Application Design and Software Structure Report

Invitations App

1. Introduction

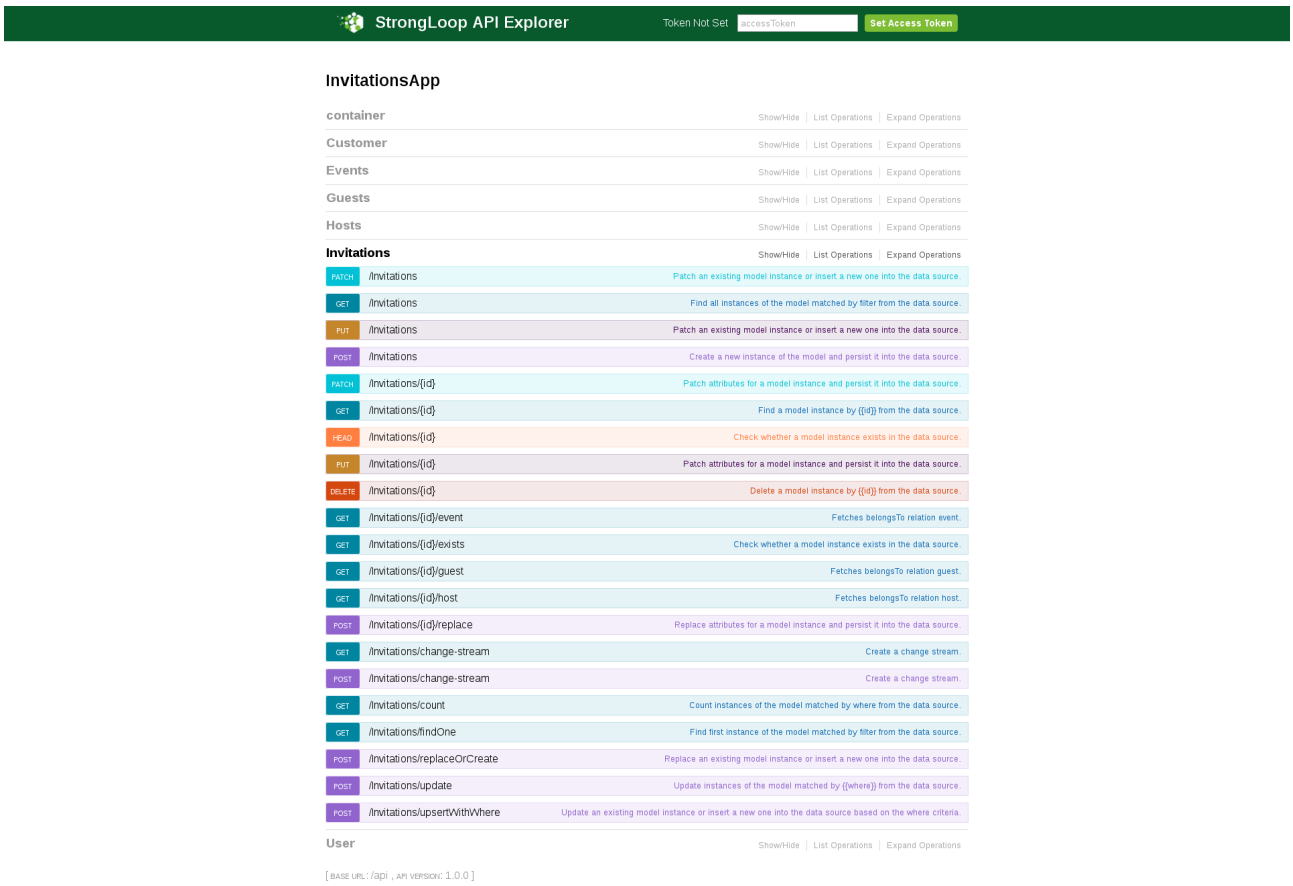
The invitationsAPP is an inovative software which can do the following:

• Be responsible for, given a person's/business' list of guests, send confirmation e-mails for events registered in the system.
• As a result, the system will provide accurate information to end-users about the number of people that should be participating in such event, enabling the end-user to plan the event's logistics ahead.

2. Design and Implementation

2.1 The REST API Specification

The application will provide end points for the main classes which will be constructed using strongloop Framework, and all classes have similar operations, and these ones can be seen in the picture below:



Each noun on the REST API represents a different class which is responsible for:

- User: Standard class generated by strongloop. Manages security aspects of the system, like login/logout;
- Customer: Inherits the User class atributes, and all operations related to users management will be done on this class;
- Host: A system user can create and manage different hosts;
- Guest: A system user can manage different guests;
- Event: A system user can manage different events, and create relationships between hosts and guests as well as send invitations to all guests within a given event planned by a host;

2.2 Front-end Architecture Design

The front end will be implemented using two different architectures. The first one will be a Web page, using a mobile first approach in a single page application, and this is the reason that the yo framework and yeoman process will be used to scaffold out the application, because this framework provides some of the best practices in the development of web sites, like the use of

angularJS, a famous MVC framework, and automation of tasks using grunt or gulp.

The other architecture will be a mobile application, and this will be achieved using the combination of Ionic framework, a hybrid development framework, which supports the use of HTML when programming for mobile devices, plus the cordova framework, which can enable the application to access some of native capabilities on the mobile devices.

2.3 Database Schemas, Design and Structure

The database to be used for this application will be the mongoDB, a javascript nonSQL database, and the documents will be organized in schemas, which will be modeled using mongoose, an object modeling used during this specialization.

Specifically talking about the schemas, these will have the following fields:

- Customer (inherits User):
    - Username;
    - Email;
    - Password;
    - Relates to Host (One to Many);
    - Relates to Guest (One to Many);
    - Relates to Event (One to Many);
- Host:
    - Name;
    - Email;
    - Phone;
    - Photo;
    - Contact Name;
    - Contact Position/Title;
    - Relates to Customer (Many to One);
    - Relates to Guests (One to Many);
    - Relates to Events (One to Many);

- Guest:
    - Name;
    - Email;
    - Phone;
    - Photo;
    - Relates to Customer (Many to One);
    - Relates to Hosts (One to Many);
    - Relates to Events (One to Many);
- Event:
    - Name;
    - Start Date;
    - End Date;
    - Confirmation Date;
    - Description;
    - Status;
    - Relates to Customer (Many to One);
    - Relates to Guest (Many to One);
    - Relates to Host (Many to One);

About the data access control, we will have the following rules:

- Unauthenticated users will only be able to register new customers in the application database;
- the admin profile can read, write and delete data in all documents;
- the normal user can read, write and delete data only in their own documents;

2.4 Communication

As the system is designed to be Restful, all the messages changed between the front-end and back-end will be in the the json format, and these messages will be sent through the end points defined before using the http verbs for each class designed.

3. Conclusions

To achieve a fully fledge website/mobile application responsible for maintaining different events in different platforms, the project will follow this development sequence:

- Scaffold out and develop the REST API using Strongloop Framework;
- Design database schemas using mongoDB/mongoose;
- Scaffold out and implement the responsive website using yo framework;
- Scaffold out and implement the mobile apllication using ionic and cordova framework;

4. References

1. *REST – https://en.wikipedia.org/wiki/Representational_state_transfer;*
2. *Strongloop framework – https://strongloop.com/;*
3. *MongoDB – https://www.mongodb.com/;*
4. *Mongoose – http://mongoosejs.com/;*
5. *Yeoman framework – http://yeoman.io/;*
6. *Ionic Framework – https://ionicframework.com/;*
7. *Cordova Framework – https://cordova.apache.org/*