Front-End API Documentation

# Object Prototypes

These are definitions of objects used as inputs to some API functions listed below.

### TransitLine

var TransitLine = function(name, time);

These objects are used to describe transit lines in the “Nearest Transit” feature.

**name**: A string that describes the name of the transit line used for “Nearest Transit” feature.

**time**: The expected arrival time of the transit line.

### AvailableTicket

These objects are used to describe each ticket in the “Buy Tickets” panel.

var AvailableTicket = function(name, price);

**name**: A string that describes the name of the ticket available for purchase.

**price**: The price of the ticket.

### FrontQRCaption

This object describes the 3-lined caption underneath the QR code.

var FrontQRCaption = function(ticketText, expirationDate, numGuests);

**ticketText**: A string describing the active ticket that will be used when the QR code is scanned. (Set this to “Base Fare” if no active ticket)

**expirationDate**: A string showing the expiration date of the active ticket. (Set this to null if no active ticket

**numGuests**: An integer describing the number of guests the user has selected. (This shouldn’t be touched by the back-end)

# Global Setters

Description: Modify the max number of guests allowed on a single QR scan. This could be capped further depending on the user's account balance.

Input: An integer upper bound for number of guest spots.

Output: N/A

Error: N/A

function setMaxGuests(newMaxGuests);

Description: Change the base fare (displayed to user and used to calculate guest pass price)

Input: A monetary value

Output: N/A

Error: N/A

function setBaseFare(newBaseFare);

Description: Set the displayed TAP balance in top right of header and in Add Funds under balance.

Input: An integer or float representing the monetary value of their account balance.

Output: N/A

Error: N/A

function setTAPBalance(balanceIntValue);

Description: Set if this user has an active ticket ready to scan.

Input: True if they have a ticket, false if not.

Output: N/A

Error: N/A

function setHasActiveTicket(newHasActiveTicket);

# Home Screen

Description: Set the ticket text in the caption displayed under the front-facing QR code.

Input: Text identifying the ticket being used.

Output: N/A

Error: N/A

function setFrontQRCaptionTicket(ticketText);

Description: Set the expiration date in the caption displayed under the front-facing QR code.

Input: Text identifying the date.

Output: N/A

Error: N/A

function setFrontQRCaptionExpDate(expirationDate);

Description: Set the number of guests in the caption displayed under the front-facing QR code (if 0, line is not shown)

Input: Number of guests on the QR scan.

Output: N/A

Error: N/A

function setFrontQRCaptionGuests(numGuests);

Description: Set the caption displayed under the front-facing QR code.

Input: A **FrontQRCaption** object

Output: N/A

Error: N/A

function setFrontQRCaption(frontQRCaption);

Description: nil (not currently using this)

Input:

Output: N/A

Error: N/A

function setBackQRCaption(backQRCaption);

/Nearest transit lines

Description: Set the first nearest transit line.

Input: A **TransitLine** object

Output: N/A

Error: N/A

function setClosestTransit1(transitLine);

Description: Set the second nearest transit line.

Input: A **TransitLine** object

Output: N/A

Error: N/A

function setClosestTransit2(transitLine);

Description: Define a list of ticket objects to make available in the "Buy Tickets" tab on the home screen.

Input: A list of **AvailableTicket** objects to make available on the front end.

Output: N/A

Error: N/A

function setAvailableTickets(ticketList);

Description: Append a ticket to the "Buy Tickets" tab on the home screen.

Input: An **AvailableTicket** object.

Output: N/A

Error: N/A

function addAvailableTicket(ticket);

Description: Call this when a ticket QR code was successfully processed and payed for by admin application.

Input: Boolean to indicate if TAP balance was used (so front end can deduct it) and the remaining TAP balance (if tapBalanceUsed is true).

Output: N/A

Error: N/A

function processedTicketSuccessful(tapBalanceUsed, remainingBalance);

Description: Call this when a ticket QR code was failed to process.

Input: An error code to identify the issue.

Output: N/A

Error: N/A

function processedTicketFailed(errorCode);