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,


    paid: boolean,
    billingAddress: {            Address class
       street: String,
       city: String,
       state: String,
       zipcode: String
    },
    creditCard: String,          TBD


    lots: Map<String, String>,   maps a lot id to the corresponding role,
                                    including:
                                 owner, manager, patrol, etc.
    roles: Set<String>           set 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[],          (lng, lat) 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,
   available: boolean,             if the lot is available (does not include
                                       reservable spaces)

   reservable: boolean,
   reserveMax: int,                the maximum number of reservable spaces


   calendar: long[]                a list of the weekly reservable times

Reservation:
   id: String,                     unique for each reservation
   accountId: String,              the account associated with the reservation
   firstName: String,
   lastName: String,
   lotId: String,                  the lot associated with the reservation
   space: int,                     the space in the lot
   start: long,                    start time
   end: long,                      end time
   duration: int,                  duration of reservation (in hours)
   status: String                  the status of the reservation, including:
                                       active, cancelled, deleted, etc.


LotHistory:
   lotId: String,                  the id of the associated lot
   history: List<{                 list of LotHistoryPoint class
      time: long,
      occupied: int
   }>,

SearchHistory:
   accountId: String,              the id of the associated account
   locations: List<double[]>       list of locations of searches
```

<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
    /getLots   *returns all of the lots of the user*
        accountId

Lot:
    /listLots   *returns all lots*
    /lotById   *returns lot with corresponding id*
        lotId
    /searchNear   *returns lots within 1 mile of the location*
        latitude
        longitude
    /getMembers   *returns the account ids of members of the lot*
        lotId

Reservation:
    /listReservations   *returns all reservations*
    /reservationById   *returns reservation with corresponding id*
        reservationId
    /reservationsByAccountId   *returns reservations for corresponding account*
        accountId
    /reservationsByLotId   *returns reservations for corresponding lot*
        lotId
    /activeReservationsByAccountId*returns active reservations for corresponding*
        accountId   *account*
    /activeReservationsByLotId   *returns active reservations for corresponding lot*
        lotId

LotHistory:
    /listLotHistory   *returns all lot history*
    /lotHistory   *returns lot history for corresponding lot*
        lotId

SearchHistory:

```
/listSearchHistory          returns all search history
/searchHistory              returns search history for corresponding account
    accountId


POST
Account:
/login                      handles login (returns false if incorrect
    email                       password)
    password
/signup                     handles signup and creates a new account
    firstName
    lastName
    email
    password
/addRole                    adds a systems admin role to the account
    accountId
    role
/addLot                     adds a lot and corresponding role to the account
    accountId
    lotId
    role


Lot:
/newLot                     creates a new lot
    accountId
    name
    type
    address
    latitude
    longitude
    capacity
    reserveMax
    calendar                    a list of the times for the calendar
/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
/setCalendar                adds a member to the lot
    lotId
    calendar                    a list of the times for the calendar
```

```
Reservation:
    /reserve                  creates a new reservation
        accountId
        lotId
        start                     start time of reservation
        duration                  duration of reservation (in hours)

SearchHistory:
    /addSearchHistory         adds a new search history point
        accountId
        latitude
        longitude

DELETE
Account:
    /deleteAccount            deletes account with corresponding id
        accountId

Lot:
    /deleteLot                deletes lot with corresponding id
        lotId

LotHistory:
    /deleteAllLotHistory      deletes all lot history
    /deleteLotHistory         deletes lot history for corresponding lot
        lotId

SearchHistory:
    /deleteAllSearchHistory   deletes all search history
    /deleteSearchHistory      deletes search history for corresponding account
        accountId
```