

CT30A9301 - Code Camp on Platform Based Application Development

Open Data and Green IT CodeCamp - Spring 2015

LocalEAT

Anar Bazarhanova - 0446968

Julien Dhallenne - 0446926

Khan Mohammad Habibullah - 0446890

Marie Leslie Melanie Pittumbur - 0446942

LocalEAT

Introduction:

The basic idea of food sovereignty is that the aspirations and needs of those who produce, distribute and consume food should be at the heart of our food systems, rather than the demands of transnational corporations. It prioritizes local food production, based on agro-ecology and family farming, and local markets.

Why is eating locally Important? :

- Locally grown food tastes, looks and is better for health.
- Eating locally grown food reduces global warming and preserves genetic diversity of local flora.
- Activities derived from producing local food generates revenue, builds community, keeps taxes down, and is an investment for the future.

What is LocalEAT?

LocalEAT is a hybrid mobile application that will help users to find locally produced food in a certain area in Finland. By using this application, producers will be able to advertise their food and buyers will be able to find out food in their nearest local market as well. This application is used to encourage the sustainability of local and small producers and creates awareness among citizens about buying local products reaping on the way the benefits of eating fresh food.

The application is based on an open REST API through which everyone can participate and contribute to by adding information about where different types of fresh food can be found from markets of their locality.

The goals of the LocalEAT application is summarized below:

- Introduce locally produced food and make it more familiar.
- Help producers and market sellers to advertise their food.
- Consumers can find food that they want to buy very easily in the nearest local market.
- localEAT application will show the market locations and directions to the users to find market.

The use of Open Data in localEAT application

By providing timely and useful information to users through an open-based data application, it is possible to incite and help people to eat more healthily and sustainably. Also a public platform for open-data can play an important role in supporting local producers of fresh products in a particular area.

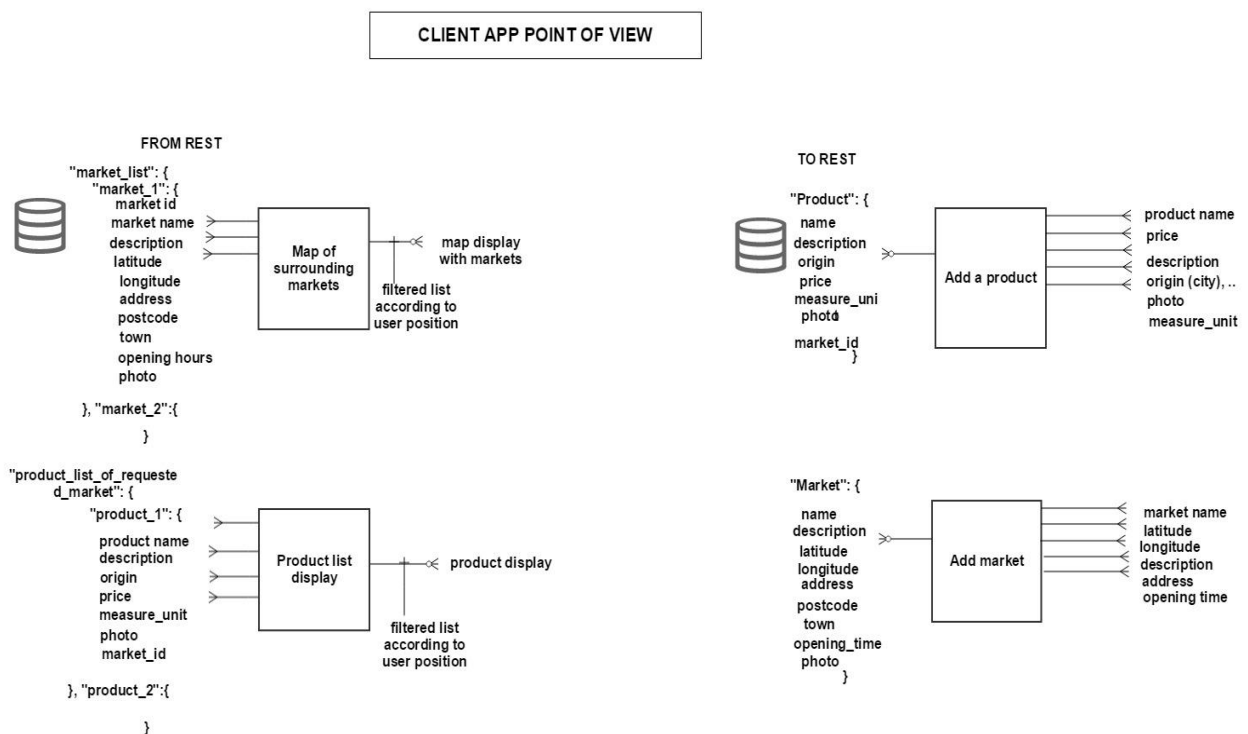
The application localEAT was built on a prosumer model, which means that the application is both a producer and consumer of open data. The aim is to encourage fresh food buyers and sellers participation by enabling them to enter new information about fresh products available in markets in a certain area. The greater the user participation the more open data becomes available. The

application also consumes open data about readily available location coordinates so as to interactively map current location as well as nearby markets within a certain range.

On a long term basis, a growing open data source about availability of fresh products in a locality can be useful to other businesses and services involved in the following areas: 1) Local planters and farmers who can get a better insight to source their harvested fruits, vegetables, fresh meat and dairy products 2) Restaurant chefs and caterers who are looking for local source of fresh products 3) Food manufacturers eager on decreasing their carbon footprint and increasing the sustainability of the their supply chain.

Therefore the goal of our API design is to be able to be expanded or to be used in creating new applications that deal with sustainable food supply and production. The datasets that we are producing are the fresh products that are available in nearby markets, listing all the local markets in a particular area in Finland, list all the available fresh products available in Finland.

Architecture of LocalEAT



The basic features of the application are summarised below:

- Consumers can browse food advertised from markets so that they can look for food which is local in a certain area and find what they need.
- Consumers can find the market locations and get an overview of the direction using integrated map of localEAT application.

- Producers and consumers can add market they know about as well as details concerning them.
- Producers can add their produced food name, picture, description and location.

Back-end Design

Node.js is a software platform that allows creating a webserver and building web applications. It contains a built-in HTTP server library, reducing the load of running a separate web server, giving more control over the web server work, but does increasing the complexity of running it. Meaning, NodeJS is Javascript running outside the browser, in the server

ExpressJS is a Node.js web application framework, designed for building, in our case, the hybrid web application, making the application creation and testing easy, while repeating common tasks. So, ExpressJS is web application framework that runs on NodeJS, allowing to build web applications and APIs endpoints.

The ability to store and use data is ensured by MongoDB – a document database, storing data in the form of documents with key and value pairs, and a group of documents, which are the analogous to a table in relational DB. Because of the urgency of having some structure in many applications a Mongoose driver is needed, for “elegant mongodb object modeling for node.js”.

The following project structure was initiated:

```

16 router.get('/create/', function(req, res) {
17   // render the test page, and pass Test POST page as the title
18   res.render('products', { title: 'Test POST Page' });
19 });
20
21 router.post('/', function(req, res, next) {
22   if(!req.body) {
23     res.status(500).send({error: "No body found!"});
24     return;
25   }
26   var newproduct = new productmodel({name: req.body.name, price: req.body.price, desc: req.body.desc,
27     origin:req.body.origin, measureunit: req.body.measureunit,
28     market_id: req.body.market_id,
29     photo: req.body.photo, });
30
31   newproduct.save(function(err){
32     if(err) return res.send(500, 'Error occurred: database error');
33     console.log('New product created');
34     res.json(newproduct);
35   });
36
37 });
38
39
40
  
```

Pic.1 – Back-end project structure

In *marketschema.js* and *productschema.js* we define the schemas - structures of our collections: markets and products. Each schema maps to a MongoDB collection and defines the shape of the documents within that collection.

In *products.js* and *markets.js* where was the creation of RESTful API, allowing users to use some of CRUD operations. In order to fetch data from specified resource GET method is used and POST is meant to submit data to a specified resource, which are two common HTTP Request used for building REST API's. A connection to a database is performed in *app.js* file. This is the minimum needed to connect our application database running locally on the default port (27017).

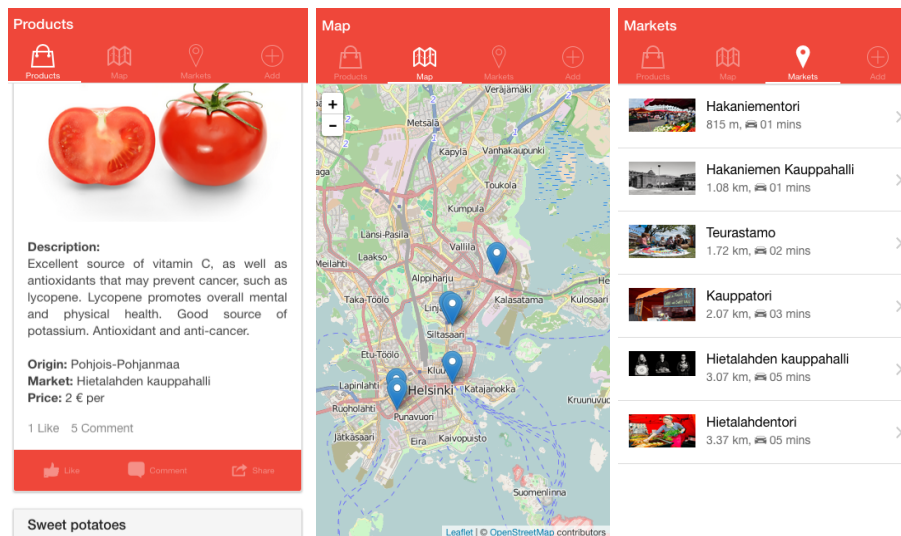
Front-end design

The front-end was built thanks to the ionic framework which implements new directives and style thanks to AngularJS over the Cordova Framework which allows writing HTML/JS applications for mobile use.

So far, the following features were implemented :

- A list of the products available in the markets fetched via http from our REST API
- A map listing the surrounding markets (using leaflet and OSM). Due to the lack of time, the pins are raw data structures copied from the database
- A list of the surrounding market limited by 25 KM of reach distance by car (OSRM API used for calculating it)
- The user interface which is overall almost in its final version. It is user centred.

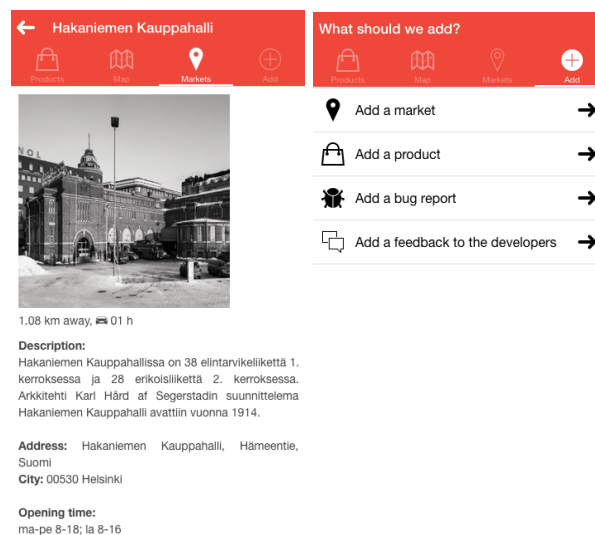
Results



Product page

Map page

Market list page



Market details page

Prototype of "Add menu"

Customers can browse surrounding products via the "Product page", see what markets are around them whether on a map or a list. They can also check details about a market and later, will be able to add it on their own. As far as the markers are concerned, they will be able to add their products as well.

Discussion

Our experience

The development wasn't as smooth as expected to be. Whenever a problem occurs with AngularJS, finding the exact cause can be a bit difficult. Moreover the main issues were faced due to the asynchronous nature of node (which is also its strength). Sometimes the data took more time to come to the application from REST than it takes for the process to start the display and in the following case the view won't always refresh so it is necessary to make sure that things get done one after another. Filter the data received from REST is also not easy for beginners in the language since it is necessary to understand the way AngularJS manage scopes and why the function won't always return the modified data.

Future developments

In the future, a search function might be added. This will help customers to look for product they want if the database gets big. Open API can serve other developers in developing their app. If people use competitors app, they will still reinforce a common database shared with the competitor and it will be profitable to everyone.

Conclusion

Open API provides the means for applications to be developed in order to provide useful and timely information to users.

LocalEAT was developed with the aim to provide consumers an application to see the nearby markets from their current location on a map in order to find fresh products and to allow suppliers to advertise their produce as well as to add the markets from which these are available. Through the open API allowing access to the information collected, it is possible for other services to use our freely available data and expand on the idea of sustainable food supply to create their own applications.