Install Guide
Webapp

Pre-requisites:
Any system, preferably built on Unix, is capable of running CookForMe. The system have internet access in order to connect to Amazon servers.

Dependent libraries:
Node.js (v6.10.2 or above)
All node packages as specified by included package.json files
Git, though it is not necessary for downloading.

Download instructions:
With Git:
Clone the CookForMe Github repository into a directory of your choice.
`$git clone https://github.com/Mete0/cook-for-me`
Without Git:
Download the CookForMe repository as a .zip file via
https://github.com/Mete0/cook-for-me/archive/master.zip
and extract all files into a directory of your choice.

Build instructions:
The application comes with libraries that will build the application when it is run.

Installation of actual application:
In the CookForMe root directory, run npm install.
Next, navigate into the webapp directory, and run npm install.
From the root directory, the list of commands needed should look like:
```
$npm install
$cd webapp
$npm install
```
The application should now have all the node dependencies it needs to run.

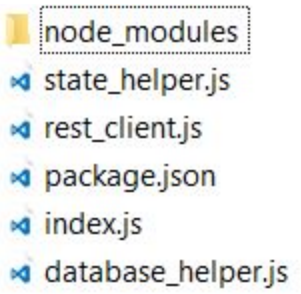
Run instructions:
In the webapp directory, the frontend component and backend need to be run separately. Two separate terminal tabs or windows should be open.
In the webapp directory, execute the following command:
`$npm run start`
In the server directory within webapp, and in a separate tab, execute the following command:
`$node index.js`
Both webapp components should now be running.

Alexa Skill

Pre-requisites:
Program to handle zip files or git

Download instructions:
With Git:
Clone the CookForMe Github repository into a directory of your choice.
`$git clone https://github.com/Mete0/cook-for-me`
Without Git:
Download the CookForMe repository as a .zip file via
https://github.com/Mete0/cook-for-me/archive/master.zip
and extract all files into a directory of your choice.

Add the following files and folder to a zip file



Uploading to AWS

1. If you do not already have an account on AWS, go to Amazon Web Services and create an account.
2. Log in to the AWS Management Console and navigate to AWS Lambda.
3. Click the region drop-down in the upper-right corner of the console and select either **US East (N. Virginia)** or **EU (Ireland)**.
4. Lambda functions for Alexa skills *must* be hosted in either the **US East (N. Virginia)** or **EU (Ireland)** region.
5. If you have no Lambda functions yet, click **Get Started Now**. Otherwise, click **Create a Lambda Function**.
6. When prompted to configure triggers, click the box and select Alexa Skills Kit, then click Next.
7. When prompted to configure triggers, click the box and select **Alexa Skills Kit**, then click **Next**.
8. Enter a **Name** and **Description** for the function.
9. Select node.js as the runtime

10. For **Role** (under **Lambda function handler and role**), select **Create new role from template(s)**.
11. Enter the **Role Name**.
12. From the **Policy templates** list, select **Simple Microservice permissions**
13. Select the **Triggers** tab.
14. Click **Add trigger**.
15. Click the outlined box and choose **Alexa Skills Kit**.
16. Click **Submit**.
17. Select the **Upload a .ZIP file** option and upload the zip file you created earlier.
18. Make note of the Amazon Resource Name (ARN) for your new Lambda function.

    The **ARN** is displayed in the upper-right corner of the function page.

Creating the Alexa Skill

Log on to the Developer Portal.

1. Navigate to the Alexa section by clicking **Apps & Services** and then clicking **Alexa** in the top navigation.
2. In the Alexa Skills Kit box, click **Get Started**.
3. Find the skill in the list and click **Edit**.
4. On the Skill Information page, copy the **Application Id** shown.
5. Fill in the infomration on the Skill Information page
6. Click on **Interation Model**
7. Copy paste the following information into the Intent Schema

```
{
  "intents": [
    {
      "intent": "AMAZON.CancelIntent"
    },
    {
      "intent": "AMAZON.StopIntent"
    },
    {
      "intent": "AMAZON.HelpIntent"
    },
    {
      "intent": "load_intent"
    },
    {
      "slots": [
        {
          "name": "CONTINUE",
          "type": "CONTINUETYPE"
```

```
        }
      ],
      "intent": "continueIntent"
    },
    {
      "slots": [
        {
          "name": "SEARCH",
          "type": "SEARCHTYPE"
        }
      ],
      "intent": "searchIntent"
    },
    {
      "slots": [
        {
          "name": "QUERY",
          "type": "AMAZON.Food"
        }
      ],
      "intent": "queryIntent"
    },
    {
      "intent": "storedRecipesIntent"
    },
    {
      "intent": "beginSearchIntent"
    },
    {
      "slots": [
        {
          "name": "SELECT",
          "type": "AMAZON.NUMBER"
        }
      ],
      "intent": "selectIntent"
    },
    {
      "intent": "ingredients_intent"
    },
    {
      "slots": [
        {
          "name": "MULTIPLIER",
          "type": "AMAZON.NUMBER"
        }
      ],
      "intent": "multiplier_intent"
```

```
    },
    {
      "slots": [
        {
          "name": "STEPS_SELECTION",
          "type": "STEPS_CHOICE"
        }
      ],
      "intent": "steps_choice_intent"
    },
    {
      "slots": [
        {
          "name": "USERSTEP",
          "type": "STEPS_MOVE"
        }
      ],
      "intent": "step_by_step_intent"
    },
    {
      "intent": "save_intent"
    }
  ]
}
```

8. Add the following custom slot types

Custom Slot Types (Optional)
Custom slot types to be referenced by the Intent Schema and Sample Utterances. For general information about custom slots, see Custom Slot Types.

Type	Values		
CONFIRM_TITLE	confirm | redo	Delete	Edit
CONTINUETYPE	continue | new session	Delete	Edit
SEARCHTYPE	keyword | keywords | ingredient | ingredients	Delete	Edit
SELECTNUMBER	one | two | three	Delete	Edit
STEPS_CHOICE	all at once | step by step	Delete	Edit
STEPS_MOVE	previous | next | repeat	Delete	Edit
TEXT	anything | any value	Delete	Edit

9. Copy/paste the following into **Sample Utterances**

```
load_intent favorite
load_intent load favorite
load_intent favorites
load_intent load favorites
load_intent saved
load_intent load saved
load_intent favorite recipe
load_intent load favorite recipe
load_intent favorites recipe
load_intent load favorites recipe
load_intent saved recipe
load_intent load saved recipe
load_intent favorite recipes
load_intent load favorite recipes
load_intent favorites recipes
load_intent load favorites recipes
load_intent saved recipes
load_intent load saved recipes
continueIntent {CONTINUE}
searchIntent search {SEARCH}
searchIntent find {SEARCH}
searchIntent search by {SEARCH}
searchIntent find by {SEARCH}
queryIntent {QUERY}
storedRecipesIntent stored recipes
beginSearchIntent begin search
selectIntent {SELECT}
ingredients_intent what ingredients
ingredients_intent what are ingredients
ingredients_intent what the ingredients
ingredients_intent what are the ingredients
multiplier_intent multiplier {MULTIPLIER}
multiplier_intent set multiplier {MULTIPLIER}
multiplier_intent multiplier to {MULTIPLIER}
multiplier_intent set multiplier to {MULTIPLIER}
steps_choice_intent {STEPS_SELECTION}
step_by_step_intent {USERSTEP}
step_by_step_intent {USERSTEP} step
save_intent save
save_intent save recipe
```

10. Under the **Configuration** tab, add the ARN that you recorded previously

11. Under the **Test** tab, you may now enable and test the skill