom,  **initial**: any, # original record on load for detecting changes on update/save.  **meta**: Map.t,  **vsn**: float  } |

## Comment

|  |
| --- |
| {  identifier: integer,  universal\_identifier: integer,  comment\_type: atom,  subject: tuple,  subject\_uid: integer,  parent: tuple,  owner: tuple,  comment: String.t,  location: any,  geo: any,  status: any,  moderation\_status: any,  content\_flag: any,  path: nil | Jetzy.PathEncoding.t,  children: list,  interactions: tuple | any,  sphinx\_index: atom,  created\_on: DateTime.t,  modified\_on: DateTime.t,  deleted\_on: DateTime.t,  initial: Map.t | nil,  meta: Map.t,  vsn: float  } |

## Interaction Snippet

|  |
| --- |
| {  identifier: nil,  subject: nil,  comments: 0,  reaction\_01: 0,  reaction\_02: 0,  reaction\_03: 0,  reaction\_04: 0,  reaction\_05: 0,  reaction**\_**06: 0,  reaction\_07: 0,  reaction\_08: 0,  reaction\_09: 0,  reaction\_10: 0,  synchronized\_on: nil,  modified\_on: nil,  initial: nil,  meta: %{},  vsn: @vsn  } |

## CMS

### CMS.MomentEntity

|  |
| --- |
| {  identifier: integer,  universal\_identifier: integer,  title: Noizu.MarkdownField.t | nil,  body: Noizu.MarkdownField.t | nil,  attributes: Map.t,  article\_info: Noizu.Cms.V2.Article.Info.t | nil,  moderation\_status: atom,  content\_flag: atom,  sphinx\_index: atom,  created\_on: DateTime.t,  modified\_on: DateTime.t,  deleted\_on: DateTime.t,  initial: any,  meta: Map.t,  vsn: float  } |

### CMS.CrisisPostEntity

|  |
| --- |
| {  identifier: integer,  universal\_identifier: integer,  title: Noizu.MarkdownField.t | nil,  body: Noizu.MarkdownField.t | nil,  attributes: Map.t,  article\_info: Noizu.Cms.V2.Article.Info.t | nil,  moderation\_status: atom,  content\_flag: atom,  sphinx\_index: atom,  created\_on: DateTime.t,  modified\_on: DateTime.t,  deleted\_on: DateTime.t,  initial: any,  meta: Map.t,  vsn: float  } |

### CMS.DescriptionEntity

|  |
| --- |
| {  identifier: integer,  universal\_identifier: integer,  title: Noizu.MarkdownField.t | nil,  body: Noizu.MarkdownField.t | nil,  attributes: Map.t,  article\_info: Noizu.Cms.V2.Article.Info.t | nil,  moderation\_status: atom,  content\_flag: atom,  sphinx\_index: atom,  created\_on: DateTime.t,  modified\_on: DateTime.t,  deleted\_on: DateTime.t,  initial: any,  meta: Map.t,  vsn: float  } |

### CMS.ShoutOutEntity

|  |
| --- |
| {  identifier: integer,  universal\_identifier: integer,  title: Noizu.MarkdownField.t | nil,  body: Noizu.MarkdownField.t | nil,  attributes: Map.t,  article\_info: Noizu.Cms.V2.Article.Info.t | nil,  moderation\_status: atom,  content\_flag: atom,  sphinx\_index: atom,  created\_on: DateTime.t,  modified\_on: DateTime.t,  deleted\_on: DateTime.t,  initial: any,  meta: Map.t,  vsn: float  } |