Instructions:

Today you will be building a Node / Express based web applications for handling reservation requests.

Your application will be made up of two parts: 1) A front-end set of HTML/CSS/JS pages for entering and viewing data and 2) A back-end composed of Node/Express and basic JS for storing, updating, and relaying reservation data.

Spend the time necessary to map this application out. Consider the concepts we've covered in class so far:

Servers

Routing

APIs

AJAX (GET and POST Requests)

You should be referencing the code from the previous Star Wars application.

Feel encouraged to use the following application as a reference: http://hot-restaurant.herokuapp.com/

Note: We know this is a hard activity. We know you aren't yet comfortable with Node or Express. But push yourself. The best way to learn is to push through the discomfort and BUILD! Ask for help when you need it. We're here to help you through the process.

Bonus:

If by some miracle you finish early, feel encouraged to work on any one of these additional tasks to take your application to the next level.

Add code to your server so that it "counts" every time a person visits any of the pages. Display a running count on the website.

Add buttons for "checking off" individuals from the reservation list. Once this happens, the next person on the waitlist should be addd to the main reservation list.

Add a button for sending emails to individuals on the wait-list letting them know they have a table ready. You will need to use node-mailer or a similar npm package to make this work. (A hard task, but very cool).

Add a button for sending text messages to individuals on the wait-list letting them know they have a table ready. You will need to use the Twilio library to make this work. (A hard task, but very cool). Instructions:

Today you will be building a Node / Express based web applications for handling reservation requests.

Your application will be made up of two parts: 1) A front-end set of HTML/CSS/JS pages for entering and viewing data and 2) A back-end composed of Node/Express and basic JS for storing, updating, and relaying reservation data.

Spend the time necessary to map this application out. Consider the concepts we've covered in class so far:

Servers

Routing

APIs

AJAX (GET and POST Requests)

You should be referencing the code from the previous Star Wars application.

Feel encouraged to use the following application as a reference: http://hot-restaurant.herokuapp.com/

Note: We know this is a hard activity. We know you aren't yet comfortable with Node or Express. But push yourself. The best way to learn is to push through the discomfort and BUILD! Ask for help when you need it. We're here to help you through the process.

Bonus:

If by some miracle you finish early, feel encouraged to work on any one of these additional tasks to take your application to the next level.

Add code to your server so that it "counts" every time a person visits any of the pages. Display a running count on the website.

Add buttons for "checking off" individuals from the reservation list. Once this happens, the next person on the waitlist should be addd to the main reservation list.

Add a button for sending emails to individuals on the wait-list letting them know they have a table ready. You will need to use node-mailer or a similar npm package to make this work. (A hard task, but very cool).

Add a button for sending text messages to individuals on the wait-list letting them know they have a table ready. You will need to use the Twilio library to make this work. (A hard task, but very cool).