

## Lisp in Summer Projects Submission

<b>Submission Date</b>	2013-10-03 05:00:39
<b>Full Name</b>	Bastien Guerry
<b>Zip/postal code</b>	75018
<b>Country</b>	France
<b>Project Name</b>	WLM MAP
<b>Type of software</b>	web app
<b>General category</b>	web tool
<b>LISP dialect</b>	Clojure
<b>GitHub URL</b>	<a href="https://github.com/bzg/wlmmmap">https://github.com/bzg/wlmmmap</a>
<b>Did you start this project?</b>	Yes, all the code is written by me
<b>Project Description</b>	I want to describe my project in this form.
<b>Purpose</b>	<p>Display a database of 1.2M cultural heritage monuments on a map.</p> <p>The web application is available here: <a href="http://www.panoramap.org">http://www.panoramap.org</a></p>
<b>Function</b>	The purpose of the map is to display cultural heritage monuments as gathered through the years during the Wiki Loves Monuments contests, and to encourage people to upload new pictures for monuments that don't have any.
<b>Motivation</b>	The Wiki Loves Monuments needed a map, I wanted to help.
<b>Audience</b>	Wikipedians, Wikimedia commonist, people that are passionate about cultural heritage, and the general public who may discover and share monuments they did not know about.
<b>Methodology</b>	The MySQL database of monuments is maintained by a team of wikipedians on the Wikimedia toolserver---see <a href="https://commons.wikimedia.org/wiki/Commons:Monuments_database">https://commons.wikimedia.org/wiki/Commons:Monuments_database</a>

for details.

Some countries have a lot of monuments: e.g. France has more than 50000. Retrieving +50000 entries takes too long when using the database API.

So the first step was to build a backend for syncing a local Redis database with data from the toolserver, as the Redis instance gives faster results.

The second step was to display entries from the local Redis database on a per-country basis.

The app uses the mapbox.js javascript library and the MarkerCluster method to display monuments as clustered markers. Displaying +50000 entries at once is too long from a user point of view: as a trade-off, the application uses the Clojure core.async library to display the monuments progressively, allowing the user to explore monuments while they are added up to the map.

The third step was to allow displaying entries from the toolserver directly when the geographical area is limited enough.

The last step was to localize the application (mainly the list of countries) and to implement permalinks for a monument.

## Conclusion

The map is functional and displays all monuments from the whole world, with the list of countries available in many languages.

There are two important limitations : (1) the time it takes to load monuments for countries that have many ; (2) the fact that monuments are not automatically displayed when the user moves the map.

(1) could be solved by using Clojure core.async and mapbox.js in a more clever way.

(2) could be solved by automatically deleting previously shown monuments and displaying monuments for the newly visible area.

## Build Instructions

This assumes you have an heroku account.

```
~$ git clone https://github.com/bzg/wlmmmap
```

Create a heroku instance for the app.

Add an openredis addon and setup the OPENREDIS\_URL environment variable.

Setup the "backendpwd" environment variable on the heroku app.

Push the app on the heroku instance and login through <http://yourapp.herokuapp.com/login>

Go to <http://yourapp.herokuapp.com/backend> and sync monuments from the list of country/languages.

## Test Instructions

There are no tests implemented for this app.

## Execution Instructions

Go to <http://yourapp.herokuapp.com> and browse the map.

**Describe any bugs or caveats**

Syncing takes long. Sometimes you seem to be disconnected while you are not, you just need to refresh the page.

**Screen shots**

□ [wlm\\_screenshot\\_1-960x539.png](#)

□ [wlm\\_screenshot\\_2.png](#)

**Official**

I have read rules and have abided by them.  
I am 18 years of age or older.  
I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran, Myanmar (Burma), North Korea, Sudan, or Syria.