Document Properties

|  |  |  |
| --- | --- | --- |
| **Document/Part Number** | **Document Type** | **Feature Description** |
|  | Feature Document | *tiwiPro Paginated Tables* |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision No.** | **Revisions Made** | **Author** | **Date** |
| **1.0** | Initial Draft | *Colleen Jennings* | *01/21/10* |

# Overview

*This document describes which tables in the current tiwiPro portal need to support pagination and the new hessian methods that are needed to implement this.*

# GUI Pages

The tables below outline the data from Red Flags and Events that are actually displayed in the GUI.

**Red Flags**

***Notifications/Red Flags Tab***

Displays the red flag level and whether or on the red flag generated an alert and event data from the following event types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type ID** | **Event Type** | **Model Class** | **Extra Data Fields** |
| 2 | Note Event | AggressiveDrivingEvent | deltaX  deltaY  deltaZ |
| 93 | Speeding | SpeedingEvent | topSpeed  speedLimit |
| 3 | Seat belt | SeatBeltEvent |  |
| 202 | Unplugged | TamperingEvent |  |
| 213 | Unplugged Asleep | TamperingEvent |  |
| 22 | Low Battery | LowBatteryEvent |  |
| 207 | Low Tiwi Battery | DeviceLowBatteryEvent |  |
| 1 | Full Event | FullEvent |  |
| 209 | Rollover | FullEvent |  |
| 117 | Zone Arrival | ZoneArrivalEvent | zoneName  zonePoints |
| 118 | Zone Departure | ZoneDepartureEvent | zoneName  zonePoints |

**Events**

***Notifications/Safety Tab***

Displays event data from the following event types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type ID** | **Event Type** | **Model Class** | **Extra Data Fields** |
| 2 | Note Event | AggressiveDrivingEvent | deltaX  deltaY  deltaZ |
| 93 | Speeding | SpeedingEvent | topSpeed  speedLimit |
| 3 | Seat belt | SeatBeltEvent |  |

***Notifications/Diagnostics Tab***

Displays event data from the following event types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type ID** | **Event Type** | **Model Class** | **Extra Data Fields** |
| 22 | Low Battery | LowBatteryEvent |  |
| 207 | Low Tiwi Battery | DeviceLowBatteryEvent |  |
| 202 | Unplugged | TamperingEvent |  |
| 208 | Idle | IdleEvent | lowIdle  highIdle |
| 213 | Unplugged Asleep | TamperingEvent |  |

***Notifications/Emergency Tab***

Displays event data from the following event types:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type ID** | **Event Type** | **Model Class** | **Extra Data Fields** |
| 1 | Full Event | FullEvent |  |
| 209 | Rollover | FullEvent |  |

# Hessian

**Red Flags**

***getRedFlagsPage(groupID,start,stop,incForgiven, pageParamMap)***

Fetches a page of Red Flags associated with all drivers in this group and its subgroups,

within the specified timeframe (start, stop) and the specified page criteria

Unknown driver is also included.

Returns a list of redFlagMap, or an Integer error.

***getRedFlagsTotalCount(groupID,start,stop,incForgiven, List<FilterMap>)***

Fetches a count of red flags associated with all drivers in this group and its subgroups,

within the specified timeframe (start, stop) and the specified filters

Unknown driver is also included.

Returns a map[count], or an Integer error.

**redFlagMap:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Filterable** | **Sortable** |
| level | Integer | 0 – none  1 – critical  2 – warning  3 - info | **\*** |  |
| sent | Integer | 0 – false  1 – true  Indicates whether or not an alert was sent for this red flag | **\*** | **\*** |
| event | eventMap |  |  |  |

**Events**

***getDriverEventPage(groupID,start,stop,incForgiven, pageParamMap, typeList)***

Fetches a page of notes associated with all drivers in this group and its subgroups,

within the specified timeframe (start, stop) and the specified page criteria

(optional)typeList would contain a list of note types to be fetched.

Unknown driver is also included.

Returns a list of eventMap, or an Integer error.

***getDriverEventTotalCount(groupID,start,stop,incForgiven, List<FilterMap>, typeList)***

Fetches a count of notes associated with all drivers in this group and its subgroups,

within the specified timeframe (start, stop) and the specified filters

(optional)typeList would contain a list of note types to be fetched.

Unknown driver is also included.

Returns a map[count], or an Integer error.

**eventMap:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Type** | **Description** | **Filterable** | **Sortable** |
| noteID | Long |  |  |  |
| driverID | Integer |  |  |  |
| vehicleID | Integer |  |  |  |
| groupID | Integer |  |  |  |
| type | Integer | Current supported types are:  1,2,3,22,93,202,  207,208,209,213,  117,118 |  |  |
| time | Long |  |  | **\*** |
| lat | Double | Latitude |  |  |
| lng | Double | Longitude |  |  |
| driverName | String | First M Last suffix from driver’s person record -- not in map if unknown driver | **\*** | **\*** |
| tzName | String | Driver’s time zone name or ‘gmt’ if unknown driver (this may change depending on how we handle the unknown driver timezone in general) |  |  |
| vehicleName | String | Name field in vehicle record | **\*** | **\*** |
| groupName | String | Base group name without hierarchy | **\*** | **\*** |
| forgiven | Integer | 1 or 0 |  | **\*** |
| speed | Integer | Mph |  |  |
| *From here down the fields are optional and depend on the event type.* | | | | |
| topSpeed | Integer | Mph |  |  |
| speedLimit | Integer | Mph |  |  |
| deltaX | Integer | Divide by 10 for float |  |  |
| deltaY | Integer | Divide by 10 for float |  |  |
| deltaZ | Integer | Divide by 10 for float |  |  |
| lowIdle | Integer | Seconds for idle events |  |  |
| highIdle | Integer | Seconds for idle events |  |  |
| zoneName | String | Zone Name for zone arrival/departure events |  |  |
| zonePoints | List<LatLng> | List of lat/lng pairs for zone |  |  |

**pageParamMap:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Type** | **Description** |
| firstRowIdx | Integer | first row to fetch (0 based index) |
| lastRowIdx | Integer | last row to fetch (0 based index) |
| sortMap | SortMap | field to sort on and direction to sort (asc/desc) |
| filterMapList | List<FitlerMap> | fields to filter on and filter criteria |

**sortMap:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Type** | **Description** |
| field | String | Field name – must match field name in return object map |
| order | Integer | 0 ascending, 1 descending |

**filterMap:**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Type** | **Description** |
| field | String | Field name – must match field name in return object map |
| filter | Object | Type depends on type stored in ‘field’   * For strings do a sql <field> like ‘%<filter>%> * For numbers do a sql <field> = <filter> |

# Other

1. We are currently doing some filtering of ‘bad’ events on our side. The filter is on speeding events, i.e. type 93. The events that are filtered out have a missing or 0 speedLimit field or a missing or 0 topSpeed field. These will need to be filtered out on the backend.
2. The time zone for unknown driver that comes back in the event map, will need to follow the convention we decide to follow for the rest of the app.
3. Not sure about sorting on fields where we display something in the gui that is different from the value in the backend, for example, red flag level is 0-3, but we display Critical, Information, or Warning. Maybe we can come up with a way to do that. Also will be an issue due to different languages.