

「標準的なバス情報フォーマット」動的データ（GTFSリアルタイム）ガイドライン（案）

GTFSリアルタイムについて

フォーマットの概要

GTFSリアルタイムとは公共交通のリアルタイム情報を格納するためのフォーマットです。GTFSリアルタイムデータは単独では機能せず、GTFS(-JP)データと併せて利用します。

内容

動的データには下記の情報を含めることができます。

対象	要素	設定可能な主な情報
ルート最新情報	TripUpdate	遅延、発着時刻予測、通過
車両位置情報	VehiclePosition	車両の緯度・経度、接近情報、混雑度
運行情報	Alert	見出し、影響（運休、迂回等）、原因（天候、事故等）、URL

これら3つの要素ごとに別ファイルにしても、1つのファイルに格納しても構いません。

データ形式

GTFSリアルタイムはProtocol Buffersという、データ構造が規定されたバイナリ形式をベースとしています。

データ構造は「gtfs-realtime.proto」というテキストファイルにて定義されます。

ベースとするバージョン

- 2019年3月時点の「標準的なバス情報フォーマット」の動的データフォーマットは、GTFSリアルタイム v2.0をベースとします。
 - 日本語文書：Google Developersのサイト
 - <https://developers.google.com/transit/gtfs-realtime/reference/>
 - データ構造：gtfs-realtime.proto
 - <https://developers.google.com/transit/gtfs-realtime/gtfs-realtime.proto>
- GTFSリアルタイム v2.0以降のバージョンがリリースされた場合は、その利用を妨げるものではありません。

設定方法

GTFSリアルタイムを利用する際に推奨する設定方法について記載します。

バスロケーション情報として出力する項目

- 設定方法
 - TripUpdate（ルートの最新情報）とVehiclePosition（車両の現在位置）の両方を出力するようにしてください。
- 理由
 - 経路検索サービス等においてはTripUpdateに含まれる遅延や発着時刻予測等を必要とし、地図表示においてはVehiclePositionに含まれる車両の緯度・経度を必要とするなど、どちらの要素も必要性が高いためです。

遅延時間(delay)の設定方法

- 設定方法
 - 遅延時間（TripUpdate.delay, StopTimeEvent.delay）については、分単位などの丸め処理を行わないようにしてください。（例：143秒を120秒にする）
- 理由
 - 分単位でシンプルに表現するなどの表示の工夫はデータ利用側において行われるべき処理だからです。
 - 利用側にて到着予測などのデータ分析を行う場合に、予め丸め処理がされていると精度が落ちるからです。
- 利用方法
 - リアルタイム情報ではあるもの1分程度の遅延が発生していることを想定した表記の工夫をしてください。
 - 例えば到着予測まで120秒未満となった場合に「まもなく到着」などと表記することで、乗り遅れを防ぐことができると考えられます。

時刻の不確実性(uncertainty)の設定方法

- 設定方法
 - 通過済の停留所に対する、時刻の不確実性（StopTimeEvent.uncertainty）の値はゼロにしてください。
 - 未通過の停留所に対する、時刻の不確実性（StopTimeEvent.uncertainty）の値は秒単位の正の値にしてください。
- 理由
 - 利用側にて通過済かどうかを判定しやすくするためです。

配信方法

データの更新間隔

- 配信方法

- データ更新間隔は30秒以下としてください。これは最低限の既定であり、より短い更新間隔であることが望ましいです。
- データ配信サイト等に、更新間隔を記すようにしてください。
- 理由
 - 更新間隔が長すぎると、到着予測などの誤差が大きくなり、乗り遅れ等が発生してしまうため。
 - 既存バスロケーションシステムの更新間隔の多くが30秒以下であり、最低限準拠可能な設定と考えられるため。
- 利用方法
 - フィードや測定のタイムスタンプ（TripUpdate.timestamp, VehiclePosition.timestamp, FeedHeader.timestamp）が、現在時刻よりも著しく古い場合は、異常値と考えられるため、除去するなどの工夫をしてください。
- 参考情報
 - バスロケーションに関する情報は、「車載器 → バスロケーションシステムサーバ → 情報提供サーバ → 利用者端末」というような経路で車両から利用者に伝送されます。この過程の中で、バッファリング、送受信待ち、データ処理、通信などの時間が積み重なります。
 - システム全体としての遅延時間を抑えるために、データ提供者、データ利用者双方において、無用なバッファリングや送受信待ちをしないよう留意してください。
 - 遅延を抑える必要がある場合には、プッシュ技術を用いるなどの工夫をしてください。

エンドユーザからのデータ取得方法

- 利用方法
 - GTFSリアルタイムデータを用いた情報提供サービスを開発する際は、エンドユーザからデータ配信元に直接データを取りに行くのではなく、間に情報提供サービスのサーバを介すようにしてください。
- 理由
 - データ配信元の負荷を下げるためです。
- 参考情報
 - 2019年3月現在、オープンデータとして配信されているGTFSリアルタイムデータは、いずれも公開サーバ上に配置されております。
 - 下記のように注意事項を記している例があります。
 - 15秒おきに更新されますが、過度なアクセスは行わないようにしてください。

補足事項

車両と時刻表の紐付けについて

- ルート最新情報、車両位置情報については、時刻表との紐付け（trip_id）が事実上必須となっています。
- このため、下記のような場合にデータの作成が困難です。
 - 災害時
 - 通常時のダイヤが機能しておらず、時刻表と紐付けられない。
 - 臨時便

- 時刻表にない臨時バスのため、時刻表と紐付けられない。
- 簡易なバスロケーションシステム
 - 系統、通過停留所、座標などしか保持しておらず、仕業や時刻表と紐付いていない簡易なバスロケーションシステム
- このような状況で出力されるデータの取り扱い方法については、2019年3月時点では今後の検討課題です。