Google™

Google Realtime Transit

Implementation

CONTENTS

Overview

Implementation

Next Steps

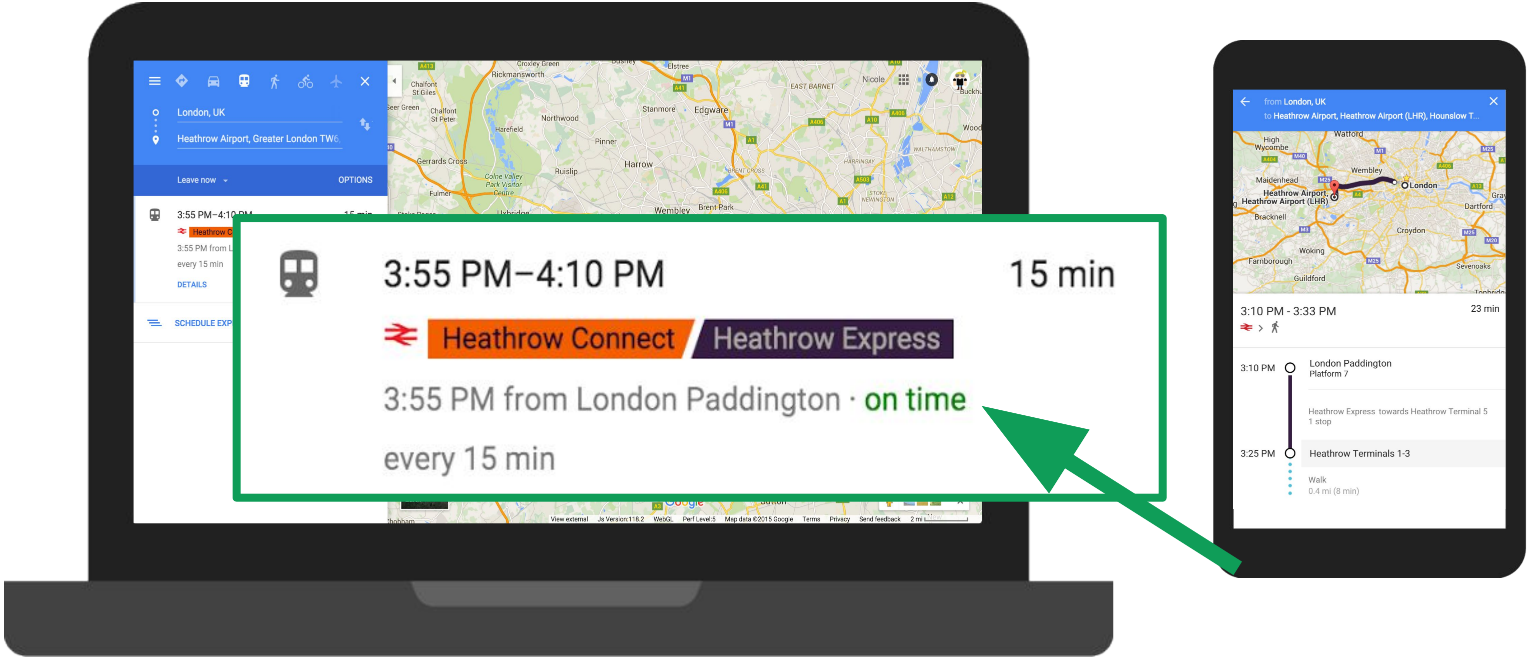
Overview

In brief:

Keep your riders up-to-date about delay and service alerts *in real time*.

Google Transit helps you connect riders with your transit information across desktops and mobile devices, in easy-to-use and informative ways. Adding real time data further improves their experience through updates as they happen.

Realtime data helps your riders plan their trips...



London, UK

Heathrow Airport, Greater London TW6,

Leave now

OPTIONS

DETAILS

SCHEDULE EXP

3:55 PM–4:10 PM 15 min

Heathrow Connect / Heathrow Express

3:55 PM from London Paddington · on time

every 15 min

from London, UK

to Heathrow Airport, Heathrow Airport (LHR), Hounslow T...

3:10 PM - 3:33 PM 23 min

3:10 PM London Paddington
Platform 7

Heathrow Express towards Heathrow Terminal 5
1 stop

3:25 PM Heathrow Terminals 1-3

Walk
0.4 mi (8 min)

And is an important tool in improving rider satisfaction

Excellent user experience

Riders get realtime updates for easy trip planning

Potential for increased ridership

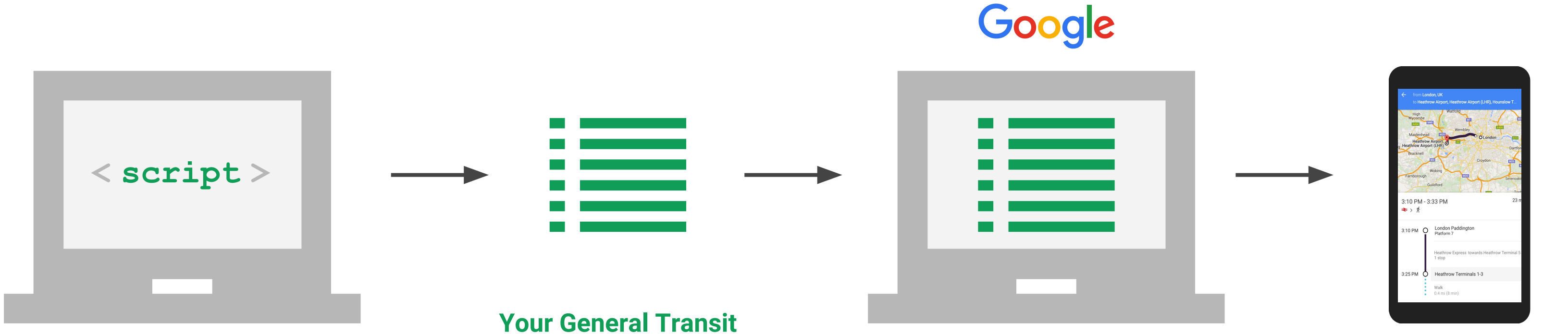
A smoother experience helps encourage repeat users

Furthering the goals of your agency

You deepen rider engagement to meet your own transit goals

Read about the benefits of real time data for riders.

How Realtime Transit works



Your Transit Database

Your General Transit Feed Spec (GTFS feed)

Google Transit Database (w/GTFS feed)

Your Riders' Maps

What you do:

Write a script that "translates" your info into a realtime feed

What you get:

Google-friendly data (a realtime feed in ProtoBuff format)

What happens:

Google ingests updates, cross-references against GTFS info, and shows real-time updates to your Google Maps riders

Implementation

Before you get started!

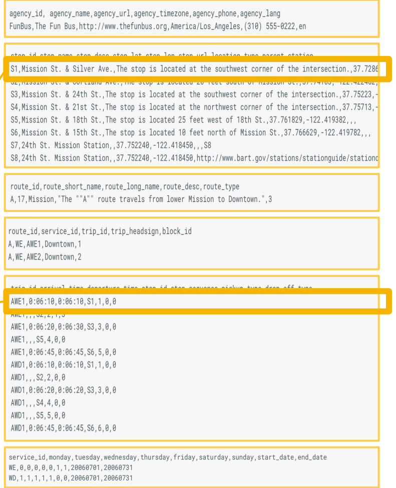
You must be submitting a static Google Transit Feed Spec (GTFS) feed that provides key information about your routes, stops, schedules, and more.

Real-time updates reference information in your static feed.

Service alert:
Stop “S1” is offline due to road detour.

Trip update:
Trip “AWE1” is 12 minutes behind schedule.

Realtime feed



```
agency_id, agency_name,agency_url,agency_timezone,agency_phone,agency_lang
FunBus,The Fun Bus,http://www.thefunbus.org,America/Los_Angeles,(310) 555-0222,en
```

```
stop_id,stop_name,stop_desc,stop_lat,stop_lon,stop_url,location_type,parent_station
S1,Mission St. & Silver Ave.,The stop is located at the southwest corner of the intersection.,37.7286
S2,Mission St. & Cortland Ave.,The stop is located 20 feet south of Mission St.,37.74103,-122.422482,
S3,Mission St. & 24th St.,The stop is located at the southwest corner of the intersection.,37.75223,-
S4,Mission St. & 21st St.,The stop is located at the northwest corner of the intersection.,37.75713,-
S5,Mission St. & 18th St.,The stop is located 25 feet west of 18th St.,37.761829,-122.419382,,,
S6,Mission St. & 15th St.,The stop is located 10 feet north of Mission St.,37.766629,-122.419782,,,
S7,24th St. Mission Station,,37.752240,-122.418450,,,S8
S8,24th St. Mission Station,,37.752240,-122.418450,http://www.bart.gov/stations/stationguide/stationc
```

```
route_id,route_short_name,route_long_name,route_desc,route_type
A,17,Mission,"The ""A"" route travels from lower Mission to Downtown.",3
```

```
route_id,service_id,trip_id,trip_headsign,block_id
A,WE,AWE1,Downtown,1
A,WE,AWE2,Downtown,2
```

```
trip_id,arrival_time,departure_time,stop_id,stop_sequence,pickup_type,drop_off_type
AWE1,0:06:10,0:06:10,S1,1,0,0
AWE1,,,S2,2,1,3
AWE1,0:06:20,0:06:30,S3,3,0,0
AWE1,,,S5,4,0,0
AWE1,0:06:45,0:06:45,S6,5,0,0
AWD1,0:06:10,0:06:10,S1,1,0,0
AWD1,,,S2,2,0,0
AWD1,0:06:20,0:06:20,S3,3,0,0
AWD1,,,S4,4,0,0
AWD1,,,S5,5,0,0
AWD1,0:06:45,0:06:45,S6,6,0,0
```

```
service_id,monday,tuesday,wednesday,thursday,friday,saturday,sunday,start_date,end_date
WE,0,0,0,0,0,1,1,20060701,20060731
WD,1,1,1,1,1,0,0,20060701,20060731
```

Static feed

***Ask about the *Static Transit Onboarding Guide*!**

Realtime Transit implementation steps

Step 1

Submit your interest.
Use the online form to let us know you’d like to send real-time Transit updates.

Step 2

Decide what realtime data you’ll submit.
Determine which feeds you have the ability to submit.

Step 3

Create your data feed.
Prepare a feed following the General Transit Feed Specification (GTFS) realtime.

Step 4

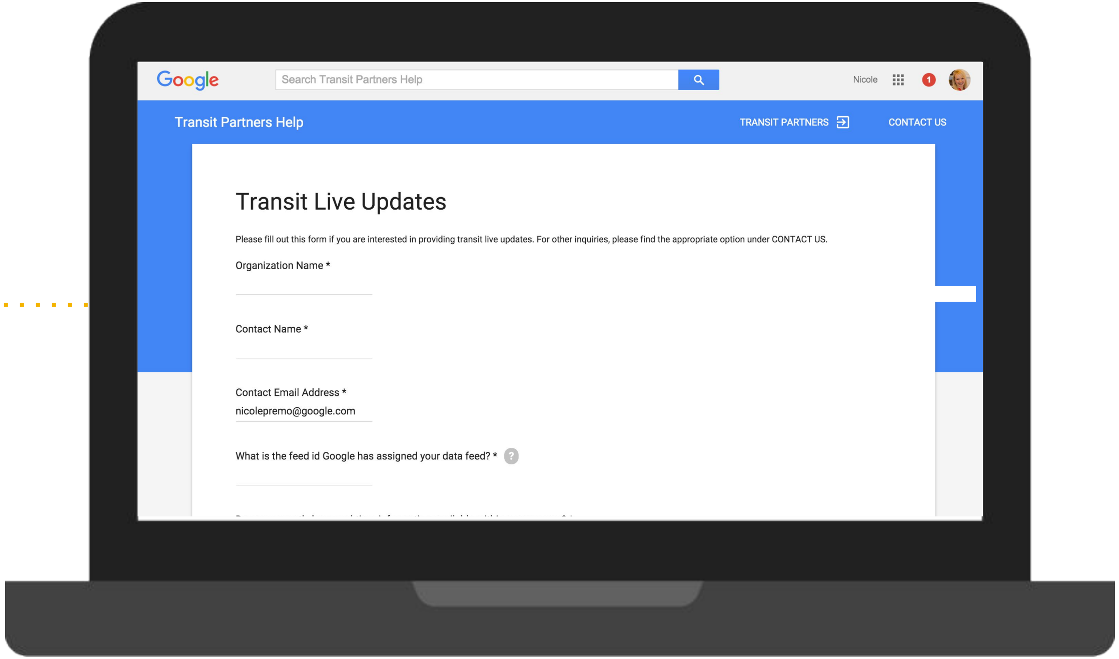
Work with Google to provide your data feed.
Choose to either publish your feed to the web or push it directly to Google.

Step 5

Validate your feed data.
Work with your Google team to provide high quality, accurate data.

Step 1: Submit your interest

Submit the interest form.
Google will verify eligibility
and be in touch with next steps.



Transit Live Updates

Please fill out this form if you are interested in providing transit live updates. For other inquiries, please find the appropriate option under CONTACT US.

Organization Name *

Contact Name *

Contact Email Address *
nicolepremo@google.com

What is the feed id Google has assigned your data feed? * ?

Step 2: Decide what realtime info you'll submit

Submitting all 3 feed types provides the most robust rider experience.

Trip update

Bus X delayed 8 minutes.

Flag timetable fluctuations and provide new arrival / departure predictions.

Service alerts

Station Y closed for construction.

Describe higher-level interruptions and optionally provide a URL for more info.

Vehicle positions*

Train Z is [location].

Pinpoint where a vehicle is - e.g. latitude and longitude.

Learn more about Realtime Transit feed files.

*Note: information from the *Vehicle positions feed* is not currently surfaced.

Deep dive: Trip updates feed

Each *StopTimeUpdate* references 1 *StopSequence* to provide changes to the standard arrival / departure timetable - these changes are provided through *StopTimeEvent* info.

```
stop_time_update {
   stop_sequence: 4
   arrival {
      time: 1450758839
    }
   departure {
      time: 1450758856
   }
}
```

EXAMPLE

Agency provides 3 *StopTimeUpdates* for the same trip instance:

StopTimeUpdate	*StopTimeEvent*	*StopSequence*
300 sec delay at stops 1-3	300	1
60 sec delay at stops 4-7	60	4
Unknown delay at stops 8-10	unknown	8

Learn more about trip updates feeds.

Deep dive: Service alerts feed

Trigger a warning informing riders of service alerts.

```
entity {
    id: "1"
    alert {
        active_period {
            start: 1451041200
            end: 1451127599
        }
     informed_entity {
         agency_id: "1"
     }
    cause: HOLIDAY
    effect: NO_SERVICE
    description_text {
        translation {
        text: "No service on Christmas Day: "en"
    }
  }
}
```

active_period - start and end for alert

Informed_entity - parts of network affected by alert

cause - reason for the alert from pre-set value options

effect - impact of the alert from pre-set value options

description_text - details about the alert

Learn more about service alerts feeds.

Deep dive: Vehicle positions feed

Use a vehicle’s onboard tracking - e.g. GPS - to provide location information. *Note that Google accepts but does not currently surface this info; providing it may make you eligible for new user interface experiments.*

```
entity {
   id: "1450809276_1402"
   vehicle {
      trip {
         trip_id: "109475583"
         schedule_relationship: SCHEDULED
         route_id: "12"
         direction_id: 0
      }
     position {
        latitude: 39.824818
        longitude: -104.97787
        bearing: 359
     }
     current_status: IN_TRANSIT_TO
     timestamp: 1450809260
     stop_id: "17193"
     vehicle {
         id: "1402"
         label: "1402"
      }
   }
}
```

vehicle - vehicle’s route (e.g. *TripID*)

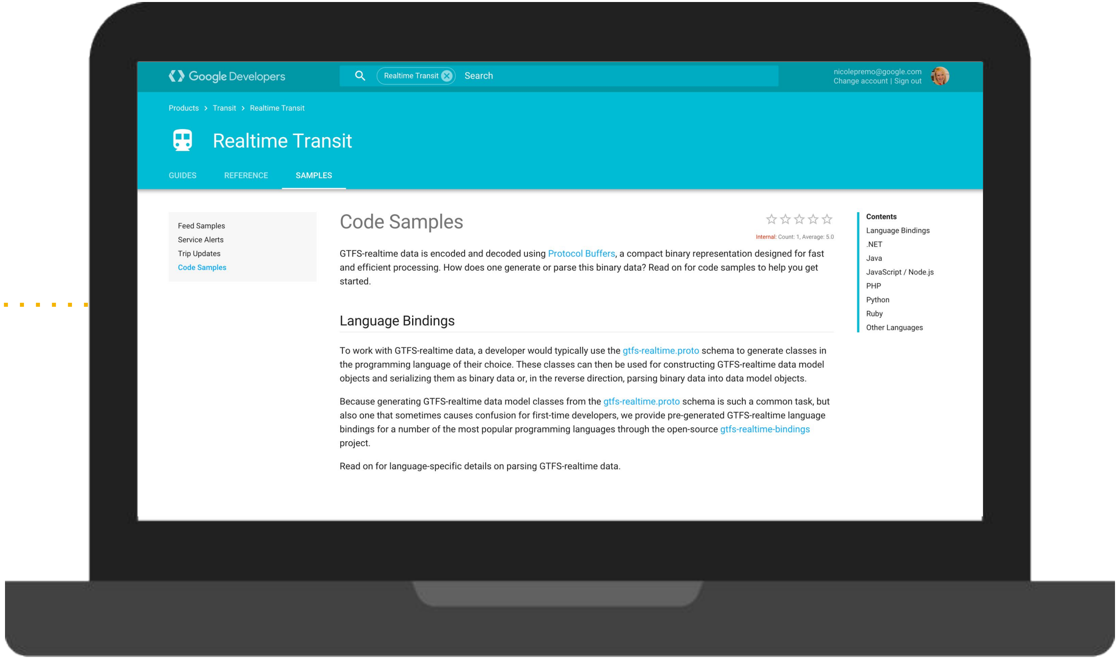
position - latitude and longitude

current_status - incoming at / stopped at /in transit to

Learn more about vehicle positions feeds.

Tip! Use the pre-compiled GTFS-realtime bindings

Save time by using pre-compiled language bindings for your programming language.



Google Developers
Realtime Transit
Search
nicolepremo@google.com
Change account | Sign out
Products > Transit > Realtime Transit

Realtime Transit

Code Samples

GTFS-realtime data is encoded and decoded using Protocol Buffers, a compact binary representation designed for fast and efficient processing. How does one generate or parse this binary data? Read on for code samples to help you get started.

Language Bindings

To work with GTFS-realtime data, a developer would typically use the gtfs-realtime.proto schema to generate classes in the programming language of their choice. These classes can then be used for constructing GTFS-realtime data model objects and serializing them as binary data or, in the reverse direction, parsing binary data into data model objects.

Because generating GTFS-realtime data model classes from the gtfs-realtime.proto schema is such a common task, but also one that sometimes causes confusion for first-time developers, we provide pre-generated GTFS-realtime language bindings for a number of the most popular programming languages through the open-source gtfs-realtime-bindings project.

Read on for language-specific details on parsing GTFS-realtime data.

Step 4: Work with Google to provide your data feed

Google fetch

You host file and provide URL - Google pulls data approx. every 30 seconds

Partner push

You submit the feed to Google servers when data changes using HTTP POST request

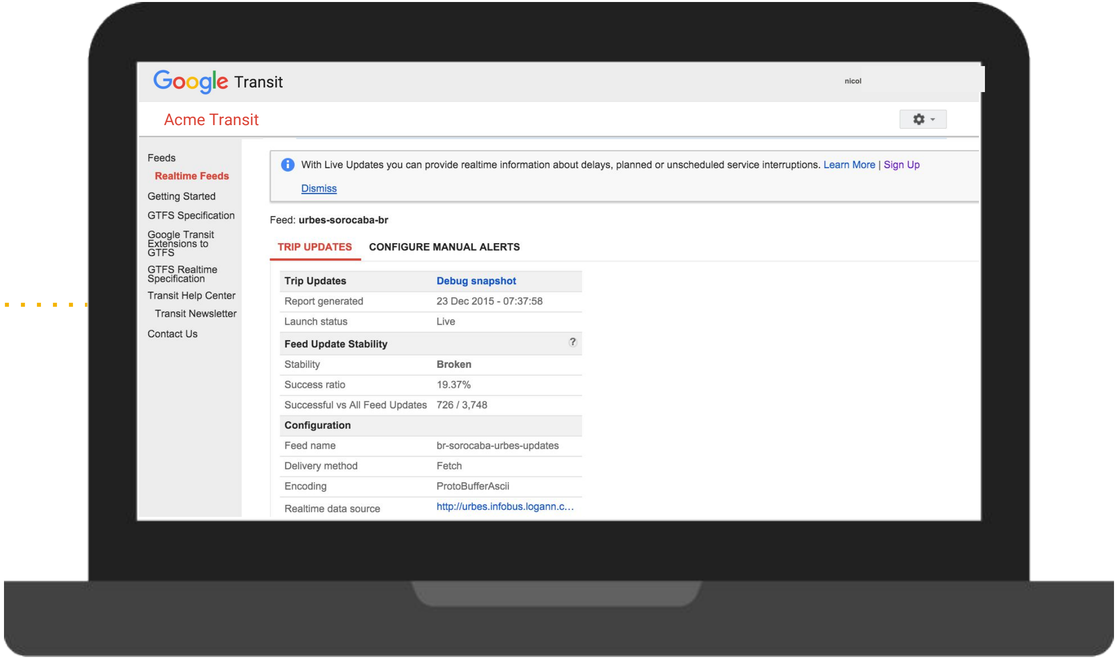
GOOD TO KNOW:

- With fetch, must provide updates / alerts within 90 seconds of receiving
- Data matches to static GTFS feed which contain route and schedule info
- Updates immediately visible in Google Maps

Learn more about publishing options.

Tip! Use your *validation report* if processing fails

Fix errors flagged in the validation report in your Transit dashboard account.



With Live Updates you can provide realtime information about delays, planned or unscheduled service interruptions. Learn More | Sign Up

Dismiss

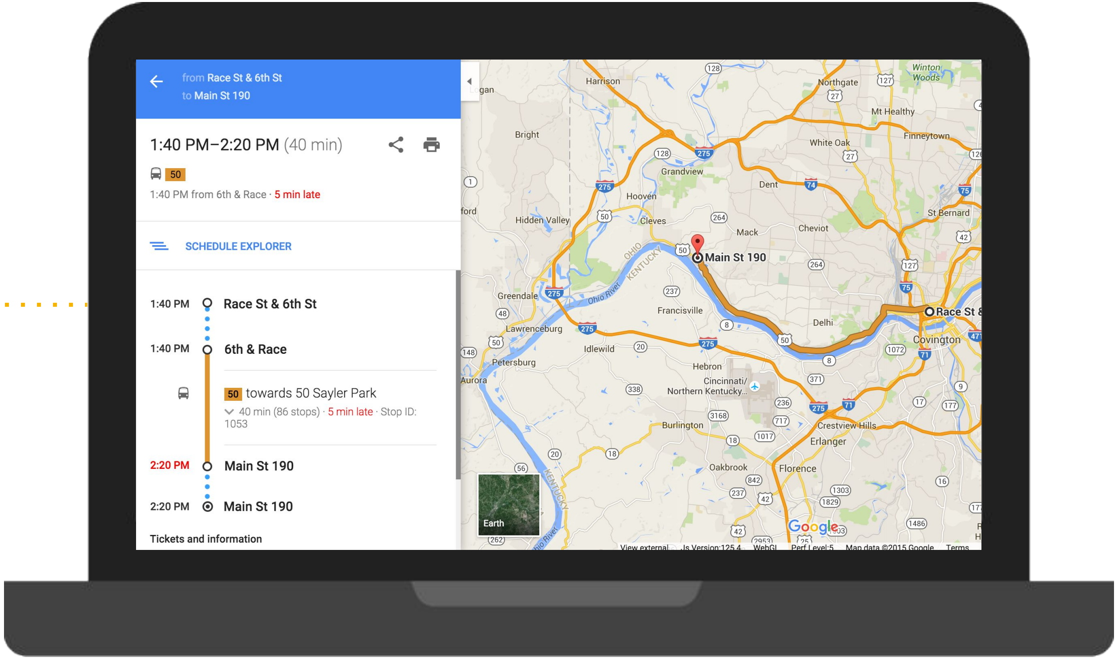
Feed: **urbes-sorocaba-br**

Trip Updates	**Debug snapshot**
Report generated	23 Dec 2015 - 07:37:58
Launch status	Live
Feed Update Stability	?
Stability	**Broken**
Success ratio	19.37%
Successful vs All Feed Updates	726 / 3,748
Configuration	
Feed name	br-sorocaba-urbes-updates
Delivery method	Fetch
Encoding	ProtoBufferAscii
Realtime data source	http://urbes.infobus.logann.c...

Learn more about using *validation reports*.

Step 4a: Test queries using *private preview*

Test your queries in the private preview mode, noted by red *Confidential*.



from Race St & 6th St
to Main St 190

1:40 PM–2:20 PM (40 min)

50

1:40 PM from 6th & Race · 5 min late

SCHEDULE EXPLORER

1:40 PM Race St & 6th St

1:40 PM 6th & Race

50 towards 50 Sayler Park
40 min (86 stops) · 5 min late · Stop ID: 1053

2:20 PM Main St 190

2:20 PM Main St 190

Tickets and information

Learn more about accessing *private preview*.

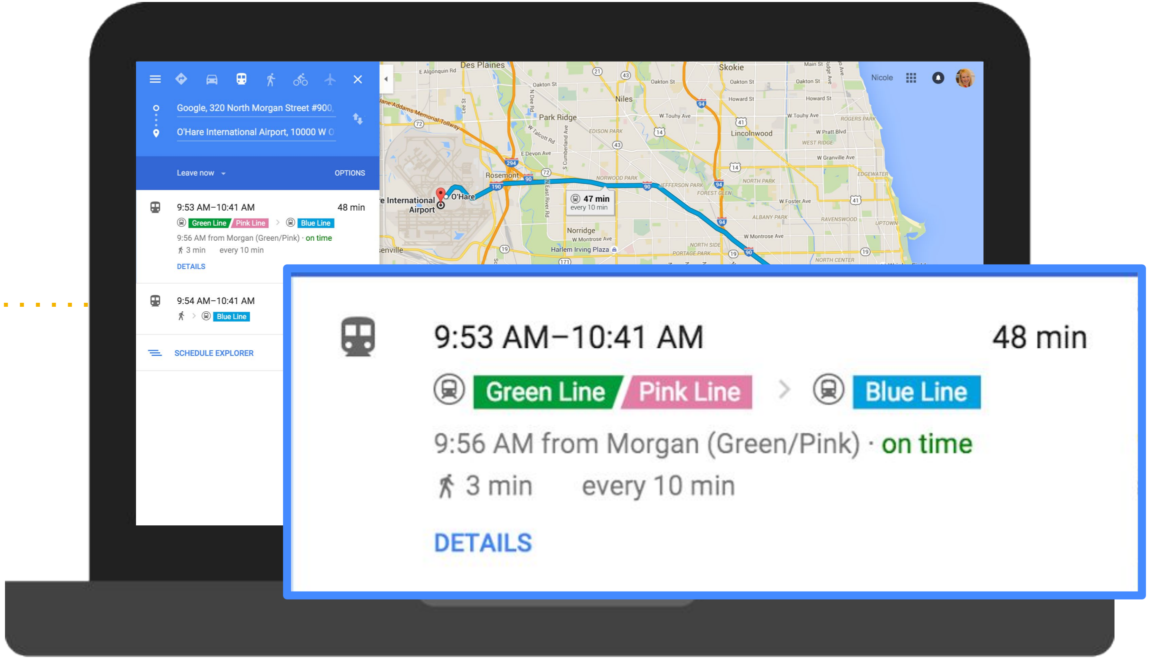
Step 4b: Reach out to your Google team for final review!

Email transit-realtime-partner-support@google.com for a final review. Google will flag any remaining issues and work with you to fix them.

Step 5: Launch your Google Transit Realtime integration!

During the agreed upon week, your integration will go live on Google Maps.

Get approval for planned press releases by emailing press@google.com.



Google, 320 North Morgan Street #900,

O'Hare International Airport, 10000 W O

Leave now

9:53 AM–10:41 AM 48 min

Green Line / Pink Line > Blue Line

9:56 AM from Morgan (Green/Pink) · on time

3 min every 10 min

DETAILS

9:54 AM–10:41 AM

Blue Line

SCHEDULE EXPLORER

Next Steps

How to get started:

☑ Submit your interest.

☑ Decide what realtime data you'll submit.

☑ Create your data feed.

☑ Provide your data feed.

☑ Validate your feed data.

Additional resources:

Transit website

Transit Help Center

Discussion group

Thank you!