Lightning Out - Heroku Quickstart

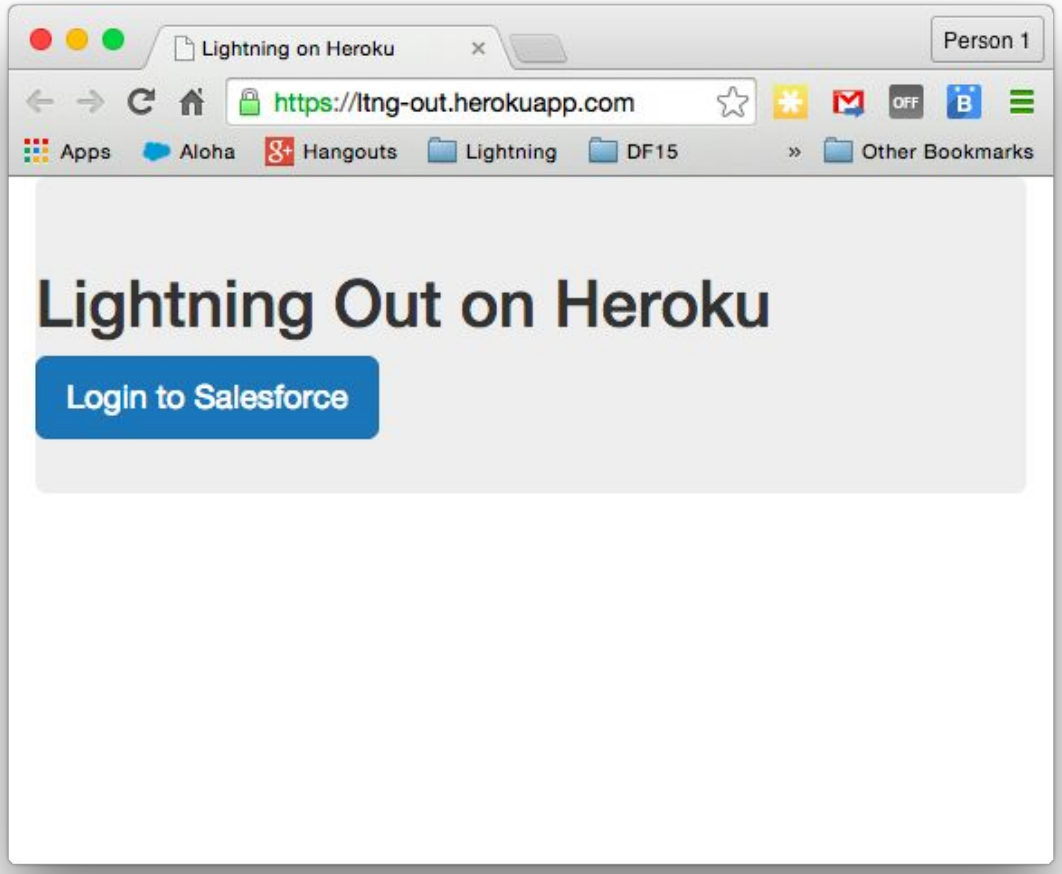
Prerequisites

Salesforce Developer Edition (DE) org on Winter ‘16 with My Domains enabled
Heroku account with Heroku Toolbelt
Familiarity with Lightning Components - Trailhead!
Git on your local environment, and reasonable familiarity with it

Create the Remote App

1. Repository link: https://github.com/skipsauls/heroku-lightning.git
2. Open a shell/terminal/etc.
3. Navigate to the parent directory for your project:
   `$ cd dev`
4. Clone the repository:
   `$ git clone https://github.com/skipsauls/heroku-lightning.git`
5. Change to the newly created directory:
   `$ cd heroku-lightning`
6. Install the Node.js modules:
   `$ npm install`
7. Login to Heroku, if you haven’t already done so:
   `$ heroku login`
8. Create the Heroku app:
   `$ heroku create`
   or
   `$ heroku create my-app`
9. Push the repository to Heroku
   `$ git push heroku master`
10. Ensure the app is running on Heroku:
    `$ heroku ps:scale web=1`
11. Open a browser to see the app running on Heroku:
    $ heroku open

12. You should see the following in a browser window:



Configure the Connected App and CORS

1. Login to your DE org on a Winter '16 or later instance
2. Create a new Connected App
    a. Navigate to the Apps Setup page:
    Setup > Build > Create > Apps
    b. Click the *New* button in the Connected Apps section
    c. Enter the following values for the new app:
    Connected App Name: ltng heroku
    API Name: ltng_heroku
    Contact Email: you@yourdomain.com
    d. Toggle the *Enable OAuth Setting*s checkbox
    e. Enter the Callback URL (change as needed):
    https://ltng-out.herokuapp.com/oauthcallback

    f. Configure the OAuth Scopes
       Add *Full Access (full)* to the Selected OAuth Scopes
    g. Save the newly configured app
    h. Continue to the Connected App details
    i. Copy the Consumer Key to the clipboard
3. Configure CORS for the remote site
    a. Open the CORS Setup page:
       Setup > Administer > Security Controls > CORS
    b. Click the New button
    c. Enter the base URL for the remote app (no trailing slash):
       https://ltng-out.herokuapp.com
    d. Save the new whitelist entry
4. Create the Lightning Application
    a. Open the Developer Console
       Your Name > Developer Console
    b. Create a new Lightning App
       File > New > Lightning Application
    c. Enter the name for the app, e.g., *loTest* and click the Save button
    d. Add the dependencies for the Lightning Components

```
<aura:application access="GLOBAL" extends="ltng:outApp">
    <aura:dependency resource="forceChatter:feed"
                     type="COMPONENT"/>
</aura:application>
```

    e. Save the Lightning Application
       File > Save

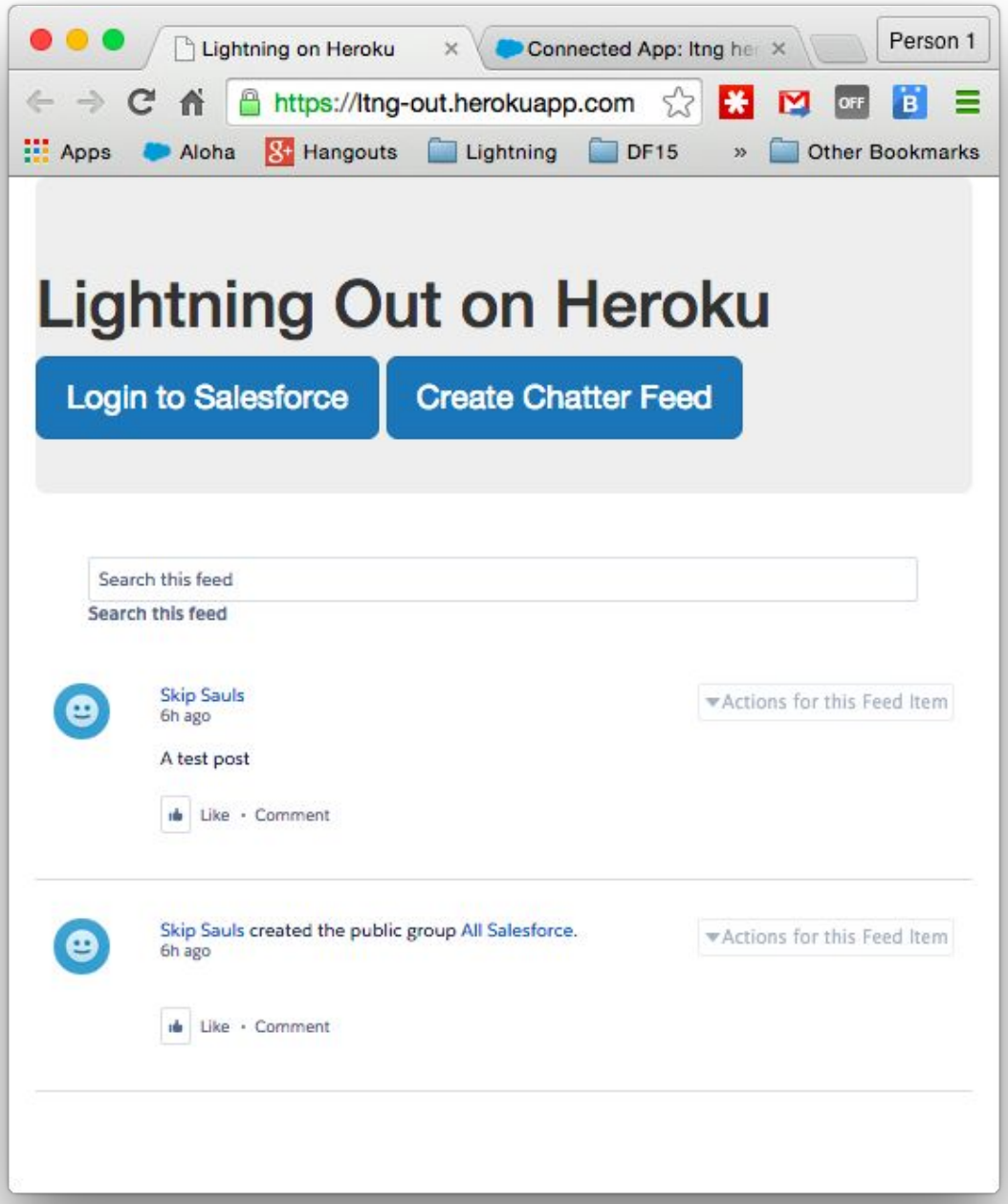
Configure the Environment Variables for the Heroku App

1. Open the terminal/shell/etc.
2. Navigate to the Heroku app
   `$ cd dev/heroku-lightning`
3. Set the APPID environment variable to the Connected App Consumer Key:
   `$ heroku config:set APPID=`*`Consumer Key copied from the Connected App`*
4. Set the LOAPP environment variable to the name of the Lightning App:
   `$ heroku config:set LOAPP=c:loTest`

Test the App

1. Open the browser window with the app, or use *heroku open*
2. Refresh the page
3. Login to Salesforce using your username/password
4. Click *Allow* on the *Allow Access?* page
5. When the *Create Chatter Feed* button appears, click it

6. The Chatter Feed component should appear in the Heroku app:



Next Steps

This is meant to be a straightforward introduction to Lightning Out, barely scratching the surface of what's possible. It should work with most any web, app, and client stack, and we've already tried it with PHP, SAP HANA, SharePoint and other Microsoft technologies, and of course

Node.js. If you do use it on other technology stacks, we'd love to hear about it. If you can provide a simple example such as this, we'll be happy to add it to a list of proven integrations.

For support, please use the Lightning Out Chatter Group on 62org, the forums on developer.salesforce.com, and other forums/groups. Please let us know of any issues, suggestions for improvements, etc..