

**JUXT**

**Func MK Meetup**  
**Vent**

# Table of Contents

Start the project .....	1
Application Structure .....	1
Vent .....	1
User .....	2
Challenges .....	3
Fetch database vents .....	3
Add author to vents .....	3
Implement favorites list .....	3
Implement toggle favorite button .....	4
Implement Add Vent .....	4
Implement Following page .....	4
Implement Followers page .....	4
Implement (un)Follow Button .....	4

Vent is a Microblogging (twitter-like) application. The purpose of this tutorial is to learn the basics of Clojure, with some guidance from a tutor.

## Start the project

1. Clone edge and go into the tutorial

```
$ git clone --single-branch --branch tutorial https://github.com/juxt/edge.git
$ cd edge/tutorial.vent
```

2. Start a REPL

```
$ ../bin/rebel --cljs -A:dev:build:dev/build
```

3. Start the server

```
dev=> (go)
```

There is a front-end you can browse at <http://localhost:3000>.

## Application Structure

In `src/tutorial/vent/lib.clj` there are a number of functions which haven't yet been implemented. The majority of challenges require:

1. Fetching information from a "db" (A file on disk)
2. Manipulating that data
3. Writing new data to the db

There are 3 operations for the database:

`(db/reset)`

This will reset the database to a known good state

`(db/store new-db)`

Save a new version of the database

`(db/read)`

Fetch the current version of the database

These are available in your REPL.

## Vent

Vents are stored in the database as a sequence. Each Vent has these keys:

**:id**

A globally unique id for this message, a number works.

**:text**

The text of the actual vent

**:username**

The username of the author, a string.

**:favorite?**

(Optional) Either **true** or **false**, dictates where the tweet is a favorite.

**:author**

(Optional) The data of the User (see below) who created it. This should be added after reading from the database, it should not be stored in the database.

An example Vent in the database

```
{:id 2
 :text "Lord but I dislike poetry. How can anyone remember words that aren't put to music?"
 :username "ar1_the_bard"
 :favorite? true}
```

An example Vent with an author attached

```
{:id 2
 :text "Lord but I dislike poetry. How can anyone remember words that aren't put to music?"
 :author {:name "Arliden"
          :following? true}
 :username "ar1_the_bard"
 :favorite? true}
```

How Vents look in the database

```
{:vents [{:id 1 :text "Text" :username "foo"} ...]}
```

## User

Users are stored in the database as a map. This is like a dictionary or hash map in other languages. The keys are usernames, and the values are Users.

Users look like so:

**:name**

A full name

**:follows**

A list of usernames this user follows

**:following?**

(Optional) **true** or **false**, defaults to false. Whether or not the user is being followed

## An example user

```
{:name "John Smith"
 :follows ["janesmith" "arl_the_bard"]
 :following? false}
```

## How users are stored in the database

```
{:users {"jsmith" {:name "John Smith"
                  :follows ["janesmith" "arl_the_bard"]
                  :following? true}
         "arl_the_bard" {:name "Arliden"
                        :follows ["janesmith"]}}}
```

# Challenges

These can all be achieved by modifying files in `src/tutorial/vent/lib.clj`.

## Fetch database vents

Currently, the vents from `all` are hard-coded. It would be great if `all` would return all the vents from the database.

**TIP** Use `(db/read)` and `get` [<https://clojuredocs.org/clojure.core/get>] the `:vents` key out of it.

## Add author to vents

Depends on [Fetch database vents](#).

Now that vents are being fetched from the database, the user details are missing! Fetch the user information from the database based on the `:username` key in the vent, and add it to the vents as the `:author` key.

**TIP** Use `map` [<https://clojuredocs.org/clojure.core/map>] to modify each of the vents with `assoc` [<https://clojuredocs.org/clojure.core/assoc>], and `get-in` [<https://clojuredocs.org/clojure.core/get-in>] to get the corresponding user from the database.

## Implement favorites list

Take the list of vents in the database, and filter them to show only the favorites on the favorites page.

**TIP** Use `filter` [<https://clojuredocs.org/clojure.core/filter>] and `anonymous functions` [[https://clojure.org/guides/learn/functions#\\_anonymous\\_functions](https://clojure.org/guides/learn/functions#_anonymous_functions)] to filter the list of vents.

## Implement toggle favorite button

You won't be able to see this working until you've done [Implement favorites list](#).

Add the ability for the user to make a vent a favorite.

**TIP** Use `map` [<https://clojuredocs.org/clojure.core/map>] and check using `=` [<https://clojuredocs.org/clojure.core/=>] if the current vent is the one you're supposed to `update` [<https://clojuredocs.org/clojure.core/update>]. You might want to <https://clojuredocs.org/clojure.core/not> the current value.

## Implement Add Vent

Add the ability to create a new vent.

**TIP** `update` [<https://clojuredocs.org/clojure.core/update>] the `:vents` in the database with `conj` [<https://clojuredocs.org/clojure.core/conj>]. Use the `generate-id` function to create an id for the vent.

## Implement Following page

Add the ability to see who you follow.

**TIP** `select-keys` [<https://clojuredocs.org/clojure.core/select-keys>] over users in your `:follows` key.

## Implement Followers page

Add the ability to see who your followers are.

**TIP** `select-keys` [<https://clojuredocs.org/clojure.core/select-keys>] on users, based on a `keep` [<https://clojuredocs.org/clojure.core/keep>] on the `:users` which have `username` in their `:follows`.

## Implement (un)Follow Button

You will need to complete [Implement Followers page](#) to see this working.

Make it so that Follow/Unfollow buttons toggle the current state.

**TIP** `update-in` [<https://clojuredocs.org/clojure.core/update-in>] the database and `conj` [<https://clojuredocs.org/clojure.core/conj>] in the new `username`.