

DoubleMap embeddable pages

The easiest way to include DoubleMap in your website or app is to embed one of the embeddable DoubleMap pages in an iframe.

For mobile apps, the **/map/mobile** map can be embedded, but keep in mind that it was designed to take up the entire screen of an iPhone display.

For interactive kiosks, such as touchscreen displays, there is **/map/kiosk** that functions like the normal desktop map, except with modified timeout behavior so that it can be displayed for an extended amount of time.

Example: <http://iub.doublemap.com/map/kiosk>

For non-interactive screens and web pages, there is **/map/passive** that displays the map without any interactive controls. There is an optional GET parameter "stop" that takes a numeric stop ID and centers the map on that stop.

Example (Wells Library, IUB): <http://iub.doublemap.com/map/passive?stop=1>

(beta) For web sites that want to display only certain routes on separate pages, there is **/map/embed** which takes an optional "name" GET parameter. The name parameter controls which routes will be visible - any route containing that name as a substring in its public name will be shown.

(beta) For textual ETAs, use **/map/arrivals**. The dropdown at the top can be used to select a stop. The stop can also be specified in a URL hash parameter "stop=[stop_id]".

Example (Wells Library, IUB): <http://iub.doublemap.com/map/arrivals#stop=1>

URL hash parameters

The public live maps support setting the initial map center and zoom level through the URL hash. The parameters "lat", "lon", and "zoom" can be used. If lat and lon are selected, then the map will also display a star marker on those coordinates.

Example (Wells Library, IUB):

<http://iub.doublemap.com/map/passive#lat=39.17105&lon=-86.5169&zoom=17>

DoubleMap v2 API specification

API requests are made to **/map/v2/[resource]** with an HTTP GET request. Data is returned in JSON format. For example, bus data for IU Bloomington would be

<http://iub.doublemap.com/map/v2/buses>.

Authentication

If you get a 403 Forbidden error on any endpoint, that indicates that you need a valid API key to fetch data. Please contact DoubleMap to set up an API key.

The API key is passed as a GET parameter with the name "api". For example, to get buses data using the key "aBcDeFgH", you would need to take "/map/v2/buses" and add "?key=aBcDeFgH", resulting in "/map/v2/buses?key=aBcDeFgH".

Incorrect or missing API key, when an API key is required, will result in HTTP status code 403.

v2/routes

Returns a list of currently-active routes and their pertinent information. This feed needs to be polled no faster than once every 5 minutes.

- name: text
- id: unique integer
- short name: a one or two-character abbreviation of the route
- active: boolean specifying whether the route is currently running
- description: text
- color: in RRGGBB hex format
- stops: as an array of stop IDs in the order they are visited by the route, such as [3,10,5,10]
- path: the outline to be drawn on the map, as an array of latitude and longitude coordinates: [lat1, lon1, lat2, lon2, lat3, lon3, ...]
- fields: an object with the the user-defined fields key-values for this route

Inactive routes: You may pass the GET parameter "inactive=true" to also get routes whose start date is before the current time and end date is after the current time, but may not be currently running. These routes will have active=false.

v2/stops

Returns a list of all stops from the system. This feed only needs to be fetched once per session.

- id
- name
- description
- latitude
- longitude
- buddy: The ID of another stop with which this one shall be paired with. For example, Northbound Q and Southbound Q would be buddies
- fields: an object with the the user-defined fields key-values for this stop

v2/buses

Returns a list of all currently running buses. This feed should be polled no faster than once every 3 seconds.

- id: unique integer
- name: (may not be present for all transit systems) a text name for the vehicle, usually identical to the id
- capacity: nominal maximum number of passengers on the vehicle
- load: estimated current number of passengers on the vehicle
- latitude
- longitude
- heading: the direction of movement, in degrees (0-360). Heading may be 0 if the bus is not moving or if heading is unavailable, depending on the GPS hardware.
- route: the ID of the route that this bus is currently assigned to
- lastStop: the ID of the stop that this bus was most recently at, or its current stop.
- lastUpdate: the Unix timestamp of the last GPS update from the bus
- fields: an object with the the user-defined fields key-values for this bus

Sample response:

```
[
  {
    "id": 636,
    "capacity": 40,
    "load": 19,
    "lat": 39.17948,
    "lon": -86.52667,
    "heading": 0,
    "route": 325,
    "lastStop": 76,
    "lastUpdate": 1393255452
  },
  ...
]
```

v2/announcements

Returns a list of all currently-displayed announcements, sorted with the newest date first. This feed needs to be polled no faster than once every 5 minutes.

- title
- date: The Unix time when the announcement began being displayed (e.g. 1102951291).
- message: The message body of the announcement. Be aware that announcements often have linebreaks, which are represented as the literal string “\r\n”.

v2/eta

Parameters (choose one of the following):

- stop: the stopID for which ETAs are requested
- route: the route ID for which ETAs are requested

Returns a list of buses that will be arriving at that stop in the future, along with an estimate of how long the bus will take. If a route ID is provided, ETAs for all stops along that route are generated, but only for that route. If a stop ID is provided, then ETAs for all routes serving that stop are generated, but only for that stop.

Consider all fields not listed here undocumented and subject to change without notice. This field should be queried when requested by the user or, if constantly polling the same stop, no more often than once per minute.

- etas
 - stopID of the stop
 - etas: an array of ETA objects each representing one future arrival.
 - avg: The predicted amount of time, in minutes, that this bus will take to reach this stop.
 - route_id: the ID of route that this bus is on