AvokTech API Documentation

V 0.1.7

6/23/2015

<u>Schemas</u>

Account:
```
    id: String,                 unique for each user
    firstName: String,
    lastName: String,
    email: String,              functions as a username (unique)
    password: String,           the value is hashed for security
    phone: String,
    photo: Object,

    reservations: List {        Reservation class
       lotId: String,           referencing the lot id of the reservation
       space: int,              referencing the space of the lot
       start: long,             start time of reservation
       duration: int,           duration (in hours) of reservation
    },
    paid: boolean
    billingAddress: {           Address class
       street: String,
       city: String,
       state: String,
       zipcode: String,
    },
    creditCard: String,         specifics TBD

    lots: Map<String, String>,  maps a lot id to the corresponding role,
                                   including:
                                   owner, manager, patrol, etc.
    roles: List<String>,        list of systems admin roles, including:
                                   super, analytics, onboarding, etc.

Lot:
    id: String,                 unique for each lot
    name: String,
    type: String,               `residential` vs. `parking lot`
    address: {                  Address class
       street: String,
       city: String,
       state: String,
       zipcode: String,
    },
```

```
    location: double[],        (lat, lng) pair
    photo: Object,

    rate: Map<String, double>,  maps the rate type (hourly, daily, etc.) to the
                                    value of the rate
    rateRatio: double,          scale multiplier of rate (default 1.0)

    capacity: int,
    occupied: int,
    reservable: boolean,
    reserveMax: int,            the maximum number of reservable spaces
    spaces: List {              Space class
        reservable: boolean,    if the space is reservable
        photo: Object,
        calendar:               maps the time (by hour) to the associated account
            Map<Long, String>,      id reserved at that time (null if available)
    },

    members: List<String>,      list of account ids with access to the lot

    lotHistoryId: String,       id pointing to the lot's data history

LotHistory:
    id: String,                 unique for each lot
    history:                    list of pairs of time and occupancy
        List<(Long, Integer)>,

RawHistory:
    accountId: String,          account id associated with reservation
    lotId: String,              lot id associated with reservation
    space: int,                 space of lot
    date: int,                  start time of reservation
    duration: int,              duration (in hours) of reservation
    search: String,             search query associated with reservation
    events: List<String>,       list of events near reservation location
```

<u>Endpoints</u>
Required fields are listed under the endpoint.
All `date` fields are represented in milliseconds.

GET
Account
    /listAccounts   *return all accounts*
    /accountById   *returns account with corresponding id*
        accountId
    /accountByEmail   *returns account with corresponding email*
        email

Lots
    /listLots   *returns all lots*
    /lotById   *returns lot with corresponding id*
        id
    /searchNear   *returns lots within 1 mile of the location*
        latitude
        longitude

LotHistory
    /listLotHistory   *returns all lot history*
    /lotHistory   *returns lot history with corresponding id*
        lotHistoryId

RawHistory
    /listRawHistory   *returns all raw history*
    /rawHistoryByAccount   *returns raw history for corresponding account*
        accountId
    /rawHistoryByLot   *returns raw history for corresponding lot*
        lotId
    /rawHistoryByDate   *returns raw history for corresponding date*
        date

POST
Account
    /newAccount   *creates a new account*
        firstName
        lastName
        email
        password
    /addRole   *adds a systems admin role to the account*
        accountId
        role

```
Lots
    /newLot                     creates a new lot
        accountId
        name
        type
        address
        latitude
        longitude
        rateType
        price
        capacity
        reserveMax
    /entered                    increases the occupancy of the lot by 1
        lotId
    /exited                     decreases the occupancy of the lot by 1
        lotId
    /changeReserveMax           changes the number of reservable spots
        lotId
        reserveMax
    /addMember                  adds a member to the lot
        lotId
        email
        role
    /setCalendar                sets the reservation calendar of the lot
        lotId
        startTimes                  a list of the start times for each time block
        durations                   a list of durations for each time block
    /reserve                    adds a reservation for the lot
        lotId
        accountId
        startTimes                  a list of the start times for each time block
        durations                   a list of durations for each time block
        search                      the search query of the reservation
    /clearReservations      clears all reservations of the lot
        lotId
```