



The maturity of public transport APIs 2019



Contents

Foreword	– 02
The ten cities & their API rankings	– 04
Method	– 05
Indicators of maturity	– 06
Main findings	– 13
London	– 15
Manchester	– 18
Helsinki	– 21
Barcelona	– 24
Singapore	– 27
Sydney	– 30
San Francisco	– 33
Chicago	– 36
NYC	– 39
Vancouver	– 42
Conclusions & recommendations	– 44
Signs you need a Realtime API	– 47

Foreword



Matthew O'Riordan
Founder and CEO Ably

Our most important digital interactions now happen in realtime. We expect the same from the apps and transport systems we use to traverse our cities on a daily basis. Do public transport sector APIs meet this new demand?

A number of public transport providers have been at the forefront of the open data movement for years. Across the globe many of these transport providers now offer programmatic access to their data via REST APIs. For some, it's been almost a decade. As hardware and software have become more sophisticated, so too have the demands of developers working with data of all kinds. REST APIs brought about a revolutionary change in the way organizations exchange data and the way in which developers create new applications and experiences. In transport, this saw the creation of CityMapper, Moovit, and many others. But REST APIs only satisfy request/response cycles and aren't well-suited to streaming live updates to third-party apps and services when latency of updates really matters - like with travel updates.

As a result of this, at Ably and across the wider realtime ecosystem, we're seeing developers seek out Realtime APIs in many aspects of their work - largely driven by end-user demand. This is what's behind projections that up to 30% of all data in just three years will be realtime¹. Furthermore, Gartner estimates that 50% or more of APIs need to be event-driven by then².

The nature of transport data - event-driven and time-sensitive - makes it the perfect candidate for exposing in realtime. This, combined with the rise of multi-modal transport trips, means many transport providers are already investigating or trialling Realtime APIs. Pioneers such as Transport for London (TfL) and Helsinki Regional Transport Authority (HSL) are leading the way. But behind all of this lies complex, intricate infrastructure and APIs. Proven cloud platforms like Ably are already helping transport providers offload the cost and heavy engineering lifting involved in effectively distributing data streams. And Realtime API programs are increasing developer engagement with products and services, encouraging higher levels of integration, and fostering innovation.

Foreword

But how is the industry in general faring? This maturity report looks at ten cities spread across the globe that offer programmatic access to their data via APIs, including Realtime APIs. The report focuses on three overarching categories consisting of ten different dimensions (see Indicators of Maturity section). We rank the Realtime API maturity of cities based on these dimensions.

One clear finding is that many cities are coalescing around GTFS and GTFS-RT as a de facto format for transport data. In a fragmented environment with a multitude of data formats this is good to see as it means developers can more easily consume data from multiple providers. And while GTFS has its drawbacks - it has less flexibility compared to proprietary formats - it is an open standard with active contribution and it recently saw a v2.0 release. There are now even new data formats emerging based on the GTFS standard, namely the General Bikeshare Feed Specification (GBFS), which further reduces fragmentation.

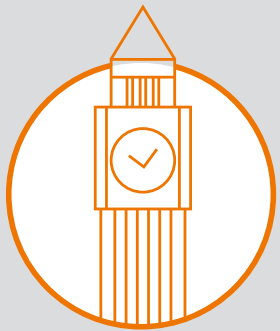
The hope for this report is that by defining a set of criteria for effectively providing Realtime APIs, both pioneers and those just beginning on their Realtime API journey can provide better Realtime APIs - and ultimately a better transport experience for citizens the world over.

¹ Data Age 2025, IDC, 2018

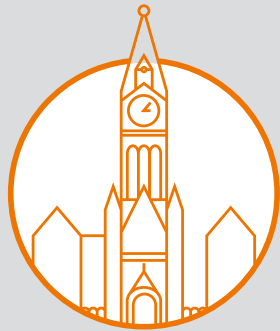
² The Impact of Event-Driven IT on API Management, Gartner, 2017

The ten cities and their rankings

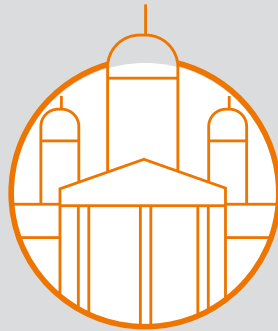
With such diverse approaches to city living around the world we aimed for a global representation of cities in this report.



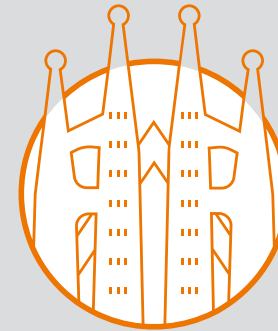
London 78/100



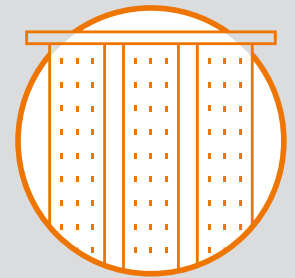
Manchester 52/100



Helsinki 80/100



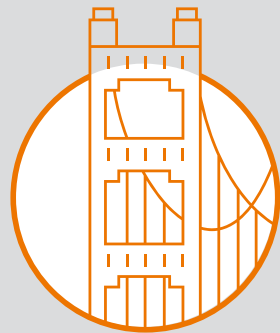
Barcelona 47/100



Singapore 58/100



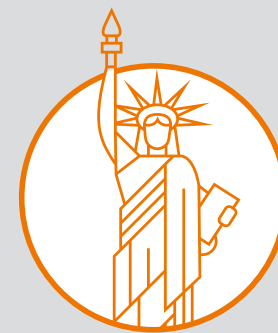
Sydney 78/100



San Fran 39/100



Chicago 47/100



NYC 36/100



Vancouver 55/100

Section 1: Method

The process of developing these dimensions began when Ably engineers sought to aggregate open streaming data sources on the Ably Hub³ - Ably's app store-like data portal where developers can find and consume realtime data streams for free as part of Ably's Open Data Streaming Program⁴.

Some of the obstacles Ably engineers faced when consuming transport APIs ranged from a lack of 'real' realtime information, to a lack of protocol support, to heterogeneous collection of data structures.

Once we began talking with transport providers about this data, some of whom feature in this report, it became apparent that some providers had a clearer vision than others about what constitutes a good developer and API experience. As such, the dimensions presented here aim to provide a ranking based on a developer's perspective on how to maximise the uptake and innovative use of realtime transport data.

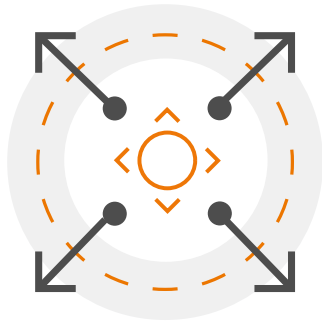
Assessment and ranking was achieved through a combination of qualitative and quantitative methods. Qualitative research began with desk research to collect freely-available information & documentation, alongside conversations with some representatives of transport providers. This data was collated and given a numeric ranking. To ensure consistency and objectivity, the same research was conducted independently of the author by a technical engineer.

³ Ably Hub: www.ably.io/hub

⁴ Ably's ODSP: www.ably.io/blog/ably-open-data-streaming-program

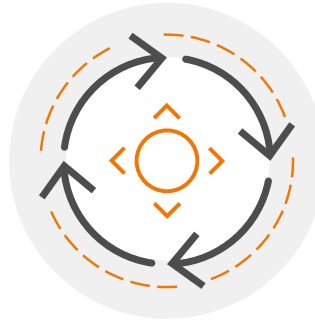
Indicators of maturity

The report ranks maturity across three categories and ten dimensions capturing different aspects of effectively deploying Realtime APIs. The three categories and ten dimensions are:



Category 01: Deployment

- #01 Type of API(s)
- #02 Data format
- #03 Data quality
- #04 Terms of use



Category 02: Consumption

- #05 Ease of API access
- #06 Documentation
- #07 Integration points and protocols
- #08 Performance and reliability



Category 03: Impact

- #09 Community
- #10 Wider economic / social impact

Category 01: Deployment



This category looks at whether a city provides useful data through Realtime APIs in a standardized format, with supportive terms of use. Each city is scored out of 10 for each Deployment dimension below.

Indicator of maturity

#1: Type of API(s)

Type of API ranks cities on two criteria:

- Does the city push streaming updates over a persistent connection to subscribers using a protocol or transport such as WebSockets, MQTT, SSE etc.?
- How many APIs are there? Is there a unified API or do developers need to work with multiple APIs for different data sets?

Indicator of maturity

#2: Data format

Data format evaluates the way data is provided based on:

- Actual format (JSON, XML, GTFS, GTFS-R etc.)
- Ease of reusing data without additional processing

Category 01: Deployment



Indicator of maturity #3: Data quality

Simply exposing data as an API doesn't make it useful:

- What data is available?
- Is the data useful for developers?
- Has the data been processed by the provider to make it easier to use?

Indicator of maturity #4: Terms of use

Transport data is generally available under an open license, but each transport provider attaches different terms of use to their own data sets:

- Do the terms of service generally encourage free, innovative use of data on a commercial basis?
- Can developers consume directly from a data provider, or must they ingest data themselves and republish it for their end-users?
- Do providers rate limit and throttle data? If so, how restrictive are the terms?

Category 02:

Consumption



This category looks at how easy it is for developers to integrate, consume, and scale consumption of realtime transport data. Each city is scored out of 10 for each Consumption dimension below.

Indicator of maturity

#5: Ease of API access

Some organizations have built excellent API experiences over the past few years, setting the developer expectation bar high. If friction of access and integration is high, developers are less likely to consume realtime data. How easy it is to access an API is a key factor:

- How easy is it to find APIs from search and from a user experience perspective?
- Is there a dedicated developer portal? Do developers need to request access?

Indicator of maturity

#6: Documentation

Documentation is one of the most important aspects of any API because, in theory, it provides everything a developer needs to know about using an API:

- Does documentation follow a common and recognizable spec such as OpenAPI / AsyncAPI?
- Does documentation provide all the information a developer needs? This includes background context on concepts or ideas relevant to consuming the API.

Category 02:

Consumption



Indicator of maturity

#7: Integration points & protocols

The ecosystem of realtime protocols is fragmented and continues to fragment. This tightly couples data producers and consumers. Consumers must often carry out custom integration work so they can consume data. For many this is a blocker to integration. This indicator determines how easy is it to consume an API and integrate it into other systems:

- How many and which protocols are supported? Are they open protocols? Is it possible to consume data directly from a provider's servers?
- How likely is it that developers will be able to use their existing code base? Would they need to do custom integration work? If so, how much?
- What are the error rates and quality of error messages on debugging?

Indicator of maturity

#8: Performance and reliability

Delivering effective, performant Realtime APIs is difficult: new types of infrastructure operating on a publish/subscribe messaging pattern are required, as opposed to traditional request/response. And unlike REST APIs, the responsibility for performance and scaling is put on the producer of data rather than the consumers:

- Do APIs follow request/response or publish/subscribe patterns? For example, does an API poll for data or push new events to consumers as they occur?
- What is the average latency and consistency of latency?
- How reliable is an API? For example, does data often fail to arrive, arrive in the wrong order, or does the service have intermittent uptime?
- Does the provider support push protocols to help consume data at scale?

Category 03:

Impact



Transport providers often expose data to drive innovation around their services and provide a better transport experience for citizens through third-party apps and services. Each city is scored out of 10 for each Impact dimension below.

Indicator of maturity

#9: Community

Building a community around open data is often beneficial to both data providers and developers.

- Is there a community around data / APIs?
- How active is the community?
- Does the community engage with one another to collaborate and drive further innovation, rather than simply requesting help from the data provider?

Indicator of maturity

#10: Wider economic / social impact

What evidence is there of developer uptake, an economic impact, and a better end-user transport experience?

- Are developers building things with this data? This should go beyond apps with global reach such as CityMapper, Moovit, Transit, and Google Maps.
- Is there any evidence of economic impact? Perhaps through internal or third-party reports/evaluations.

Section 2: Main findings & individual rankings

The next section of the report looks at the overall findings and then goes on to look at the individual cities, what they do well, and what they could do better.



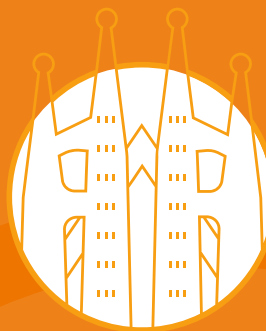
London →



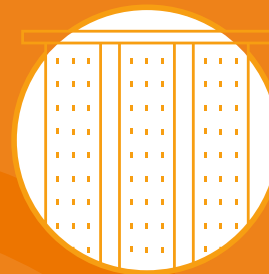
Manchester →



Helsinki →



Barcelona →



Singapore →



Sydney →



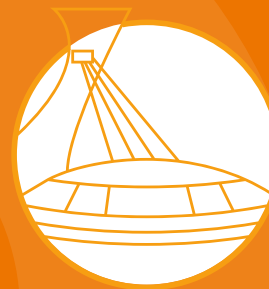
San Fran →



Chicago →



NYC →



Vancouver →

Main findings

Many cities are only just embarking on their Realtime API journey. Of the ten cities featured in this report, only two provide Realtime APIs as defined below.

Summary of key findings

What the most mature cities have in common:

- Both REST and Realtime APIs (whether WebSocket or MQTT-based) provided
- Self-service developer portals with API keys used for authentication
- Support for various data formats that make it easy for developers to reuse data
- Progressive terms of service allowing developers to consume directly from the servers of transport providers
- Reliable, performant, and available APIs developers can consistently rely on
- A strong community around data encouraging developers to interact with one another and collaborate to drive economic growth and better end-user experiences

A Realtime API is one that pushes new data from server to subscriber as it becomes available, using a protocol or transport designed for this purpose - such as WebSockets, MQTT, or SSE. This is in direct contrast to REST APIs where new data must be frequently requested.

Main findings

Summary of key findings

Common attributes of cities with less mature APIs:

- Misrepresenting data and APIs as 'realtime' when in reality updates only occur between a range of 15 seconds to five minutes
- Implementing GTFS-RT feeds and access to those feeds by polling REST APIs (aka pull mechanisms) rather than using push-based protocols such as MQTT
- Poor developer experiences that not only discourage integration and make it difficult to even get started with APIs in the first place
- Inaccessible documentation, often a PDF that follows no standardized specification
- Restrictive terms of service that discourage easy consumption of data
- A lack of community - or a forum that consists solely of troubleshooting

London



Rating: 78/100

Population served: 9,000,000

Type(s) of API: Unified REST and Realtime (WebSocket)

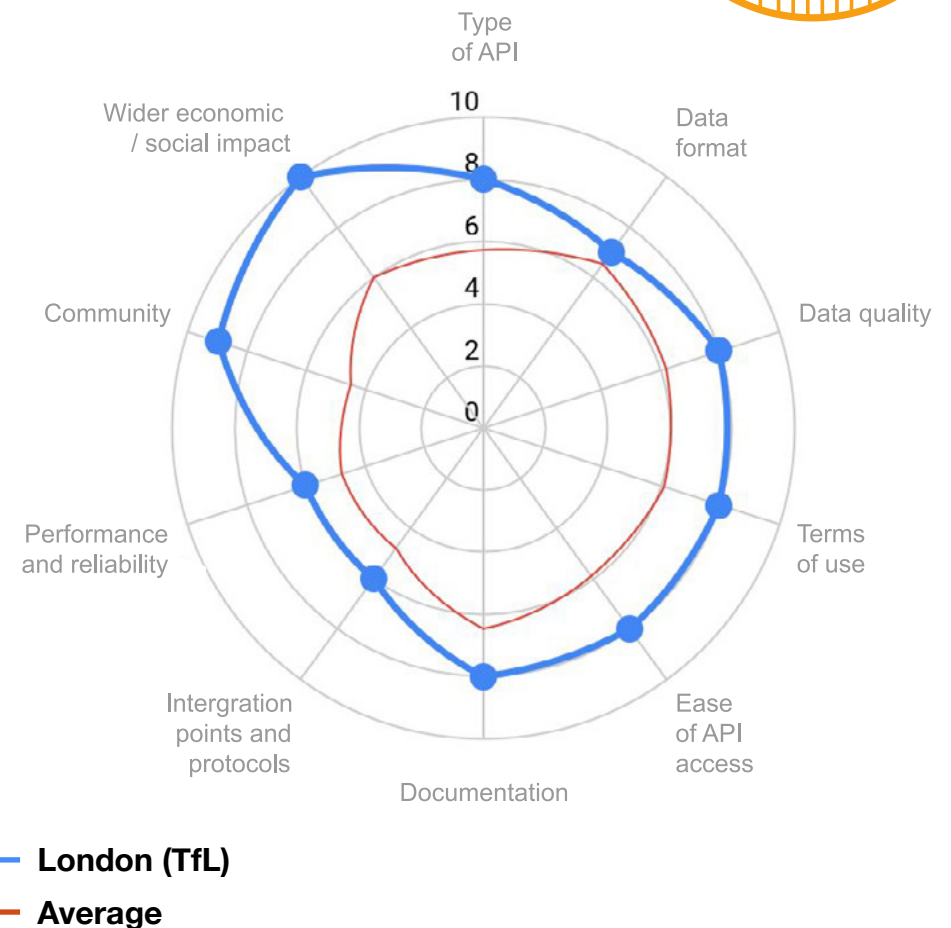
TfL's open data portal: tfl.gov.uk/info-for/open-data-users

Transport for London (TfL) is the government body responsible for implementing transport strategy and managing transport services across London. It oversees almost all aspects of transport in Europe's largest city with 24 million journeys made across London's transport network every day. TfL makes available 62 separate datasets across a mix of real-time feeds, fixed datasets, and transparency-oriented data sets- available through a unified API.

TfL now view accurate, real-time travel data as a complement to transport infrastructure in their overriding goal of serving London's transport users.

- From the Open Data Institute (ODI)⁵

5 ODI case study:
odimarket.org/case-united-kingdoms-transport-for-london.html





What TfL do well

- A single, unified API encompassing every dataset across REST and Realtime. This makes it much easier for developers to integrate with as there's only a single API. Accessing it is easy through TfL's developer portal or Postman.
- The quality of data is varied and rich, "designed with customer-facing applications in mind... supportive of common customer-facing application use cases." TfL does a lot of work to make this data available in a single format (TransXChange⁶), delivered as JSON (you can request XML, too): "The unified API presents all the data that is semantically similar for each mode of transport in the same format and consistent structures...making multi-mode application development easier."
- Documentation is good from a raw API capability perspective and is based on the OpenAPI specification. That said, code examples and live demos would add a lot.
- Community and economic impact are strong. An independent report by Deloitte⁷ found that TfL's open data generates economic value of £165m+/year to London.



What TfL could do better

- It's not immediately clear that TfL's unified API covers both REST and Realtime (WebSocket) protocols. Aside from a passing mention on the Unified API page, the only documentation available for the WebSocket aspect of the unified API is a blog post⁶. A forum post⁹ from 2017 raises this and it seems this still hasn't changed.
- Support for other protocols (such as MQTT or SSE) is also non-existent and doesn't appear to be on any roadmap.
- TfL rate limits API calls to 300 calls per minute per data feed. And when overall service is being degraded due to extensive use, TfL will throttle API access. This used to be higher (500 calls/min/feed). Perhaps TfL's infrastructure is unable to cope - it is a popular source of data with 13,000 registered developers powering 650 apps.
- TfL scored poorly overall on performance, reliability, and scalability. The service sometimes sees performance issues and high latencies. And there's zero support for pushing data into third-party systems, which would help both TfL and developers.

The verdict

TfL has a mature data offering with accessible Realtime APIs that encourage data consumption and provide a breadth of data that's seen a measurable economic impact on London. However, limited protocol support, intermittent reliability, and limited performance and scaling support mean TfL doesn't score as high as it could. Nonetheless, TfL is a leader in providing Realtime APIs for transport. We expect future scores to improve.

⁶ TransXChange: www.gov.uk/government/collections/transxchange

⁷ Deloitte report: content.tfl.gov.uk/deloitte-report-tfl-open-data.pdf

⁸ SignalR blog: blog.tfl.gov.uk/2015/12/07/unified-api-part-5-aot-arrivals-of-things

⁹ TfL forum post: techforum.tfl.gov.uk/t/javascript-signalr-interface-documentation/397

Manchester



Rating: 52/100

Population served: 2,780,000

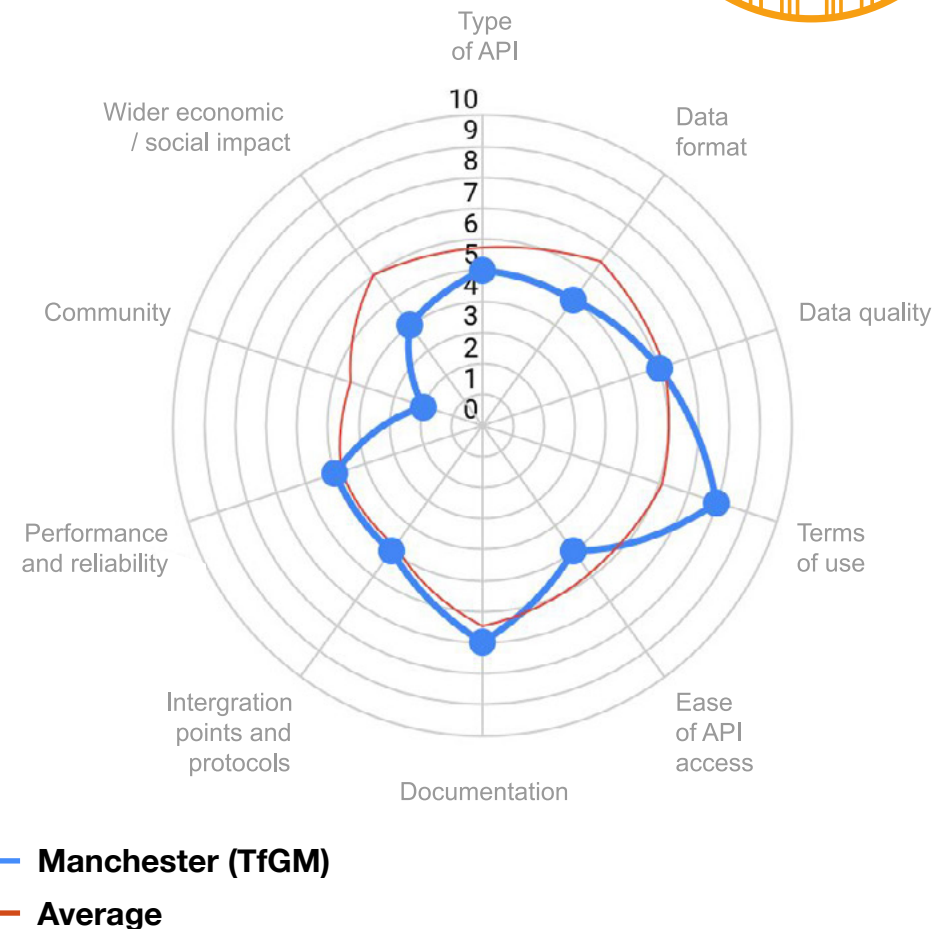
Type(s) of API: REST (long-polling)

TfGM's developer portal: developer.tfgm.com

Transport for Greater Manchester (TfGM) is the local government body responsible for delivering Greater Manchester's transport. 5.6 million journeys are made across Greater Manchester's transport network each day. TfGM provide a RESTful API so developers can access realtime and semi-static data about the transport network in Greater Manchester.

TfGM hopes that making more data available from across the transport network will inspire developers to build high quality applications creating better-informed travellers and stimulating smarter travel choices.

- Transport for Greater Manchester





What TfGM do well

- TfGM's documentation follows the OpenAPI specification, meaning the API is clearly defined with code examples across eight languages/request methods. There's also an API console where you can test the API in your browser.
- Data is provided in JSON or XML.
- TfGM is committed to actively adding new data. Metrolink fares were recently added. However, it's unclear how frequently new data types will be made available.
- Compared to some other providers with a similar score, TfGM's terms of service are a little more developer-friendly: API consumers are able to poll the API 2000 times per minute. While not comparable to continuous streaming data, this is a good start for encouraging API uptake.



What TfGM could do better

- Initially accessing TfGM's API is a little confusing. Visit the API or product section of the developer portal and you'll see a message stating "no APIs found". In order to access APIs and documentation you're required to first sign up for an account, subscribe to a product, and then visit the API page. While it does state that you need an account to access the API on the homepage, it's at the end of a dense block of text. It's generally the norm nowadays for all types of API to at least view some documentation before doing anything else. That said - the portal is completely self-service and provides simple analytics covering average response times, bandwidth, and successful/failed calls.
- TfGM rely on long-polling over XHR to provide 'realtime' data. TfGM would improve by adopting and offering additional protocols capable of maintaining persistent connections and pushing new data to clients as it appears.

The verdict

TfGM provides an API and documentation based on open specifications. Terms of service encourage developer uptake, although limits will need to be increased in the future. The user experience of the developer portal is frustrating: trying to access TfGM's API is a confusing experience. And long-polling delivers subpar performance. If TfGM improved the developer experience, offered a streaming protocol, and expanded the number of data sets then the maturity score would drastically improve. That said, TfGM is on the right path.

Helsinki



Rating: 80/100

Population served: 1,400,000

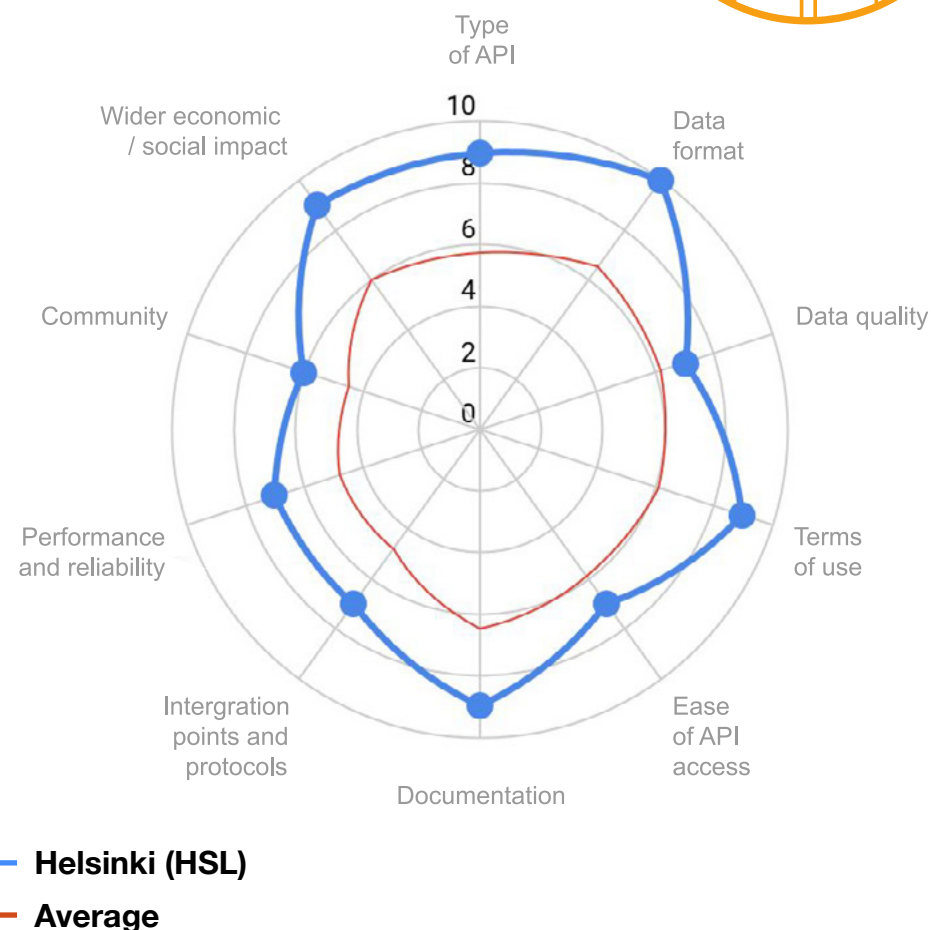
Type(s) of API: REST, GraphQL, and Realtime (MQTT)

HSL's developer portal: www.hsl.fi/en/opendata

Helsinki Regional Transport Authority (HSL) is the transport authority in the Helsinki region. It provides ticketing and information systems for all operators in the area. HSL procures, plans, and optimizes transport for all transport modes i.e. tram, metro, bus, ferry and commuter train. HSL's network is used by over 1 million passengers each day.

Documentation, community/support, and reliability are essential for providing a good Realtime API experience for developers.

- Sami Räsänen, Development Team Lead, HSL





What HSL do well

- HSL provides seven different APIs, including routing, geocoding, and map APIs. APIs are varied including GraphQL, REST, and Realtime (MQTT). Note that the realtime APIs are in beta. That said, HSL provide sophisticated MQTT endpoints that work to scale to all end-users. Unlike many other providers, HSL don't require developers to proxy: HSL is happy for them to provide endpoints directly to end-user devices.
- HSL's documentation provides clear information about APIs, HSL's architecture and infrastructure, and explaining concepts behind the technology they're using (for example, GraphQL and GraphiQL). The APIs all follow OpenAPI spec with code samples and live demos (which use the GTFS-RT library) to test APIs in the browser.
- HSL provide Docker images and access to their scripts for deploying, making it easier for developers to get up and running. HTTPS and MQTT endpoints are available, along with GTFS-RT feeds. Furthermore, HSL has a clear roadmap and is working to provide more protocols and integration options in the future.
- There is a strong focus on building a developer community around HSL's data and encouraging/supporting developers to make the best use of data. This has led to clear economic and social impact with over 230 applications¹⁰ in production, using open data to solve cross-border traffic issues, and playing host to MaaS Global¹¹ - an app that combines public and private transport into a single app and subscription.

¹⁰ Applications in production: hri.fi/en_gb

¹¹ MaaS global: maas.global



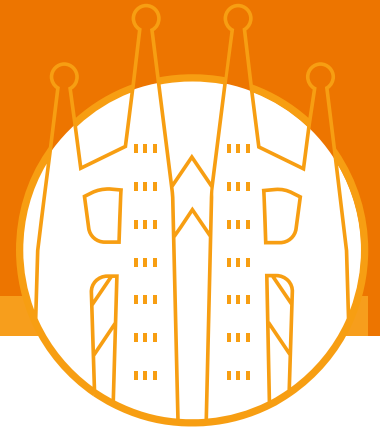
What HSL could do better

- HSL doesn't have a self-service developer portal. In order to access the main API ('sales API') and receive API keys, developers must sign a contract - accomplished via email. Access to the data is still free. While not laborious, self-service is better.
- HSL's Realtime APIs are still in beta. While there seems to be little risk of this being discontinued, it would still be good to see this graduate from beta soon.
- There are multiple APIs developers must integrate with and consume from. A unified API could help ease integration for developers and encourage higher uptake.

The verdict

HSL is leading the way. It offers a mature set of data across well-defined and documented REST and Realtime APIs. And HSL is planning to add multiple protocol support. A developer portal and graduation from beta would drastically increase HSL's already-impressive score.

Barcelona



Rating: 47/100

Population served: 1,600,000

Type(s) of API: RPC-based (polling)

Open Data BCN portal: opendata-ajuntament.barcelona.cat/en

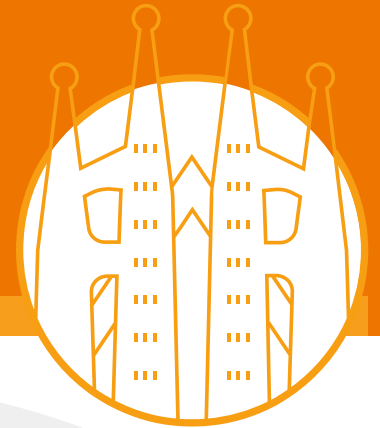
Transport data for Barcelona comes from Open Data BCN, a public sector initiative that is now part of Barcelona's wider Digital City strategy and managed by Barcelona's Municipal Data Office. Transport data sets are owned by various government departments.

Barcelona provides open data to promote innovation. Apart from the common and democratic value, another of the essential pillars of the open data project is the economic potential. Individuals and companies will be able to make the most of the products and services provided based on the re-use of this public information.

- Open Data BCN



Barcelona



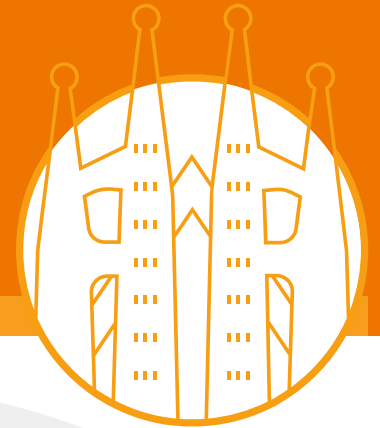
What Barcelona does well

- Barcelona is actively trying to foster community and innovation around its data.
- Support for multiple data formats: JSON, WMTS, WMS, and DCAT.
- Despite having only five realtime data sets, they're future-focused. For example, one data set shows the status of Electric Vehicle charging points across the city.
- APIs are easy to access - they are publicly available so no API keys are needed. Although this shows that Barcelona's API offering is still relatively immature.

What Barcelona could do better

- The API is currently an RPC-based API provided by CKAN - which is a good start for exposing data programmatically. But Barcelona would do better to provide its own REST and streaming/event-driven APIs as it matures its offering.
- Documentation is limited and relies on CKAN to provide API reference.

Barcelona



What Barcelona could do better

- Data classified as “instant” (aka realtime) is often updated only every minute or every five minutes. This links to a wider theme of a lack of ‘real’ realtime data.
- Terms of service are unclear with clauses reserving the right to charge for data.
- Integration points and protocols are lacking, with unknown levels of reliability or performance.

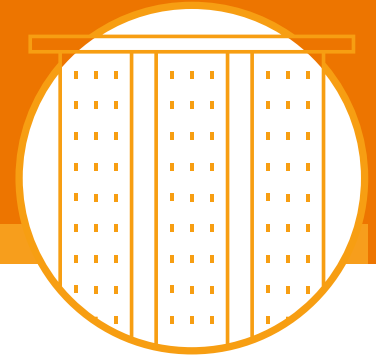
The verdict

Barcelona’s Realtime API program is still in its infancy. There is a wide set of general open data but realtime transport data is severely limited. Terms of service aren’t clear and a clause reserves the right to charge for data, which could discourage developer uptake.

The API itself is RPC-based. While not bad per se, it’s less common than REST, isn’t based on the OpenAPI spec, and is run through CKAN - an open platform for publishing and aggregating open data of all kinds. For developers familiar with REST, this might be frustrating.

Barcelona is early in its realtime data journey - and therefore Realtime API - journey. To help developers create more economic value for the city, Barcelona should continue to develop its existing API offering and look to provide more frequently-updated data over Realtime APIs.

Singapore



Rating: 58/100

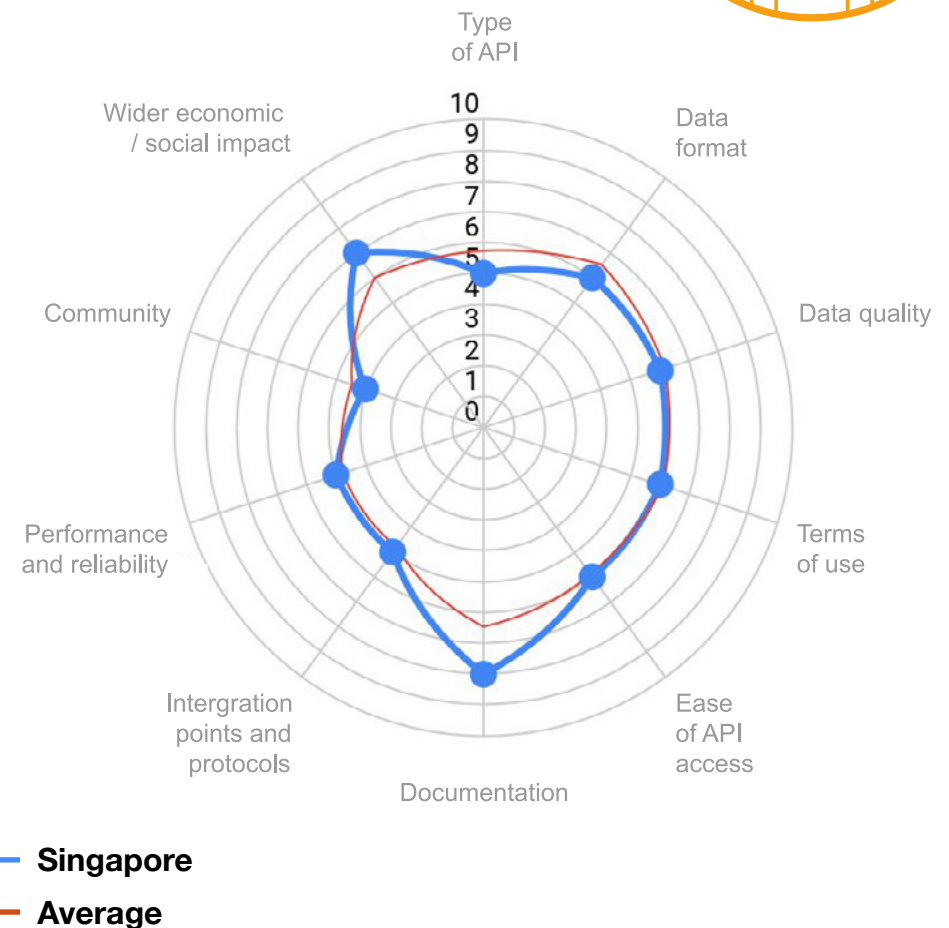
Population served: 5,600,000

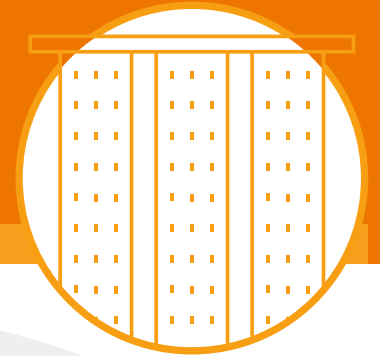
Type(s) of API: REST (long-polling)

Singapore's open data portal: data.gov.sg

Singapore has been at the forefront of open data and is one of the most technologically advanced cities in the world. The engineering team are mature and write often about the need for open data - including the need for realtime data. Transport data for Singapore is provided through the data.gov.sg open data portal. Data sets are provided by different government departments. All data sets are available through Singapore's REST API.

The aims of this portal include providing one-stop access to the government's publicly-available data and creating value by catalysing application development.

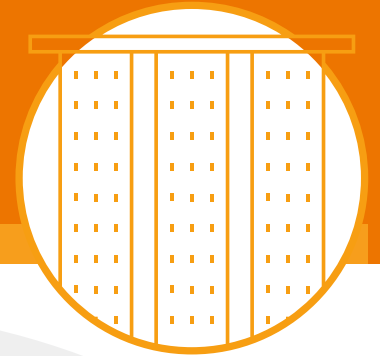




What Singapore does well

- Singapore's open data portal is developer-friendly, offering a good user experience that makes it easy to find and consume APIs. Each data set also suggests complementary data sets and, even if not realtime, can be useful for developers trying to build applications and services for Singaporeans. DataMall is another way to access APIs - a website provided by Singapore's Land Transport Authority (LTA).
- Documentation is clear and based on the OpenAPI spec, as such docs cover endpoints, parameters, response and error codes, and schema. There's also code examples and live demos that allow you to try out the API in the browser. Overall, a strong developer experience.
- Terms of service are supportive as it seems developers are encouraged to consume directly from the data.gov.sg servers. That said, there's no information on rate limiting in terms of service, aside from a passing mention that each API has individual limits. But these limits aren't to be found on individual API pages.

Singapore



What Singapore could do better

- As with many other cities in this report, Singapore claims data is 'realtime' but in reality this covers update timeframes from 1 minute to daily. The suggestion is to poll for updates. It would be more resource-efficient and a better overall experience for developers and end users if Singapore adopted protocols supporting push-based, event-driven data exchange, such as WebSockets, MQTT, or SSE.
- Wider economic impact is strong although centralized around government departments. However, general groundwork for a strong realtime API community has been laid and Singapore is actively fostering a developer community.
- Data sets are limited - more diverse data in the future will fuel further innovation.

The verdict

Singapore has a wide-ranging open data and API programme. However, Singapore is sorely let down by offering only REST APIs that rely on polling multiple endpoints. A unified API based on a persistent-connection-based protocol, such as WebSockets or MQTT, would drastically improve scores across the board. That said, it's clear Singapore is actively thinking about how it can create Realtime APIs to benefit Singaporeans. There is great potential here and we expect Singapore's score to rise in the future.

Sydney



Rating: 78/100

Population served: 5,200,000

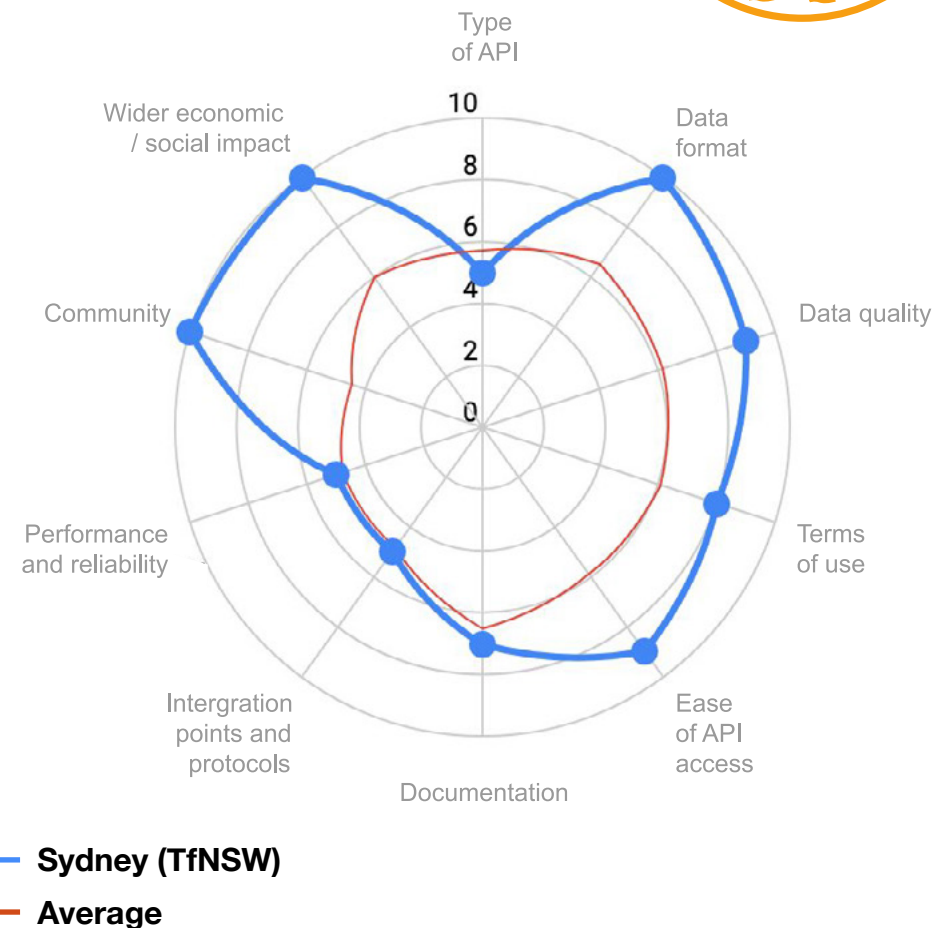
Type(s) of API: REST (long-polling)

TfL's open data portal: opendata.transport.nsw.gov.au/developer-information

Transport for New South Wales (TfNSW) lead the development of safe, integrated and efficient transport systems for the people of NSW. Through its open data portal, TfNSW supplies realtime data to apps with over 7 million unique customer downloads in total. TfNSW's Open Data Program makes these datasets, along with other transport data, more broadly available. TfNSW wants this data to support apps and 'a whole lot more'.

Transport for NSW is committed to fostering innovation by providing open access to our data.

- Transport for New South Wales





What TfNSW do well

- TfNSW provide an excellent overall API experience with extensive support for different data formats (including TransXChange - the same as TfL) and a rich range of data across all modes of transport and transport-related data (e.g. fuel levels).
- One of the best developer experiences for transport. Finding and accessing APIs was easy with extensive documentation, API explorers, tutorials, and blog posts to help developers understand APIs and what they can do with data for real impact.
- Driving innovation and strong economic impact with a great community of developers exchanging ideas about how to use realtime data to build things rather than simply complaining or seeking assistance. TfNSW also want to create a new data format to support the growth of MaaS, involving community submissions.



What TfNSW could do better

- TfNSW provide a REST API which developers must poll every 10-15 seconds. This is much more frequent than other cities ranked in this report (30 seconds to 5 minutes). But it's disappointing that TfNSW, with such a great range of data and a generally mature API offering, sadly don't provide push-based protocols like WebSockets.
- Terms of use are among the more generous - 300 calls per minute - and it's possible for end-user clients to consume directly from TfNSW's APIs. However, in reality this isn't feasible and most developers are ingesting and caching data every 15 seconds. Push-based protocols would help overcome this without overloading servers.
- Documentation is extensive. However, actual official documentation is presented as separate PDFs for each API. This means TfNSW takes a big hit on this dimension as this isn't an accessible way of providing docs.

The verdict

TfNSW's API programme is impressive but sadly falls short on a couple of major issues. Access to APIs, information to help developers, and community are by far the most mature of this report. But data is provided via GTFS-R feeds over REST, refreshing every 15 seconds and which developers must poll. TfNSW score highly but with just a single push-based protocol TfNSW would have been the leader of this report. That aside, TfNSW is a shining example of how to properly provide APIs and foster innovation around transport services for a better citizen experience. We're excited to see what TfNSW does in the future.

San Francisco



Rating: 39/100

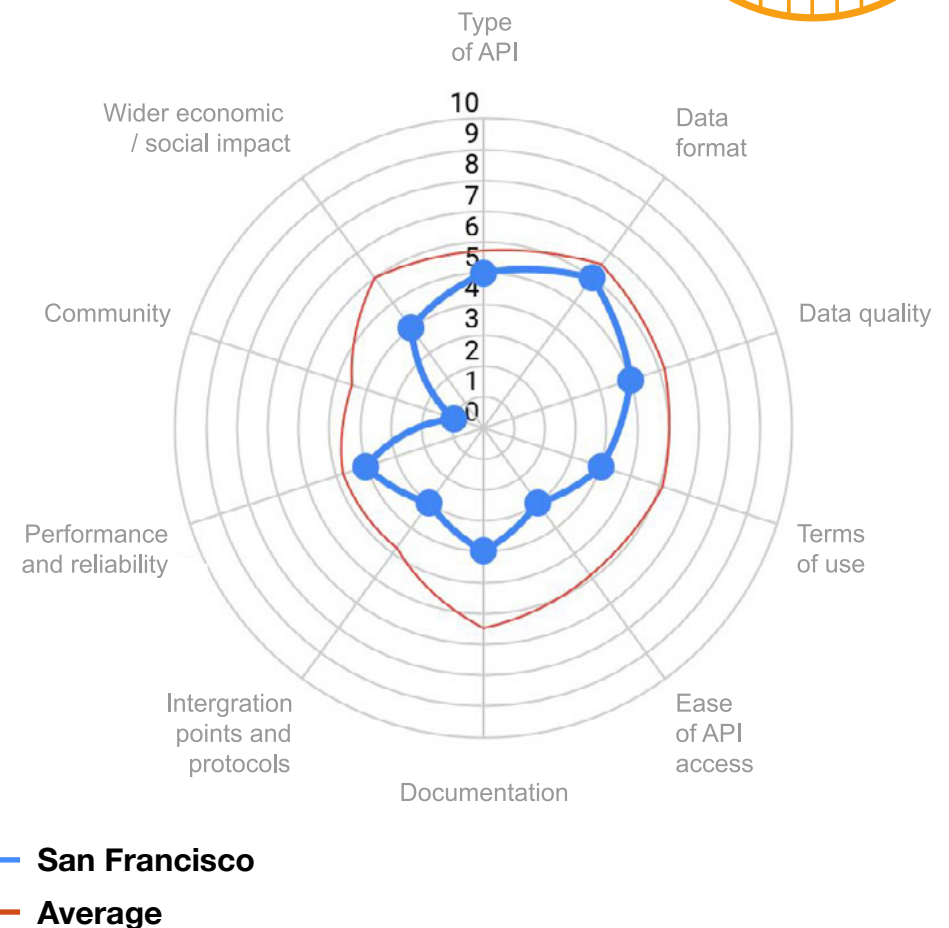
Population served: 7,750,000

Type(s) of API: REST (long-polling)

511's developer portal: 511.org

BART: api.bart.gov/docs/overview/index.aspx

Transport data available as APIs for San Francisco was somewhat difficult to find. San Francisco has an open data portal but searching for realtime APIs wasn't easy. It turns out a third party, 511, consolidates and presents the Bay Area's "up-to-the-minute" transport data. To make things more confusing, BART, 'rapid rail transport' also has its own API. As such, the ranking is a combination of 511 and BART APIs. The Bay Area is geographically vast with a large transportation network. It was disappointing to find such a fragmented open data landscape lacking good data.



San Francisco



What 511 + BART do well

- Across 511 and BART, there's an array of data formats available including XML, JSON, NeTEX, SIRI, and GTFS and GTFS-RT. Yet this also reflects the fragmented nature of transport APIs in the Bay Area.

What 511 + BART could do better

- Fragmentation of APIs makes it difficult to even initially find who provides APIs. 511's developer experience is among one of the worst in this report: when validating your email after signing up, you're redirected to a 404 error page. Another email sent after this contains API keys.
- Documentation is poor, dense, follows no clear specification and is offered only as PDFs - one of which was 102 pages. This means no live demos or in-browser testing. One user found the documentation so dense he created a cheat sheet for others.
- Integration points and protocols score badly. Developers must consume data and republish it for their end-users.

San Francisco



What 511 + BART could do better

- The API relies on GTFS-RT feeds and polling over REST so performance isn't great.
- There's no evidence of a community, support, or even much of an economic impact of this data being available. There are apps and services built on this data but, outside the large global players, there isn't clear evidence of local applications.
- Terms of service reserve the right to charge for data in the future. This isn't unique to the Bay Area but it suggests an environment that isn't focused on innovation.

The verdict

With so many innovative tech companies based in the area, the San Francisco Bay Area rating is disappointing. It was difficult to initially even find where to access transport APIs. Once found, it was a fragmented ecosystem. Breadth of data was poor along with performance: BART's API offers only realtime estimates. Generally, data is over reliant on GTFS-RT feeds and polling REST APIs. Documentation for APIs ranks joint worst with NYC. A collection of dense PDFs following no specification makes for a poor developer experience, introducing huge friction to adoption. And there's not much worse than receiving 404s after validating your account. Overall this was one of the most frustrating and disappointing API experiences. Improving developer experience across API access, documentation, and consolidation of APIs will go a long way to improving this score.

Chicago



Rating: 47/100

Population served: 3,500,000

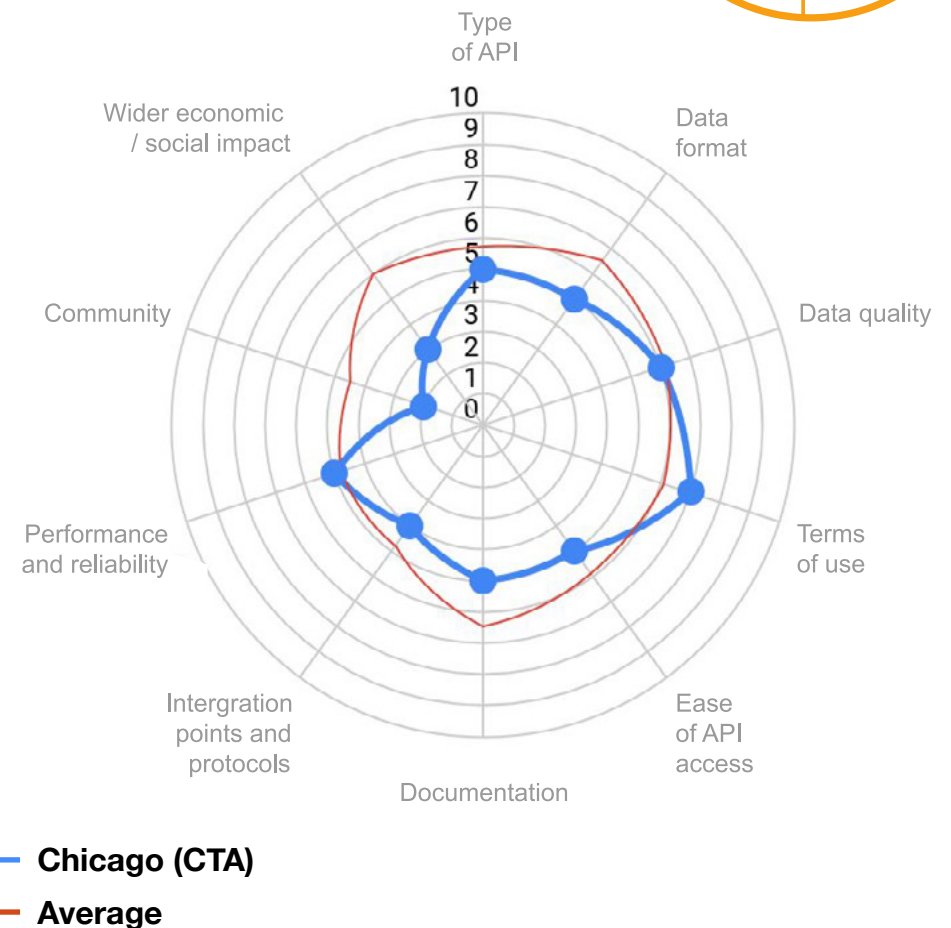
Type(s) of API: REST (long-polling)

CTA's developer portal: www.transitchicago.com/developers

The Chicago Transit Authority (CTA) operates the second largest public transportation system in the USA. It covers the City of Chicago and 35 surrounding suburbs. On an average weekday, approximately 1.6 million rides are taken on the CTA.

CTA's data offerings help developers create interesting new applications and mash-ups that help people get the information they want or need about CTA services, wherever they want to receive it.

- *The Chicago Transit Authority (CTA)*





What CTA do well

- CTA's terms of use encourage developer adoption and reduce friction by allowing developers to consume directly from CTA servers. As with other providers they do limit use with daily caps along with encouraging caching of data. CTA also provide helpful guidance on which data sets to cache and recommended polling frequency (every one minute - which is when data is refreshed).
- Documentation is helpful with a "getting started" page that helps developers understand how to make the best use of data. Despite not following an open spec, the documentation is useful, covering parameters, responses, error codes, and so on. The docs also provide additional context around how CTA provides realtime data. CTA could do better with its bus docs: they're currently only available as PDF.



What CTA could do better

- CTA rely on REST APIs limited to polling every one minute. This isn't ideal for realtime updates. It also seems the accuracy and reliability of data is not always as good as it could be, with realtime data often reverting to static schedules. Adding a protocol such as MQTT or WebSockets would help improve performance and lighten the load on CTA's servers, further helping developers consuming CTA APIs.
- Accessing APIs could definitely be better. To access the Train Tracker API you must fill out a form at the end of a terms of service page. To consume the Bus Tracker API you need to visit ctabustracker.com and then request an API key through another form. This isn't difficult but it's not coherent and leaves you feeling a little disjointed.
- CTA has tried to create a community around its data but the forums are very inactive with only a single post in 2019 as of July 2019. There is an app showcase page with transport apps built using this data so there is evidence of some economic and social impact for Chicagoans.

The verdict

CTA has made a strong start in providing a good developer experience. By consolidating access to APIs into a single developer portal, ensuring documentation is available online, and adopting a standardized specification the experience would greatly improve. To foster innovation and provide more performant and useful data, simply adopting a push-based protocol would go a long way. CTA is promising and we expect good things in the future.

New York City



Rating: 35/100

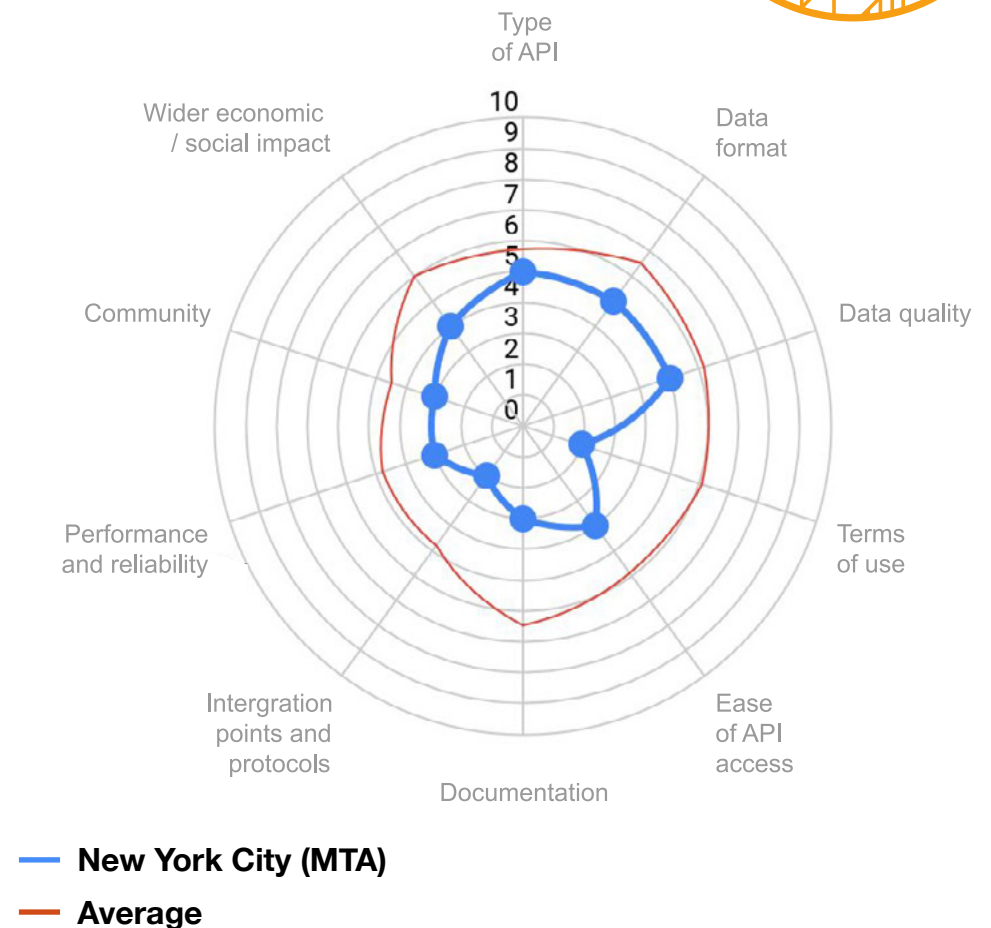
Population served: 15,300,000

Type(s) of API: REST (long-polling)

MTA's developer portal: datamine.mta.info

The Metropolitan Transport Authority (MTA) is North America's largest transportation network, serving a population of 15.3 million people across a 5,000-square-mile travel area surrounding New York City. The MTA network comprises the USA's largest bus fleet and more subway and commuter rail cars than all other U.S. transit systems combined.

Unfortunately, as with the San Francisco Bay Area, the transport data and APIs available through the MTA leave a lot to be desired.



New York City



What MTA do well

- Data formats include XML, JSON, and GTFS and GTFS-RT.

What MTA could do better

- MTA relies heavily on GTFS-RT feeds refreshed every 30 seconds and made available over a REST API. As with many other providers in this report, offering push-based protocols would increase performance and help match demand for realtime data.
- Quality and breadth of data is limited to location and schedules.
- Terms of use are the worst in this report. Developers must download and host MTA data on their own servers. Any developer not doing so will be banned. This is extremely prohibitive and undoubtedly stifles innovation and developer uptake of APIs. Many other providers allow developers to consume directly from their servers and cache data - some even allow end-user clients to consume from their servers. Forcing developers to download entire data feeds on a regular basis and then rehosting them on their own servers will only kill demand for APIs.

New York City



What MTA could do better

- Accessing APIs and documentation are frustrating experiences. There is a self-service portal but the user experience is poor. Code and API endpoint examples are displayed as hyperlinks rather than code examples - leaving you confused. Documentation once found and downloaded as a PDF provides operators and parameters but no code examples, no live demos, and doesn't follow a specification.
- The MTA developer forums are full of developers with problems and it seems the performance and reliability of MTA APIs isn't the best. A problem with MTA's CDN seemed to go on for over a week.

The verdict

A poor developer experience, restrictive terms of service, a lack of useful data, and polling over REST, all done badly, combine into the lowest ranking in this report. Offering different and push-based protocols is important. But for MTA the most important aspects to improve on should be enhancing the developer experience with a better self-service developer portal. This includes making documentation available as a webpage and following a standard spec. Working on server capacity so developers can consume directly from MTA, even if rate limited, would be better than downloading entire datasets. If MTA works on these things we expect them to rise in the ranks.

Vancouver



Rating: 55/100

Population served: 2,500,000

Type(s) of API: REST (long-polling)

TL's developer portal: developer.translink.ca

TransLink (TL) delivers public transit in a service area spanning more than 1,800 square kilometres, providing for the transportation needs of 2.5m Metro Vancouver residents. These include an extensive bus system throughout the region, SkyTrain rapid transit, SeaBus passenger ferries, West Coast Express commuter rail, and HandyDART for passengers who are unable to use conventional transit.

The TransLink Open API provides access to transportation data, allowing developers to build applications enriched with TransLink data.

- TransLink (TL)





What TL do well

- TL provide a clean, self-service developer portal making accessing API keys quick & painless.
- Documentation is clear - but lacks code examples or live demos.
- Community is in progress with active boards and semi-responsive replies from TL.

What TL could do better

- Updates only occur every two minutes over REST, with XML the default output (JSON is available with an additional header request). This brings the overall score down as both API type and default format could be improved.
- The range of data is growing but still limited. Developers have been asking for SkyTrain data for quite some time but this is still unavailable.
- Terms of service allow developers to consume directly from TL servers but 1,000 calls per day is very limiting. There's also a clause stating developers might be charged for access to data if they in turn are charging for apps downstream.
- Performance and reliability could be better. There are numerous and regular complaints about inaccurate data and API unavailability: the API sometimes returns scheduled departures rather than realtime data and there are issues with latency.

The verdict

Vancouver has a solid API program. There are some kinks, such as performance, reliability, and ability to consume from TL servers, but it does look like TL is actively working to improve its API program: the realtime API recently graduated from beta. If TL works on their server capacity, implements a streaming protocol to enable more frequent updates, Vancouver will develop a strong Realtime API program in the future.

Conclusions and recommendations

There will always be contextual dimensions when considering such diverse cities. But there seem to be common threads where cities are excelling and where they are struggling.

The areas in which cities are excelling

- As a general rule, most cities provide supportive terms of service that allow developers to consume directly from their servers. However, heavy rate-limits exist and only a few cities encourage end-user clients to consume directly from them.
- Documentation, while sometimes offered only as a PDF and not following OpenAPI spec, does provide enough information to effectively use an API.
- Most providers offer three data formats. Many cities offer GTFS and GTFS-RT, which is helpful to developers consuming transport APIs in such a fragmented ecosystem.

Conclusions and recommendations

The areas in which cities are struggling

- Only a few cities make it easy to find, access, and consume APIs. Cities across the board must get better at providing a clean user experience. Developers are accustomed to slick interfaces and exquisite developer portals in their other work.
- Performance and reliability, even from the leaders of this report, could be improved. This is often down to limited resources but there are third-party solutions to both.
- Integration points and protocols suffer too. Only two cities offer Realtime APIs - the others offer polling through a REST API at varying intervals. This is something providers will need to address sooner rather than later. Developers expect realtime data pushed to them as it happens. Again, there are solutions out there for this.

Conclusions and recommendations

Recommendations

To build thriving developer ecosystems around Realtime APIs we recommend that cities:

- Provide push feeds to keep up with the ever-growing need for realtime data.
- Provide data over as many common protocols as is feasible (WebSockets, MQTT etc), ensuring data is accessible for as many developers and applications as possible.
- Provide varied levels of detail dependant on the end-user's request, ensuring the consumer is receiving the data they need. Look to further enhance usability of APIs.
- Invest into increasing rate limits, opening data up as much as possible. In the long-run, it'll be worth it as demonstrated in London, Helsinki, and Sydney.

Signs you need a Realtime API

From the research presented in this report, we've drawn six commonly occurring shortfalls in transport data provision that could easily be solved by providing a Realtime API:

- **Consumers would benefit from a realtime experience.** If updates such as train arrival or delay information would provide a better user experience for commuters, yet delays exist because your APIs are not event-driven (updates are pushed as they happen to commuters), then you need a Realtime API.
- **Rate limiting.** REST APIs are rate-limited because your customers are pulling data too frequently.
- **Throttling.** You're throttling how much data developers can consume because your infrastructure cannot support a high concurrency of API requests.
- **WebHooks.** As developers want event-driven APIs, you've built a WebHooks solution. You're now considering support for WebSub, and worrying about scaling.
- **Quality of Service issues.** Developers want better guarantees around delivery and integrity of updates, such as guaranteed ordering and delivery of data.
- **Last digital mile delivery monetization missed.** You expect developers to consume data, process on their servers and push to their customers' devices with their own infrastructure, instead of supporting options for last digital mile delivery to devices such as mobiles, and potentially monetizing that.

Learn more about how Ably can help you distribute transit data feeds as Realtime APIs:
www.ably.io/api-streamer/public-transport

A ready-built platform for Realtime transit APIs

Offload the cost and heavy engineering involved in effectively deploying Realtime APIs beyond REST:

- Provide a better developer experience with self-service developer portals (registration, API auth, docs, and more) and optional last digital mile delivery (streaming data direct to end-user devices) to help you reduce costs or increase revenue.
- Encourage higher levels of integration with first-class support for GTFS-RT v2.0 and JSON / MsgPack interoperability. And with over 20 realtime protocols (including WebSockets, MQTT, SSE) you have the flexibility to distribute Realtime APIs regardless of data format.
- Drastically reduce Total Cost of Ownership (TOC) with Ably's fully-managed serverless infrastructure. Absolutely no upfront or ongoing engineering work with predictably low and manageable costs, all while expanding your API capabilities.

To learn more about how Ably can help you with Realtime APIs, please contact us.

**www.ably.io/api-streamer/public-transport / sales@ably.io
+1877 434 5287 (USA, toll-free) / +44 20 3318 4689 (UK)**