

EVENT.LY:

Client-Side:

I. ROUTES (in app.js)

Route	templateUrl	Controller	Services
/	../views/auth.html	AuthController	AuthServices
/signup	../views/auth.html	AuthController	AuthServices
/login	../views/auth.html	AuthController	AuthServices
/logout	../views/auth.html	AuthController	AuthServices
/newEvent	../views/newEvent.html	EntryController	Services
/listEvents	../views/listEvents.html	EventListController	Services
/deleteEvent	../views/deleteEvent.html	DeleteController	Services

II. FUNCTIONALITIES:

A.) EVENTS:

1. Create Event (**Required fields in bold**)
 - a. **Email**
 - b. **Label**
 - c. **Date**
 - d. **Time**
 - e. Street Address
 - f. City
 - g. State
 - h. **Zip Code**
 - i. Indoor/outdoor
 - j. (background) Weather notice or unavailable
 - k. (background) Status change

****WEATHER API call in this function****

2. Retrieve Event (by user)
 - a. Same info as create
3. Delete Event (by eventID)

B.) USERS:

1. Add User
2. Log in User
3. Log out User (pending)

SERVER-SIDE:

I. ROUTES (config/routes.js)

Request	Url	Controller	Function
POST	/api/signup	userController	addOneUser
POST	/api/login	userController	loginUser
GET	/api/event	eventController	getEventByOwner
POST	/api/event	eventController	addOneEvent
DELETE	/api/event/:id	eventController	deleteThisEvent

II. AUTHENTICATION: (config/helpers.js, config/middleware.js)

1. jwt-simple
2. bcrypt-nodejs

III. FUNCTIONALITIES:

A.) EVENTS:

1. Add Event ****WEATHER API call in this function****
2. Delete Event
3. Update Event
4. Retrieve Event (by user)

B.) USERS:

4. Add User
5. Log in User
6. Log out User (pending)

IV. EVENT TABLE (model/event.js)

COLUMN NAME	SQLITE DATA TYPE	JS DATA TYPE	FORMAT/EXAMPLE
Event ID	integer	Number	1,2,3,4...
Event Owner	text	String	bob@email.com
Title	text	String	Fun Party
Date	Date	String	YYYY-MM-DD
Time	Time	String	HH:MM:SS
Street address	text	String	123 Main

City	text	String	San Francisco
State	text	String	CA
Zip code	integer	Number	94114
Latitude	real	Number	12.2304234
Longitude	real	Number	-144.82371289
Indoor/Outdoor	integer	Number	0: Indoor 1: outdoor
Estimated weather	text	Stringified Obj	????
Weather status	integer	Number	-1: bad 0:unk 1:yay
Public/Private	integer	Number	0: public 1: private
Weather Icon	text	String	
Last Update	text	String	
userTableID (F. key)	integer	Number	

IV. USER TABLE: (model/user.js)

COLUMN NAME	SQLITE DATA TYPE	JS DATA TYPE	FORMAT/EXAMPLE
user ID	integer	Number	1
email	text	String	abc@email.com
password	text	String	password1

REFERENCES:

SQL Data Types: <https://www.sqlite.org/datatype3.html>

WunderGround API: <https://www.wunderground.com/weather/api/d/docs>

SQLite Date: http://www.sqlite.org/lang_datefunc.html

MOVING FORWARD:

- Friends (Event owner/attendees access)
- Attendees - decline option
- Notification outside of app (CRON)
- Link to user's google calendar
- Warning of potential travel time conflict
- Map view of event location
- Parameters for weather alerts