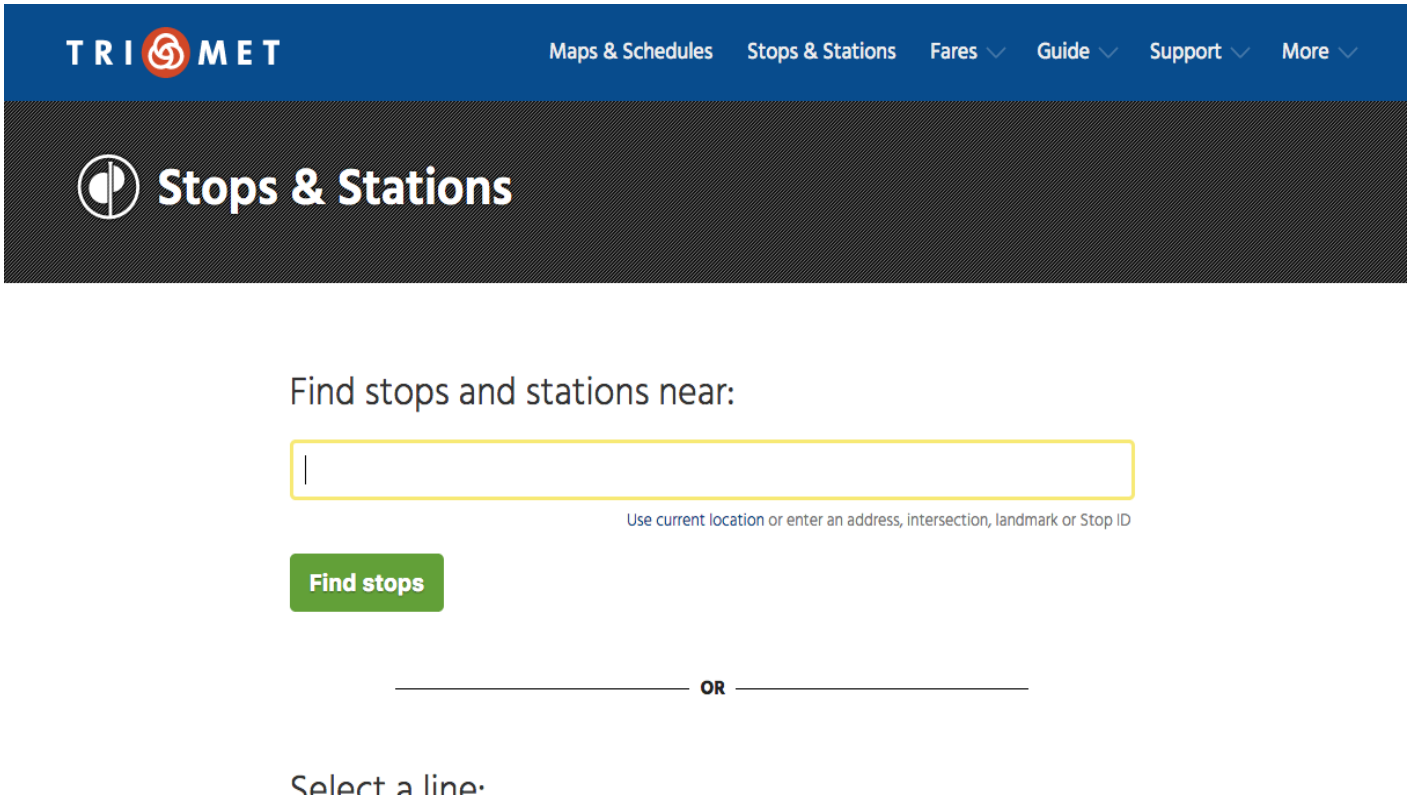
Headers & Footers

Hopefully a simple explanation

History

- Originally, there was a Stops & Schedules App.
- It was built with Mako templates, and which the graphic designers were comfortable editing.



History

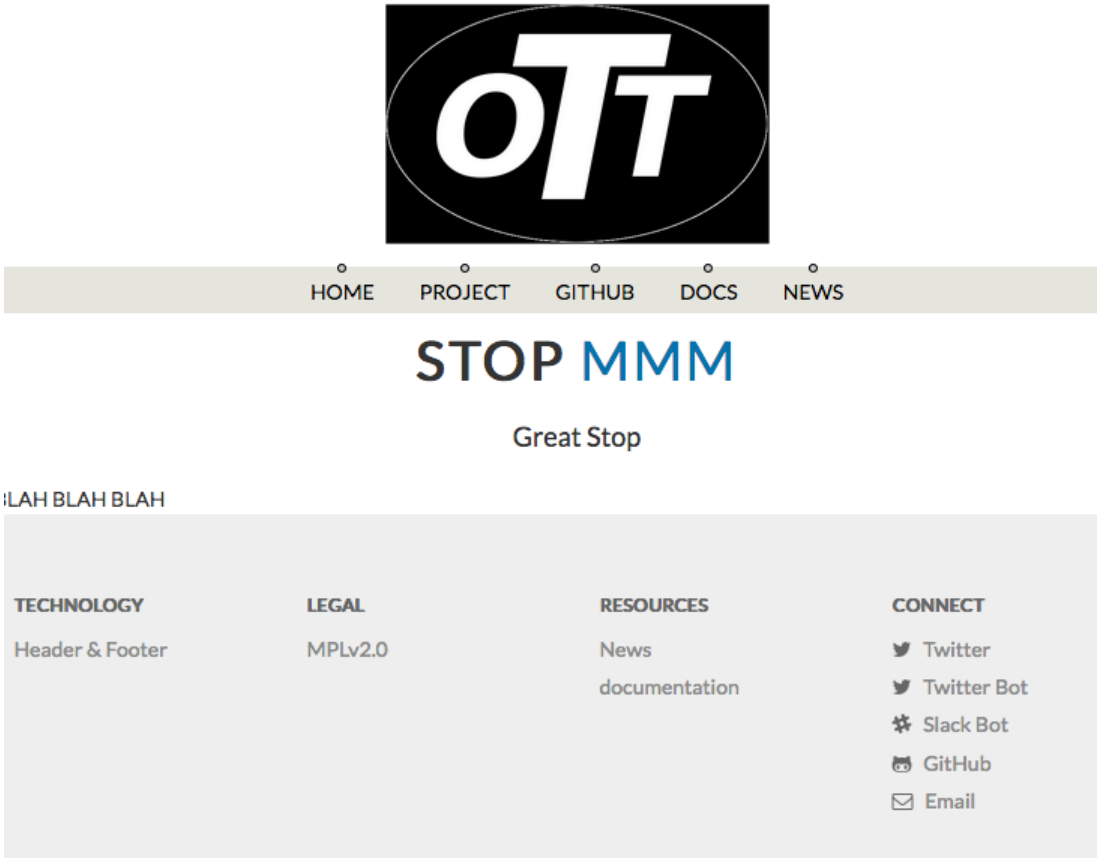
- I was recently asked to put a Header & Footer on the fieldtrip form: http://fieldtrip.trimet.org
- This app uses Java's Play! Framework (not Py/Pyramid).
- In-terms of Headers & Footers, dealing with another framework means creating a 2nd set of H&F templates.
- It's not a good solution, since now our designer must maintain another set of templates anytime the header / footer changes.
- This cut & paste approach just doesn't scale...and yes, there are other apps, that are on other platforms (e.g., timetables) that we'd like to add a Header & Footer to.

Header & Footers Service

- So rather than propagate multiple sets of header and footer templates, I adapted the original Stops and Stations project into a stand-alone Headers & Footers service.
- Going in our favor, this is a system that our designers are already comfortable working in.
- So now, anyone who want's a TriMet Header can call: http://dev.trimet.org/ride/header.html
- And for a Footer: http://dev.trimet.org/ride/footer.html
- And with those two *strings*, sandwich some goodness between, and ship back a complete look & feel webpage.

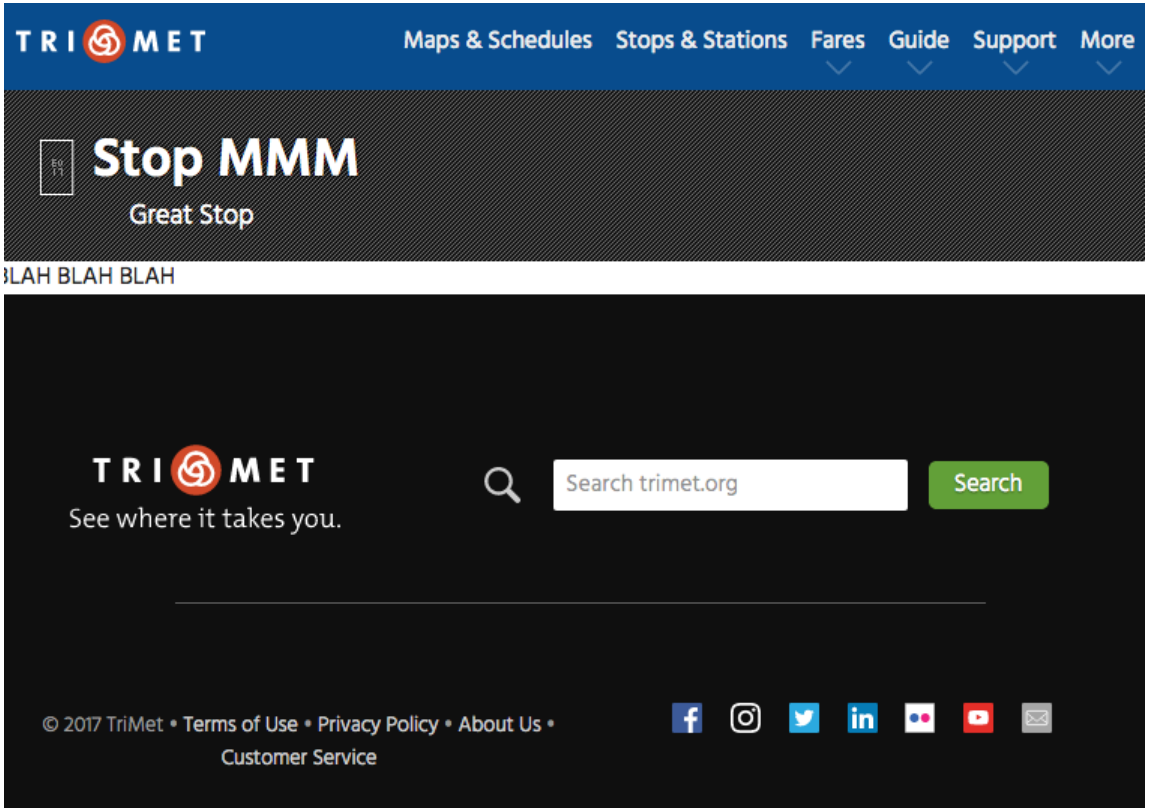
OTT Project

- Because Stops & Schedules started as an OpenSource project, I wanted to maintain some presence there...



TriMet Header & Footer

- To customize the default OTT look, a new Pyramid project is created. This custom Header & Footer project, for the most part, reuses things (pyramid views and setup, Mako templates, etc..) without change.
- In a small handful of places (2 Mako files), customizations overwrite the default OTT Mako templates. The base H&F system pulls in these customizations, and the look & feel is magically changed.



3 Options

- An app that wants to use the Header & Footer service has 3 options:
  1. Each app request can make a set of corresponding GET requests to header.html and footer.html, then return those strings in the app's response, with some content inserted between these header & footer strings.
  2. A pyramid app can import the H & F pyramid app as a dependency, and then access header.html & footer.html locally, and sandwich content in between those strings.
  3. At build time, a page (or template) can be built by pulling a header.html & footer.html from the service, and outputting a file that has the header & footer strings statically placed.

     NOTE: I use #2 (Stops & Schedules) and #3 (fieldtrip) today