OTP SUM: OTP Integration of Transit with Shared-Use Mobility Real-Time and Data Enhancements

Mobility on Demand Sandbox Program
Quarterly Report Q4 2017
10/01/17 - 12/31/17

Published March 1, 2018

TABLE OF CONTENTS

Project Summary

A project dashboard is available at www.trimet.org/mod. It provides more comprehensive information about the project and up-to-date status reports.

Challenges Addressed by Project

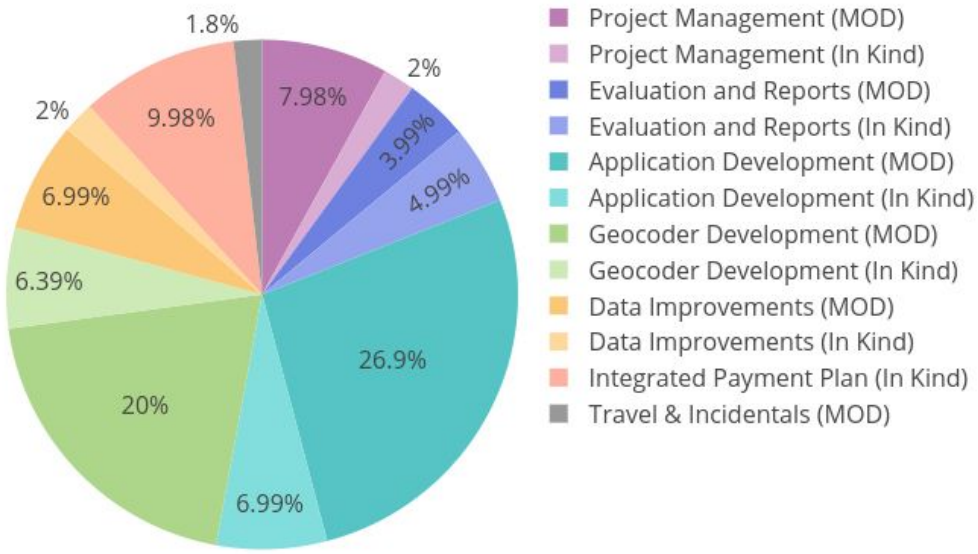
- OpenTripPlanner (OTP) does not currently incorporate shared-use modes.
- Address location for trip origins and destinations are a main requirement for trip planning, however, existing options are inadequate or cost prohibitive for government.
- Accessible trips are a challenge due to the lack of data available on the accessibility of pedestrian infrastructure and the absence of these features in a trip planner.

Anticipated Outcomes, Benefits, Impacts

- Extend the OpenTripPlanner code base to support the integration of transit trip planning with shared-use mobility modes, such as bike share and transportation network companies (TNCs), as well as updated real-time transit information.
- Implement a fully functional and comprehensive open geocoder built off the existing Mapzen Pelias geocoder. A non-proprietary and non-restrictive option for address locating would substantially lower the barrier to entry for many transit systems to offer trip planning and can achieve significant cost savings for transit agencies, government agencies, and the public.
- TriMet, in collaboration with the OpenStreetMap community, established best practices for representing accessibility information and will build out this accessibility information in the OSM network and provide a model for replicating this work in other regions.

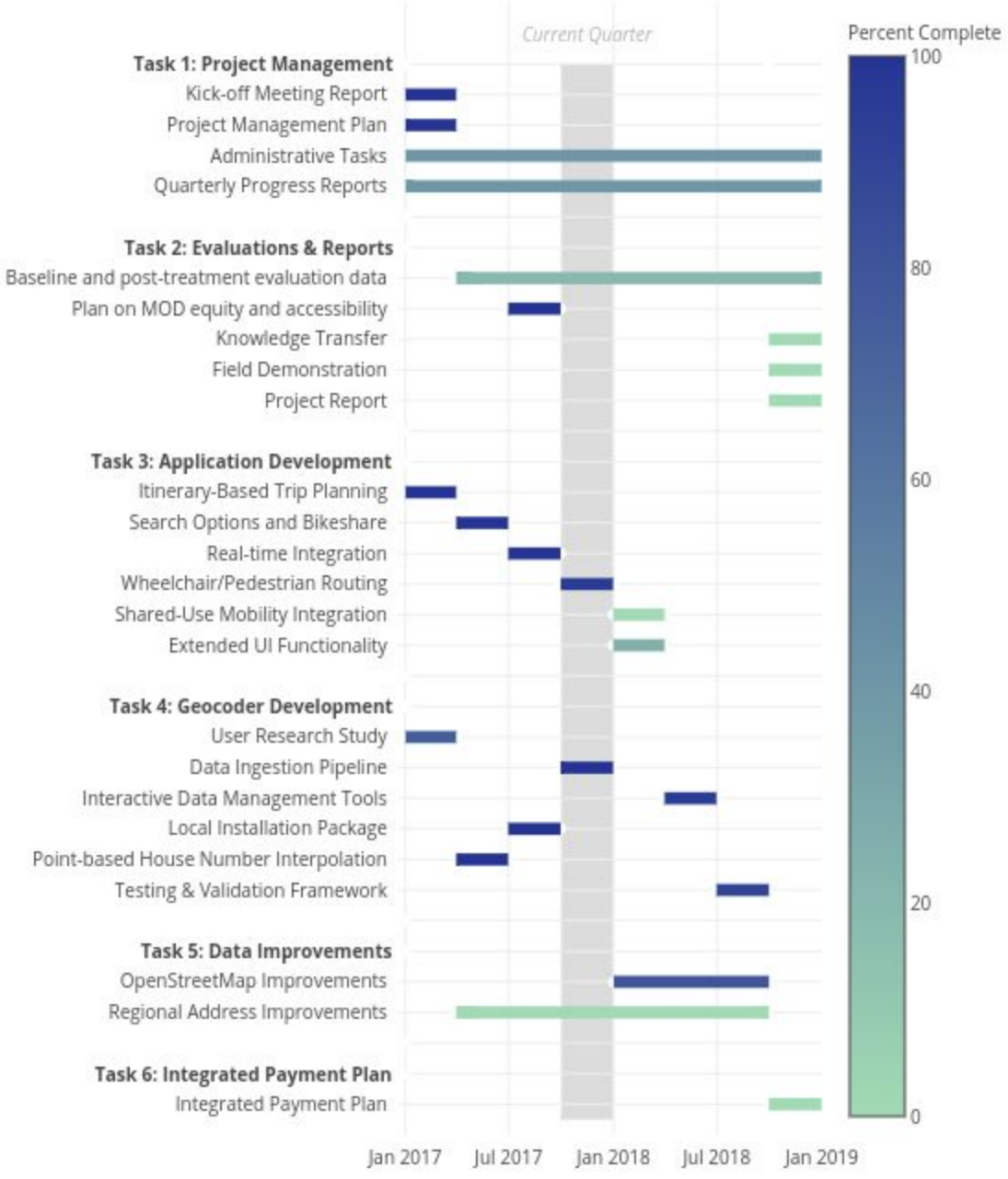
Grant Budget Allocations

TriMet's funding allocation from the FTA of $678,000 is matched with 32% of in-kind contributions, totaling over $1 million.



Project Scope and Budget Status

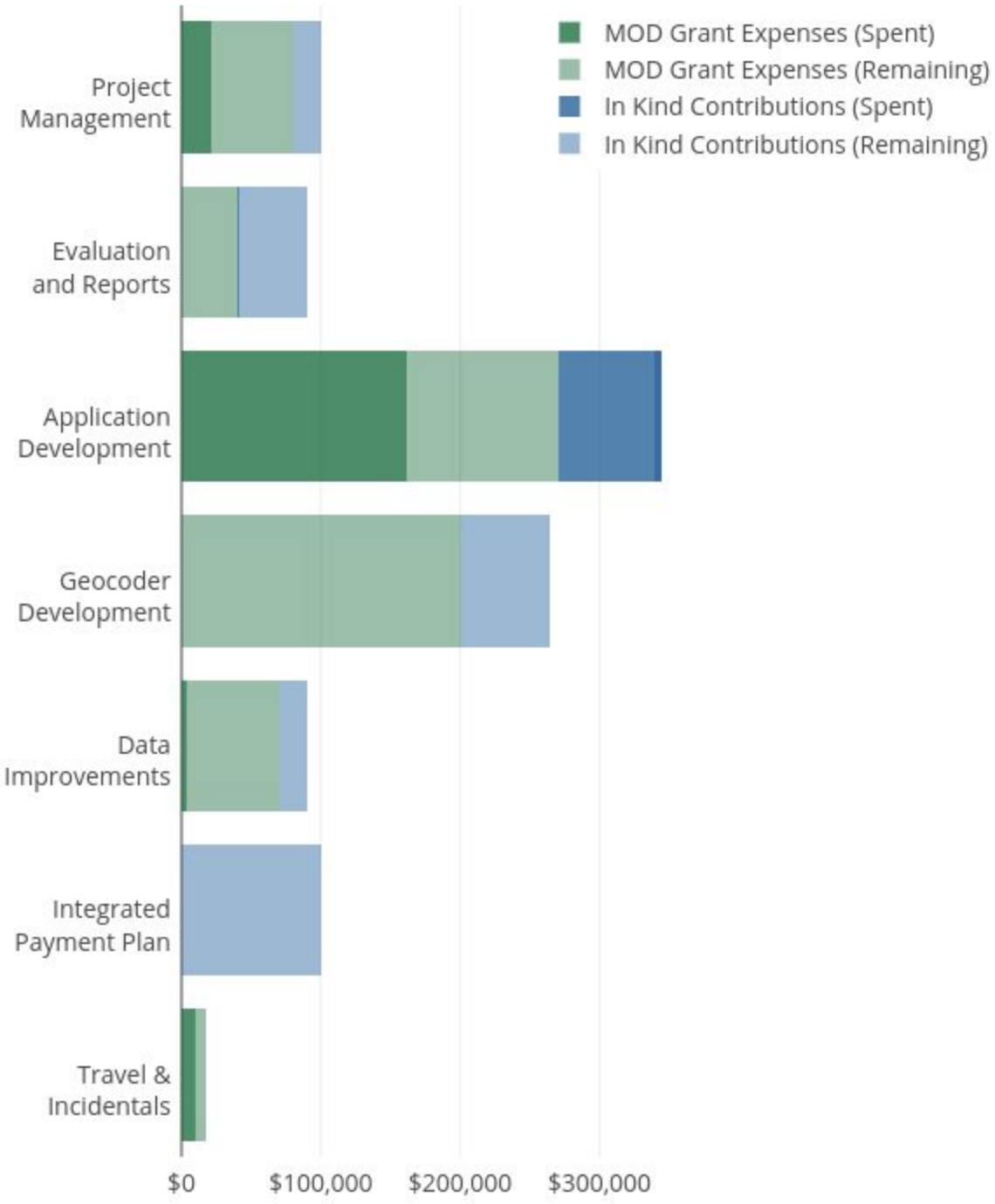
The MOD Sandbox project is divided into six main tasks: Project Management, Evaluations & Reports, Application Development, Geocoder Development, Data Improvements and an Integrated Payment Plan. The project is on schedule and in budget. Progress is as follows:



The above gantt chart illustrates the tasks and status of deliverables in Quarter 4.

Of the $678,000 that TriMet received, $193,113 (28.5% of allocated grant funds) has been spent thus far. The cleared expenditures through Q4 2017 are as follows:

- $21,669 (27% of allocated grant funds) spent toward Project Management;
- $162,000 (60% of allocated grant funds) spent toward Application Development;
- $0 spent toward Geocoder Development;
- $796 (1% of allocated grant funds) spent toward Data Improvements;
- $8,648 (48% of allocated grant funds) spent toward Travel & Incidentals.



The above bar chart shows the current amount spent for each of the tasks in Quarter 4.

Task 1: Project Management

TriMet's OTP Integration of Transit with Shared-Use Mobility Real-Time and Data Enhancements have been underway since January. All milestones and deliverables have been met and we are on schedule.

Quarterly Deliverables

Deliverables for this quarter are in the form of ongoing tasks that include scheduled weekly meetings and administrative tasks.

Quarterly Progress

Task progress includes:

- weekly scheduled meetings (slack or webinars) to ensure continued communications;
- use of Trello for project management;
- a dedicated and open TriMet MOD Project Google drive for project management;
- use of InVision for application interface development and review;
- continued update of the online project dashboard available to the public at TriMet.org/MOD to ensure transparency;
- and RealTime Board for live, remote whiteboarding sessions.

Task 2: Evaluations and Reports

The FTA requires the following project evaluations and reports: Evaluation Plan and Report, Equity and Accessibility Plan, Knowledge Transfer, Field Demonstration, Final Project Report.

Quarterly Deliverables

Evaluation and Test Plan for Application (**Appendix A - MOD Application Test Plan**).

Quarterly Progress

Inquiries into local firms have been made to perform usability studies.

Task 3: Application Development Status

A live demo of the application is now available at https://trimet-mod-dev.conveyal.com/

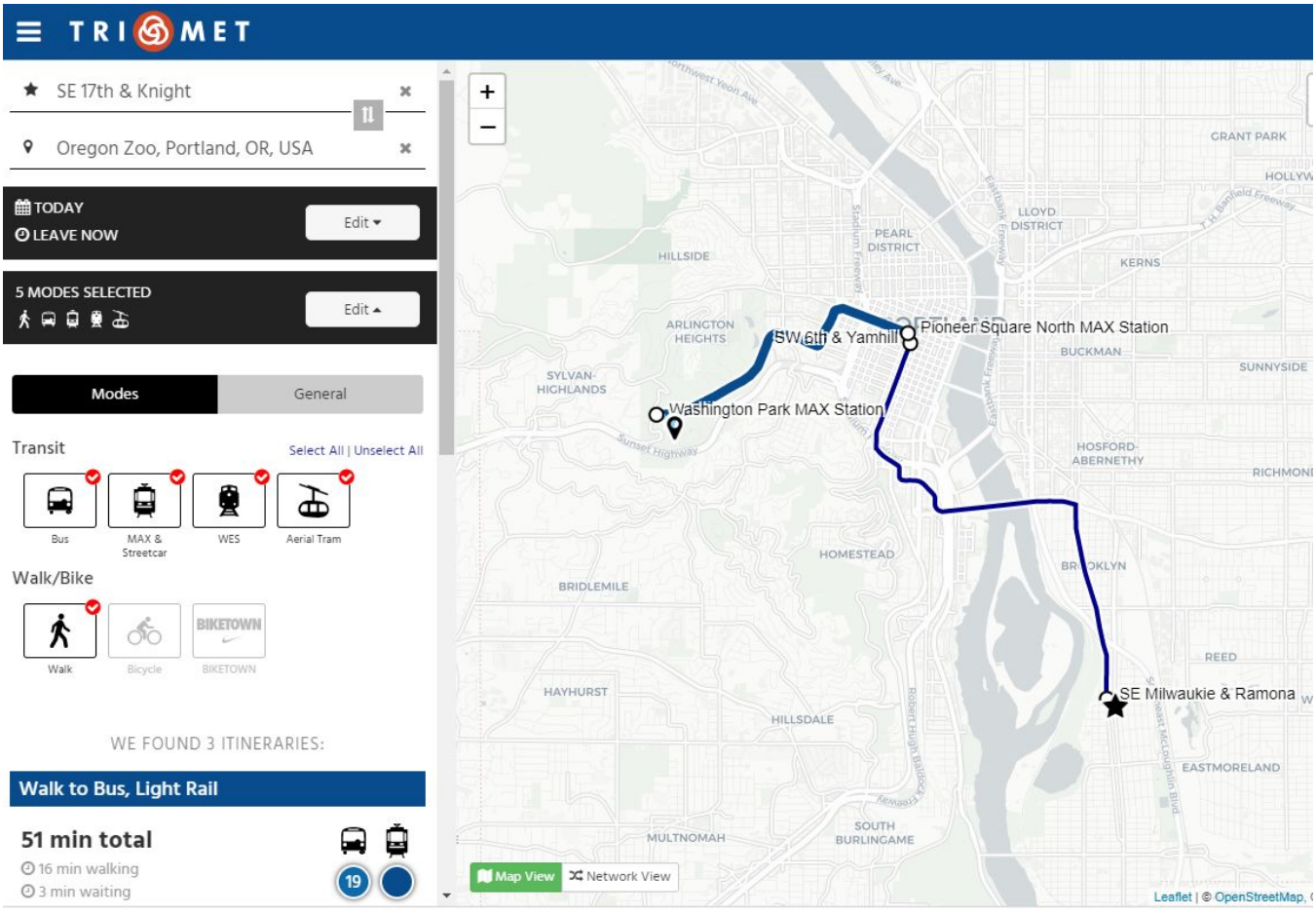
Quarterly Deliverables

Search Options and Bikeshare (**Appendix B - T3M4 RouteStopViewer_PedImprvmts_Docs**). It was delivered on Thursday, January 4, 2018. The code for this deliverable is available on a private GitHub site until production. In summary, the work includes the following features:

- **UI/UX Design:** Refinement of designs of Milestone 3 tasks; preliminary mockups for Milestone 4 tasks
- **Stop Viewer:** Detailed view of specific stop including route served, scheduled departure times, and real-time arrival information if available
- **Trip Viewer:** Detailed view of a single vehicle trip with scheduled departure times and highlighting of section utilized by current itinerary
- **Route Viewer:** Listing of all available routes with map display of selected route. Also includes initial implementation of a global "application menu"
- **Extended Pedestrian Routing:** New system for weighting pedestrian routes based on OSM way characteristics using custom configuration file. Includes new API endpoint for testing and calibration.

Quarterly Progress

In addition to the completed milestone, the user interface design continues to be refined in InVision and the live demo.



Screen capture of demo version of application.

Task 4: Geocoder Development

Pelias is a non-proprietary and non-restrictive option for address locating that is an important requirement for trip planning. This task includes the implementation of a reference framework for government agencies to auto-feed their authoritative address data into a publicly accessible geocoding service.

Quarterly Deliverables

Deliverables expected for last quarter include Mapzen's Milestone 2 – Data Ingestion Pipeline, which was submitted at the end of December. Testing has not been completed at this time. Mapzen has handed over control of the Repo for this work to Open Addresses. Documentation is available here: https://github.com/openaddresses/submit-ui

A significant amount of work has been performed on the remaining two deliverables:
Milestone 3: Interactive Data Management Tools **(Appendix C - T4M3 InteractiveDataTool_Documentation).**
Milestone 6: Testing & Validation Framework (**Appendix D - T4M6 Testing_Package_Documentation**).

Quarterly Progress

The closure of Mapzen on February 1, 2018, was announced in December 2017, which resulted in an early termination of the MOD Sandbox contract with Samsung. It is expected to have no negative impact on the project. The Pelias team at Mapzen is establishing a new company to support the open source project. In addition to the Samsung/Mapzen's $64,000 in-kind contribution, Samsung has waived the remaining $200,000 awarded contact as an in-kind contribution.

Task 5: Data Improvements

Improve OpenAddresses and OpenStreetMap (OSM) in support of comprehensive trip planning and geocoding (address matching).

Quarterly Deliverables

There were no scheduled deliverables for this task during this quarter.

Quarterly Progress

Updates to OSM continue as planned.

Task 6: Integrated Payment Plan

As a partner on this project, moovel will facilitate compatibility with their planned booking and payment features so customers can plan and pay for their trips in one app.

Quarterly Deliverables

There were no scheduled deliverables for this task during this quarter.

Quarterly Progress

Moovel participated in the October meetings and events offering insight into the development of the plan. August 25th.

Meetings and Events

October 11, 2017 TriMet MOD Sandbox Grant partners update and technical breakout sessions
October 12-13, 2017 Mobility on Demand (MOD) Workshop, held in Atlanta, GA

TriMet conducts weekly project meetings on the following rotating Slack channels every Thursday at 10am PST.

- Geocoder Meetings (https://trimet-mod-sandbox.slack.com/messages/geocoding/)
- Application Development Meetings (https://trimet-mod-sandbox.slack.com/messages/general/)

Upcoming Events

February 15, 2018 USDOT & ITS America Webinar on Standards and Specifications for MOD
March 12-14, 2018 Shared Use Mobility Summit, On-Demand Services in Trip-Planning Apps
April 5-6, 2018 TechFestNW , The New Mobility Framework
April 9-11, 2018 Fare Collection/Revenue Management & TransITech Conferences, MOD Update
April 18-19, 2018 TriMet MOD Grant Workshop II, moovel PDX

Appendices

Appendix A - MOD Application Test Plan

Appendix B - Task 3 Milestone 4 (T3M4) RouteStopViewer_PedImprvmts_Docs

Appendix C - Task 4 Milestone 3 (T4M3) InteractiveDataTool_Documentation

Appendix D - Task 4 Milestone 6 (T4M6)Testing_Package_Documentation

MOD Application Test Plan

PRELIMINARY DRAFT
December 2017

Key Team Members:

Steele, Madeline; Shank, Virginia; Green Jr., Carl; Lin, Tom; Whipple, Dave; Jon Campbell; Richardson, Myleen; McHugh, Bibiana

TABLE OF CONTENTS

Preface

Introduction

Corresponding Project Evaluations and Reports

2017 Q2	Draft Evaluation Report (ppt)
2017 Q3	Equity and Accessibility Report (FTA Requirement)
2017 Q4	MOD Evaluation Logic Model - TriMet (FTA Requirement)
2018 Q1	Evaluation and Test Plan
2018 Q3	Field Demonstration/Beta Release (FTA Requirement)
2018 Q4	Knowledge Transfer and Project Report (FTA Requirement)

Schedule

2018 Q1	Prototype Release
2018 Q2	Prototype Testing (internal)
2018 Q3-4	Beta Test I & 2 (external)

Prototype Test Phase

Goal is to prepare prototype for external beta testing phase. Includes verification/validation following by planned improvements. Internal testing will include:

- System, Stress, Performance Tests (meeting industry standards)
- Preliminary assessment of usability (UI)
- Functionality tests against requirements

Beta Test Phase I & II

A consultant will be retained to assist with the beta testing. The goals of this phase is to improve the application in preparation for:

1. A broader more extensive public beta release and study, such as TNC integration value.
2. Implementation by other agencies.
3. Provide a foundation for the design and development of the TriMet specific implementation of the trip planner.

Process

The following process will be reiteratively performed twice in the last two quarters of the project:

1. Gather qualitative and quantitative data to evaluate the usability of website
2. Recommend improvements
3. Implement the recommendations
4. Re-test the site to measure the effectiveness of changes.

Data Collection Techniques

Methods for gathering the data include:

1. Focus groups for usability testing (consultant)
2. Online surveys (Ginger)
3. In-field shadowing of small group of test users (GIS surveyors)

1. **Usability Assessment**

   A small focus group will be brought in and studied to assess the following:

   - Intuitive design: a nearly effortless understanding of the architecture and navigation of the site
   - Ease of learning: how fast a user who has never seen the user interface before can accomplish basic tasks
   - Efficiency of use: How fast an experienced user can accomplish tasks
   - Memorability: after visiting the site, if a user can remember enough to use it effectively in future visits
   - Error frequency and severity: how often users make errors while using the system, how serious the errors are, and how users recover from the errors
   - Subjective satisfaction: If the user likes using the system

2. **Surveys**

   Test groups will include the elderly, disabled, minority and low income (LEP will not be included as the **application will only be tested in English**). Beta testers will be recruited from the TriMet Rider's Club. Depending on the demographic breakdown of the Rider's Club recruits, we may also recruit people from underrepresented communities of concern for the survey. TriMet has outreach lists in place for these groups, and our Title VI coordinator can assist with this. External agencies that are project stakeholders will be involved in the recruitment of beta testers in their area for online survey participation.

   Survey Questions, both online and in-field, should address the following:

   - Survey questions designed to assess the perception of utility of SUM options in OTP
   - Change in perception of usability and design of OTP interface during a two-phased beta approach, demonstrating improvements in satisfaction and feedback.
   - Change in perception of utility of real-time information presented by the updated OTP
   - Survey questions assessing response to first-mile/last-mile information in OTP

3. **In-field Shadowing**

Evaluation Objectives

The following suite of tests correspond to the project's evaluation logic model.

1. Pelias Geocoder Match Rate

Goals

Provide OTP users with more accurate matching of addresses and other points of interest (POIs) such as business names, transit stop ID's, park and ride facilities.

Data Sources

- Dataset 1: TriMet Queries
- Description: User submitted address search strings (matched or unmatched)
- Origin: Captured by our trip planner (11m addresses) , only TriMet service district area, good geographic sampling of addresses
- Count: ?
- Baseline: April 2017

Methods of Evaluation

Geocoder Comparisons:SOLR; Metro RLIS (API); Google Maps; Pelias; ESRI ArcMap Geocoder; Mapbox; Nominatim

2. *Pelias Geocoder Accuracy of Point Locations*

Goals

Provide OTP users with more accurate matching of addresses and other points of interest (POIs) such as business names, transit stop ID's, park and ride facilities.

Data Sources

Dataset 2: FourSquare Addresses
Description: Verified address point locations
Origin: FourSquare
Count: 4,067 foursquare-validated addresses in TriMet's 3 county AOI (OR only)
Baseline: April 2017

Methods of Evaluation

Geocoder Comparisons:SOLR; Metro RLIS (API); Google Maps; Pelias; ESRI ArcMap Geocoder; Mapbox; Nominatim

3. *OTP Time & Cost Comparisons*

Goals

OTP SUM will encourage travelers to use public transit for trips that previously faced first mile or last mile challenges

Data Sources

Dataset 3: O&D
Description: Identify Set of Planned Trips by Origin and Destination
Origin: TriMet
Count:
Baseline: Q2 Prototype texting

Methods of Evaluation

Transit only trip time comparisons:

- Current OTP, Google Transit, Apple Maps
- Bike to Transit - Current OTP
- TNCs to Transit - New OTP

4. *OTP Accuracy/Validity of Planned Trips*

Goals

Ensure trip planning results are accurate.

Data Sources

Dataset 3: O&D
Description: Identify Set of Planned Trips by Origin and Destination
Origin: TriMet
Count:
Baseline: Prototype Release of new OTP

Methods of Evaluation

- Survey of OTP test group and Call Taker Reports

5. *OTP Enhanced Pedestrian Accessibility*

Goals

Enhance OTP's pedestrian routing logic to take advantage of newly added sidewalk tags as well as other attributes of OpenStreetMap that reflect safety and pleasantness for pedestrians to improve pedestrian trip plans.

Data Sources

Dataset 4: O&D Pairs
Description: Origin/destination pairs where there are several walking trip options of comparable length but with differing sidewalk coverage and stress levels will be hand selected, and then run in the new and old versions of OTP.
Origin: TriMet
Count:
Baseline: Prototype Release of new OTP

Methods of Evaluation

- T-test on count of binomial variables

6. *OTP Increase of Feasible Itineraries*

Goals

Produce results for trips that currently do not produce results due to transit service.

Data Sources

Dataset 5: Trips not possible
Description: Trips that currently depart on different dates and times than requested
Origin: TriMet customer complaints
Count:
Baseline: Prototype Release of new OTP

Methods of Evaluation

- Multimodal trips comparisons against current OTP and new OTP

7. *OpenStreetMap Sidewalk Data*

Goals

Data improvements to OSM were made to support enhanced pedestrian accessibility information.Sidewalk presence/absence data entered for all streets in the TriMet trip planner

Data Sources

Dataset 6: Street Segments
Description: Random sample of street segments in the region
Origin: OSM
Count: 300 to 400
Baseline: Prototype Release of new OTP

Methods of Evaluation

- Random sample of 300 to 400 street segments in the region validated with current 6" aerial imagery flown in June 2017. 100 random samples within that data set will be verified by ground truth.

8. Replicability of Application

Goals

Test the feasibility of implementing the Pelias geocoder and and OTP.

opentripplanner / **otp-react-redux**

Branch: **m4-enhancements**

Commits on Jan 9, 2018


feat(example): Get example working w/ latest changes ...
demory committed 8 days ago ✗
73d9952

Commits on Jan 8, 2018


fix(reducer): Fix outdated import, clean up misc formatting issues in... ...
demory committed 9 days ago
c5305f3


feat(realtime): Add filter for display of realtime effects alert
demory committed 9 days ago
5ad38eb

Commits on Jan 5, 2018


feat(realtime): Restore realtime route-modification alerts in mobile ... ...
demory committed 12 days ago
431609b


feat(mobile): Return mobile user to search form if from/to is changed... ...
demory committed 12 days ago
73420f3


fix(viewers): Update layout of stop/trip viewers to fix scrolling bug
demory committed 12 days ago
ed1e83a


feat(viewers): Add initial implementation of Route Viewer ...
demory committed 12 days ago
8a1b32f

Commits on Dec 28, 2017


feat(viewers): More fully implement stop & trip viewers
demory committed 20 days ago
71ae99b

Commits on Dec 14, 2017


refactor(mobile): Rename 'mobile' reducer to 'ui'
demory committed on Dec 14, 2017
1e9cde3


feat(mobile): Refactor responsive app functionality from trimet-mod-otp
demory committed on Dec 14, 2017 ✗
623a6a6

Commits on Dec 13, 2017


Merge pull request #40 from opentripplanner/m3-demo ...
demory committed on Dec 13, 2017 ✗
Verified
c235a82

Commits on Dec 12, 2017


refactor(api): add back code to handle server errors
evansiroky committed on Dec 12, 2017 ✓
aa504c3


refactor(map): Use buttons instead of <a> links to appease linter
demory committed on Dec 12, 2017 ✗
4fb6226


build(transitive): Add transitive.js dependency
demory committed on Dec 12, 2017 ✗
9575c3c


build(build): Update React versions in package.json
demory committed on Dec 12, 2017 ✗
9e60c6e

feat(map): Make route overlay url configurable
demory committed on Dec 12, 2017 — 6a64eb4

style(general): Fix linter issues
demory committed on Dec 12, 2017 — de127d1

Commits on Dec 11, 2017

style(general): Fix linter issues
demory committed on Dec 11, 2017 ✕ — a70e499

refactor(map): Remove unused file d3l.js
demory committed on Dec 11, 2017 — 576c8c7

Commits on Dec 7, 2017

build(yarn): Update yarn.lock
demory committed on Dec 7, 2017 ✕ — 0ab2c78

Merge branch 'map-fixes-landon' into m3-demo
demory committed on Dec 7, 2017 — b1d0017

Commits on Nov 8, 2017

fix(narrative): silence react warnings
landonreed committed on Nov 8, 2017 ✕ — 9a4e1ed

feat(map): use right-click to set origin/dest ...
landonreed committed on Nov 8, 2017 — 11c176d

feat(geocoding): use custom pelias instance and focusPoint ...
landonreed committed on Nov 8, 2017 ✕ — d4a2c2b

fix(geolocation): geolocate on page load only for mobile ...
landonreed committed on Nov 8, 2017 — 333ded5

chore(yarn): update react-select-geocoder/iso-mapzen-search
landonreed committed on Nov 8, 2017 — b8309a6

Commits on Oct 26, 2017

feat(map): use right click (long press on mobile) to set location ...
landonreed committed on Oct 26, 2017 ✕ — efb59d8

Commits on Sep 29, 2017

fix(map): Fix issues related to map auto-fitting to results
demory committed on Sep 29, 2017 — cb61e89

fix(Fix various map-resizing related issues):
demory committed on Sep 29, 2017 — 2c8eaec

Commits on Sep 28, 2017

fix(overlay): Fix conflict between stops-layer rendering and transiti... ...
demory committed on Sep 28, 2017 — 77deab9

fix(map): Make rt-annotate toggling work w/ transtive map
demory committed on Sep 28, 2017 — b660141

Merge branch 'rt-annotate' into m3-demo
demory committed on Sep 28, 2017 — 42b4ee8

Merge branch 'transitive' into m3-demo
demory committed on Sep 28, 2017 — 4cf494b

feat(overlay): Make stops overlay an empty FeatureGroup when not rend... ...

demory committed on Sep 28, 2017

5311246

feat(map): Make controlled overlays visible by default

demory committed on Sep 28, 2017

137f095

Newer Older

openaddresses / **submit-ui**

First time contributing to openaddresses/submit-ui? Dismiss

If you know how to fix an issue, consider opening a pull request for it.
Learn more about pull requests

Filters | is:pr is:closed | Labels | Milestones | New pull request

Clear current search query, filters, and sorts

1 Open ✓ 74 Closed | Author | Labels | Projects | Milestones | Reviews | Assignee | Sort

- 'See more' text to data format page ✓ — 9
  #174 by hanbyul-here was merged 12 days ago • Approved
- deleting ds_store ✓
  #173 by hanbyul-here was merged 14 days ago • Review required
- Error Messages ✓
  #169 by mkong0216 was merged 19 days ago • Approved
- Help Modal ✓ — 1
  #168 by mkong0216 was merged 14 days ago • Approved
- Added "see more" text to data-format page ✓ — 1
  #167 by mkong0216 was closed 14 days ago • Review required
- Progress tabs styling ✓ — 1
  #166 by mkong0216 was merged 19 days ago • Approved
- Styling on More-Info page ✓ — 14
  #165 by mkong0216 was merged 14 days ago • Approved
- Miscellaneous style changes ✓
  #164 by mkong0216 was merged 19 days ago • Approved
- Help Needed Model ✓
  #161 by mkong0216 was merged 27 days ago • Approved
- Successful submission page ✓
  #159 by meghanhade was merged 27 days ago • Approved
- Unit split functions ✓ in review
  #158 by meghanhade was merged 27 days ago • Approved
- change button name ✓
  #157 by meghanhade was merged on Dec 18, 2017 • Approved
- more-info page validation ✓ — 2
  #156 by hanbyul-here was merged 27 days ago • Approved
- add error message if initial column isn't selected ✓ — 7
  #155 by meghanhade was merged 27 days ago • Approved
- fix formatting error in remove-prefix-postfix.js ✓ — 5
  #154 by meghanhade was merged 19 days ago • Approved
- Copy ✓ — 3
  #151 by meghanhade was closed on Dec 18, 2017 • Review required
- added ds_store to gitinore ✓ — 1
  #144 by hanbyul-here was merged on Dec 13, 2017 • Approved
- Unload data format on cancel ✓

#143 by meghanhade was merged on Dec 14, 2017 • Approved

remove mapzen search info ✓
#141 by meghanhade was merged on Dec 13, 2017 • Approved

Intro page validation ✓ 9
#139 by hanbyul-here was merged on Dec 13, 2017 • Approved

Remove prefix and postfix ✓ 2
#137 by meghanhade was merged on Dec 15, 2017 • Approved

[WIP] miscellaneous style changes ✓ 5
#136 by mkong0216 was closed 26 days ago • Review required

data format fixes ✓
#135 by mkong0216 was merged on Dec 8, 2017 • Approved

added changeset to 'previous' nav button ✓ 1
#134 by hanbyul-here was closed on Dec 17, 2017 • Review required

design tweaks using openaddress colors ✓ 1
#133 by mkong0216 was merged on Dec 7, 2017 • Approved

Previous 1 2 3 Next

ProTip! Notify someone on an issue with a mention, like: @jon-campbell-ibigroup.

openaddresses / **submit-ui**

First time contributing to openaddresses/submit-ui? Dismiss

If you know how to fix an issue, consider opening a pull request for it.
Learn more about pull requests

Filters | is:pr is:closed | Labels | Milestones | New pull request

Clear current search query, filters, and sorts

1 Open ✓ **74 Closed** Author Labels Projects Milestones Reviews Assignee Sort

- **add ui to clear column** ✓ — 2
  #131 by meghanhade was merged on Dec 6, 2017 • Approved
- **Join dropdown options** ✓
  #130 by meghanhade was merged on Dec 6, 2017 • Approved
- **[WIP] Design Tweaks** ✓
  #129 by mkong0216 was closed on Dec 6, 2017 • Review required
- **[WIP] Mobile warning** ✓ — 3
  #128 by mkong0216 was merged on Dec 5, 2017 • Approved
- **Data format functions** ✓ — 4
  #127 by meghanhade was merged on Dec 5, 2017 • Approved
- **minor fix** ✓ — 1
  #124 by mkong0216 was merged on Nov 30, 2017 • Approved
- **use house number instead of number, just in templating** ✓
  #123 by meghanhade was merged on Nov 29, 2017 • Approved
- **add licenses from Diana's list** ✓
  #122 by meghanhade was merged on Nov 29, 2017 • Approved
- **always show attribution question. remove source question.** ✓
  #121 by meghanhade was merged on Nov 29, 2017 • Approved
- **move lon and lat to beginning of table** ✓
  #120 by meghanhade was merged on Nov 30, 2017 • Approved
- **Componentize function buttons** ✓
  #116 by meghanhade was merged on Nov 29, 2017 • Approved
- **Join multiple columns two** ✓ — 4
  #107 by meghanhade was merged on Nov 22, 2017 • Approved
- **Save Field Values** ✓ — 1
  #105 by mkong0216 was merged on Nov 21, 2017 • Approved
- **Review page reformat** ✓
  #104 by mkong0216 was merged on Nov 21, 2017 • Approved
- **UI changes** ✓ — 1
  #101 by mkong0216 was merged on Nov 15, 2017 • Approved
- **Updated next button's route** ✓
  #98 by mkong0216 was merged on Nov 8, 2017 • Approved
- **Review page segments** ✓ — 4
  #96 by meghanhade was merged on Nov 10, 2017 • Approved
- **License source** ✓ — 3

#95 by meghanhade was merged on Nov 10, 2017 • Approved

Adding Feed Progress Tabs ✓ 5
#93 by mkong0216 was merged on Nov 11, 2017 • Approved

Remove data page and related things ✓ in review 2
#88 by meghanhade was merged on Nov 3, 2017 • Approved

add text properties to nav-button component ✓
#86 by meghanhade was merged on Nov 3, 2017 • Approved

add cancel button ✓ 4
#83 by meghanhade was merged on Nov 2, 2017 • Approved

Intro update ✓ in review 4
#82 by meghanhade was merged on Nov 3, 2017 • Approved

Mapping preview ✓ 3
#67 by meghanhade was merged on Nov 2, 2017 • Approved

fix split-function bug ✓
#65 by meghanhade was merged on Oct 26, 2017 • Approved

Previous 1 2 3 Next

ProTip! Exclude your own issues with -author:jon-campbell-ibigroup.

openaddresses / **submit-ui**

First time contributing to openaddresses/submit-ui? Dismiss

If you know how to fix an issue, consider opening a pull request for it.
Learn more about pull requests

Filters | is:pr is:closed | Labels | Milestones | New pull request

Clear current search query, filters, and sorts

1 Open ✓ 74 Closed | Author | Labels | Projects | Milestones | Reviews | Assignee | Sort

- **Mapzen search** ✓ — 12
  #60 by meghanhade was merged on Oct 25, 2017 • Approved
- **Mapping step through** ✓
  #58 by meghanhade was merged on Oct 23, 2017 • Approved
- **Edit mode** ✓ — 4
  #57 by meghanhade was merged on Oct 9, 2017 • Approved
- **Column mapping** ✓ — 2
  #54 by meghanhade was merged on Sep 27, 2017 • Approved
- **add country and region info to submission preview** ✓
  #53 by meghanhade was merged on Sep 27, 2017 • Approved
- **add eslint rule for no-tabs, fix errors** ✓ — 2
  #52 by meghanhade was merged on Sep 28, 2017 • Approved
- **Added Machine backend issues for job API** ● — 1
  #49 by migurski was merged on Sep 26, 2017 • Review required
- **Install mirage** ✓ — 4
  #48 by meghanhade was merged on Sep 26, 2017 • Approved
- **fix merge conflict** ✓ — 2
  #41 by meghanhade was merged on Sep 18, 2017 • Approved
- **add country drop-down** ✓ — 3
  #40 by meghanhade was merged on Sep 15, 2017 • Approved
- **Add backend mock for testing** ✗ — 1
  #38 by dianashk was closed on Nov 1, 2017 • Review required
- **set up form validation** ✓ in review — 4
  #37 by hanbyul-here was merged on Sep 15, 2017 • Approved
- **README update** ✓
  #36 by meghanhade was merged on Sep 8, 2017 • Approved
- **Data input** ✓ — 4
  #35 by meghanhade was merged on Sep 6, 2017 • Approved
- **Create docs directory#14** ✓ — 3
  #17 by meghanhade was merged on Aug 22, 2017 • Approved
- **update circle.yml to use master for staging** ✓
  #12 by meghanhade was merged on Aug 10, 2017 • Approved
- **[do not merge this] testing netlify github notification** ✓ — 3
  #10 by hanbyul-here was merged on Aug 9, 2017 • Review required
- **[do not merge this] testing netlify hook** ✓

#9 by hanbyul-here was closed on Aug 9, 2017 • Review required

Build clean app ✓ 6
#8 by meghanhade was merged on Aug 9, 2017 • Approved

Merge initial app structure to master ✗ 3
#6 by meghanhade was closed on Aug 9, 2017 • Review required

Implement semantic ui
#5 by meghanhade was merged on Jul 28, 2017

Test semantic
#4 by meghanhade was closed on Jul 28, 2017

added gitignore
#3 by hanbyul-here was merged on Jul 28, 2017

User Flow diagram ● 12
#2 by dianashk was closed on Sep 11, 2017 • Review required

Previous 1 2 3 Next

ProTip! Exclude your own issues with -author:jon-campbell-ibigroup.

pelias / **fuzzy-tester**

A fuzzy testing library for geocoding http://mapzen.com/pelias

547 commits **14** branches **19** releases **11** contributors

Branch: **master** New pull request Create new file Upload files Find file Clone or download

trescube Merge pull request #124 from pelias/disable-package-lock ... Latest commit `1c08e8c` on Oct 27, 2017

bin	Rename most variables named 'result'	2 years ago
data	Smaller example data file is enough	2 years ago
lib	Filter completely passing test suites in email	9 months ago
output_generators	Always use red for regression count color	4 months ago
scripts	Use lodash extend instead of extend package in scripts	8 months ago
test	Adding newline characters at the end of the files	2 years ago
.gitignore	Move json failure files to output_generators	3 years ago
.jshintignore	Initial commit	3 years ago
.jshintrc	add generic csv output generator	10 months ago
.npmrc	disable package-lock in .npmrc	3 months ago
.travis.yml	update npm-check to ignore updates	3 months ago
README.md	docs(readme): add Greenkeeper badge	a year ago
autocomplete_example_output.png	Add example of autocomplete output	2 years ago
changelog.md	Update changelog	2 years ago
package.json	chore(package): update dependencies	3 months ago

README.md

Pelias Fuzzy Tester

Greenkeeper enabled

This is the Pelias fuzzy tester library, used for running our acceptance-tests and fuzzy-tests.

What are fuzzy tests? See the original problem statement that lead to the creation of this library.

Most importantly, fuzzy tests deliver more than just a single bit of pass or fail for each test: they specify a total number of points (a score) for the test, and return how many points out of the maximum were achieved. The weighting of individual parts of the test can be adjusted.

Note: fuzzy-tester requires NPM version 2 or greater. The NPM team recommends you update NPM using NPM itself with `sudo npm install -g npm`.

Usage

```
// in the root directory of the repo containing the tests
fuzzy-tester
fuzzy-tester --help
```

```
fuzzy-tester -e prod
fuzzy-tester -t dev
```

Test Case Files

Test-cases are expected to live in `test_cases/`, and are split into test *suites* in individual JSON files. Each file must contain the following properties:

- `name` is the suite title displayed when executing.
- `priorityThresh` indicates the expected result must be found within the top N locations. This can be set for the entire suite as well as overwritten in individual test cases.
- `distanceThresh` (optional) defines the accepted maximal distance (in meters) between search result coordinates and the coordinates defined in each test. Each test case can include a specific threshold value. This makes sense because location of a neigborhood is not as accurately defined as location of, say, a building. Default threshold is 500 meters.
- `tests` is an array of test cases that make up the suite.
- `endpoint` the API endpoint (`search`, `reverse`, `suggest`) to target. Defaults to `search`.
- `weights` (optional) test suite wide weighting for scores of the individual expectations. See the weights section below

`tests` consists of objects with the following properties:

- `id` is a unique identifier within the test suite (this could be unnecessary, let's discuss)
- `type` is simply a category to group the test under, to allowing running select groups of tests rather than all of them.
- `status` is the optional expected status of this test-case (whether it should pass/fail/etc.), and will be used to identify improvements and regressions. May be either of `pass` or `fail`.
- `user` is the name of the person that added the test case.
- `endpoint` the API endpoint (`search`, `reverse`, `suggest`) to target. Defaults to `search`, which will override the `endpoint` specified by the test-suite.
- `in` contains the API parameters that will be urlencoded and appended to the API url.
- `expected` contains *expected* results. The object can contain a `priorityThresh` property, which will override the `priorityThresh` specified by the test-suite, and must contain a `properties` property. `properties` is mapped to an array of either of:
  - `object`: all of the key-value pairs will be tested against the objects returned by the API for exact matches.
  - `string`: a matching object will be looked up in the `locations.json` file. Allows you to easily reuse the same object for multiple test-cases.

  If `properties` is `null`, the test-case is assumed to be a placeholder.

  `expected` can also contain a test specific `distanceThresh` value, and an array of `[lon, lat]` coordinates. With these coordinates, it is possible to compare distance between locations found in the search and expected locations. This is often useful, because matching the name labels may fail even when the geocoder has found a proper result ('Harvard' != 'Harvard University'). Location coordinates are less ambiguous.

  Coordinate based tests also help to track invalid location data in the search database.
- `unexpected` is analogous to `expected`, except that you *cannot* specify a `priorityThresh` and the `properties` array does *not* support strings.
- `weights` (optional) test case specific weighting for scores of the individual expectations. See the weights section below

Import Scripts for Test Cases

The `scripts` folder contains example scripts for creating fuzzy tests. For example, the data import script `scripts/importHSLpoi.js` can be used to create a fuzzy test from a poi data list as follows:

- Edit the import script `scripts/importHSLPoi.js` to specify which poi attributes and search attributes will be compared in the test. The current defaults serve as a good starting point.
- Run the command `node scripts/importHSLPoi.js data/poi.txt`, where poi.txt is the source data file.
- The script creates a test file called `HslPoitest.json`. You may edit it to fine tune the test setup. For example, you can change the threshold values afterwards, or add subtest specific thresholds.
- Move the test file to the testing environment `../fuzzy-tests/test_cases` and run the test there. For more information, check fuzzy-tests.

Output Generators

The acceptance-tests support multiple different output generators, like an email and terminal output. See `node test --help` for details on how to specify a generator besides the default. Note that the `email` generator requires an AWS account, and that your `pelias-config` file contain the following configuration:

```
{
        "acceptance-tests": {
                "email": {
                        "ses": {
                                "accessKeyId": "AWSACCESSKEY",
                                "secretAccessKey": "AWS/Secret/key",
                        },
                        "recipients": ["recipient1@domain.com", "recipient2@domain.com"], // the list of recipients
                }
        }
}
```

Autocomplete mode

A special output generator, `-o autocomplete` not only changes the output, but changes the behaviour of the test suite. Instead of running each test case exactly as defined, it will run many tests for each test case. The tests will be run against the autocomplete endpoint and will correspond to successively longer substrings of the input text, similar to how a user would type the text into autocomplete. It looks like this:

The results are shown underneath the input text, with each character corresponding to the result of the autocomplete query with the input text up to the character above entered. Tests that pass are green, tests that fail are red. If the expected output was not found at all, the result character will be an `F`, if the expected output was found, the character will be the zero indexed location in the API results where it was found.

To the right of the input text, some additional info might be displayed. The first is any additional parameters being sent with the API call, like a location bias. The second is a count of the number of expectations included in the test case. This helps detect situations where one expectation is found, but the other isn't (the result might be a confusing red `0` in that case).

API URL aliases

The acceptance-tests runner recognizes a number of aliases for Pelias API URLs (eg, `stage` corresponds to `pelias.stage.mapzen.com`), which can be specified as command-line arguments when running a test suite. You can override the default aliases and define your own in `pelias-config`:

```
{
        "acceptance-tests": {
                "endpoints": {
                        "alias": "http://my.pelias.instance"
                }
```

```
        }
}
```

Weights

Weights are used to influence the score each individual expectation contributes to the total score for a test. By default, all fields in expected properties, passing the priority threshold, and the absence of any unexpected properties each contribute one point.

Any score for any individual property can be changed by specifying an object `weights` in a test suite, or in an individual test case. For example, to more heavily weight the `name` property by giving it a weight of 10 points, set weights to the following:

```
{
  "properties": {
    "name": 10
  }
}
```

Weights can be nested and are completely optional, in which case the defaults will be in effect.