Transport for London

Schematic maps standard

This document introduces the TfL schematic mapping toolkit, describes the features and explains how it can be used. It gives an overview of what the toolkit does and does not provide.

Audience

- Developers
- Technical architects
- Designers

Requirements

1. Overarching requirements:

1.1. Maps on TfL Online's digital sites and services **must**:

- Work seamlessly in all our core browsers
- Work across devices as widely as possible with the same experience
- Look crisp at all zoom levels
- Be able to show disruptions in realtime
- Be supportable, and data driven
- Be interactive, and event driven
- Follow TfL's style guides
- Integrate with our release / support workflow

2. Overview

2.1 The schematic maps are developed in Adobe Illustrator, which can export images in SVG format (a specialisation of XML)

2.2 Working with a supplier, we can cut the lines up into logical sections between stations, and annotate them with codes that allow us to connect the final SVG directly to the disruptions API, and turn the SVG into an application embedded straight into a web page or third party site

2.3 SVG can be manipulated by the standard jQuery library that lies at the core of our site, allowing line colours, text, glyphs etc to be changed dynamically in the client browser

2.4 This solution is highly cache-able, as we send a default SVG to every client and embellish it with Javascript in their browser, minimising the payload

2.5 Those without Javascript on IE could get the static image and text version, and those on Chrome, Firefox or similar could get the SVG static version

3. Benefits of this approach

3.1 This approach is beneficial to TfL Online because it:

- Decouples us from all upstream design changes
- Works with all schematic maps
- Integrates into our workflow
- Ensures maps work on all modern mobiles and browsers
- Has an accessible / progressive enhancement fallback
- Provides a single source of the truth: we use the exact same map data as the print version

4. Associated risks

4.1 Risks of adopting this approach include:

- The chance a supplier will make mistakes with the data mark-up
- The chance that the maps will not be cut up according to the specification in the technical details section

5. Risk mitigation strategy

5.1 To help mitigate risks:

- The data mark-up can be run against the API and reconciled pre-release

- The cuts can be checked by hooking up a new SVG to an incident-generating tool and tested

6. Technical details

6.1 Formats and standards

6.1.1 The TfL API is based on different standards and datasets for different components

6.1.2 Stops, stations and piers come from NaPTaN

6.1.3 Routes come from TransXChange

6.1.4 Disruptions are filterable by NaPTaN ID, Mode and Route

6.2 Naming conventions

6.2.1 All line segments are referred to by LineCode_FromNaptanID_ToNaptanID

6.2.2 Simple stations are referred to by their LineCode_NaptanID pair

6.2.3 Interchange stations have more than one line serving them, and are marked as LineCode1_LineCode2_LineCodeN_NaptanID with LineCode's sorted in ALPHA ASC

6.3 Optimisations and pre-processing

6.3.1 Due to the verbose nature of SVG, and the fact the source maps can contain more data than we wish to display, (eg fare zones), the SVG files get sent through an optimiser to remove elements that are not needed, and to do any further data normalisation necessary

6.4 Cutting segments

6.4.1 Lines

As a general principle, lines need to be separated between consecutive stations. For example, to achieve a disruption between Turnham Green and Ravenscourt Park, we would want to highlight:

- From the right of Turnham Green station to the left of Stamford Brook station as a section

- Then Stamford Brook station, as detailed in the simple stations section above
- Then from the right of Stamford Brook station to the left most of Ravenscourt Park station

To allow for disruptions along any path, we need route lines to be complete, even if that means they are seemingly redundantly layered.

As a working example, allowing disruptions between Turnham Green and either Chiswick Park or Gunnersbury, we would need the line to be complete between Chiswick Park and Turnham Green, as in Figure 1:

Figure 1

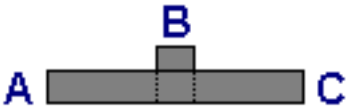
And between Chiswick Park and Gunnersbury, as per Figure 2:

Figure 2

6.4.2 Simple stations

Simple stations are cut three ways:

- The station itself is separated, and is defined as the area external to the route line
- The section from A to B is cut at the right-most extremity of the B area, being the longest length encompassing A and B
- The section from B to C is cut at the left-most extremity of the B area, being the longest length encompassing C and B



6.4.3 Interchange stations

Interchange stations come in two varieties:

- Roundels (eg Paddington)
  Roundel stations are marked as:

  LineCode1_LineCode2_LineCodeN_NaptanID with LineCode's sorted in ALPHA ASC

- Standard station stems (eg Temple)

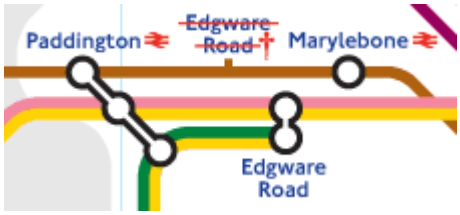
  Standard stations are just marked as LineCode_NaptanID

6.4.4 Known issues and limitations

Due to the shape of the Circle line - which resembles that of a mouse - and the nature of the API structure, we can't tell if a disruption on the Circle line between Edgware Road and Paddington is on the ‘body’ (or main Circle line) segment, or on the ‘tail’ (or Circle line extension) segment.

As such, a disruption would end up with both sections being highlighted. A longer chain of disruption would allow us to determine the branch, but this section alone could be problematic. See Figure 3:

Figure 3



Similarly, the London Overground could possibly be treated as separate logical lines (eg Gospel Oak to Barking) to avoid a similar issue, where we wouldn’t be able to determine which branch a disruption referred to in the case of Canonbury to Highbury and Islington, as shown in Figure 4:

Figure 4

Why we do this

TfL tries to use open standards as far as possible, and to release as much data as possible. To this end, SVG was chosen:

- As a cross platform solution for displaying and interacting with our schematic maps
- As being more accessible than other technologies, such as Flash and Silverlight
- As being more supportable than closed technologies, like Flash and Silverlight
- As it can be syndicated and re-used as is

Further reading

- SVG standard
- NaPTaN standard
- TransXChange standard

Type:	Standard
Owner:	TfL Online Compliance
Department:	TfL Online

Version History

Version	Date	Summary of changes
1.0	29/11/2013	First issue

Review History

Name	Title	Date	Comments