

## API v1 Draft

This is a draft specification for version 1 of the ubcnw-server REST API.

Many things are likely to change before finalization.

## Using the API

### Encoding

All body requests and responses are encoded in JSON. The fields in the specification for each model object directly correspond to its JSON fields in requests and responses, unless otherwise indicated.

### Versioning

All API calls must be preceded by a version identifier:

`http://host:port/api/v1/`

Alternatively, you may also request the latest API:

`http://host:port/api/latest/`

All endpoints documented are expressed relative to the above URL.

### Parameters

URL parameters are expressed with `:this` notation.

Optional parameters are in parentheses.

### Authentication

TBD

## Data Model

### Blips

A ‘blip’ is a general class of event that is propagated to a group’s users.

## Specification

Field	Type	Description	Required
id	Long	A unique index generated by the DB.	true
group	Group	The group of this blip.	true
title	String	A user-readable event title.	true
summary	String	A user-readable event summary.	false
link	String	A link to the event's page.	false
time	Date	The time of the event.	false
address	String	User-readable address of location.	false
lat	Double	Latitude of event location.	false
lng	Double	Longitude of event location.	false

**Example Data**    TODO

## Users

**TODO: (probably best to wait on auth (CWL) for this one)**

A ‘user’ is a data model representing a client, with whom blips are paired off based on the user’s intersts + group affiliations

## Specification

Field	Type	Description	Re
puid	Long	A unique index generated by the univerity	tru
studentNumber	Long	A university-generated student number of the user	fal
affiliation	String	The affiliation of the user to the university (student, employee or guest)	tru
firstName	String	The first name of the user	tru
lastName	String	The last name of the user	tru
interests	ListInterest	A list of the user’s interests	tru
groups	ListGroups	A list of groups the user is subscribed to	tru

TODO:

## Groups

A ‘group’ is a classification composed of users who wish to receive a certain type of notification.

TODO: groups admin API

### Specification

Field	Type	Description	Required
id	Long	Group identifier	true
name	String	A user-readable group name	true
interests	List[Long]	A list of interest IDs the group relates to.	true
concrete	Boolean	True if this is a concrete group	true

**Example Data**    TODO

### Examples

A concrete group would be the UBC math club.

An abstract group would be a “bus notifications” group.

## Interest

An interest is a classification indicated by individual users. Interests are paired with groups to identify potential Blips of interest.

### Specification

Field	Type	Description	Required
id	Long	Interest identifier	true
name	String	User-readable name of this interest	true

**Example Data**

TODO

## Endpoints

Note that all HTTP body responses are encoded in JSON.

### Blips

Path	Method	Notes
/blip	GET	Receives a list of the user's current blips.
/blip	POST	Creates a new blip with the JSON data in the request's body.
/blip/:id	GET	Retrieves the blip with the given id.
/blip/:id	PUT	Updates the blip's fields with the JSON data in the request's body. TODO: authentic
/blip/:id	DELETE	Deletes the blip with the given id.

### Examples

TODO

### Groups

Path	Method	Notes	Request
/group	GET	Retrieves a list of the user's groups.	N/A
/group	POST	Creates a new group.	Group
/group/:id	GET	Retrieves the group with the given id.	N/A
/group/:id	PUT	Updates a group.	Group
/group/:id	DELETE	Deletes a group.	N/A
/group/:id/blips	GET	Retrieves a list of blips belonging to the group, recent first.	count: Int, after
/group/search	GET	Finds groups with the given query string.	query: String

### Examples

TODO

## Blip Stream

Endpoint:

/blip/stream

ubcnw-server can communicate real-time events to clients connected using unidirectional WebSockets.

JSON blobs representing ‘updates’ are sent over the WebSocket stream.

## Update Specification

This specification corresponds to JSON fields in the messages sent via WebSocket.

Field	Type	Description
type	enum { add, update, delete }	Denotes the type of update.
id	Long	The blip being added, updated, or deleted.