

# AvokTech API Documentation

V 0.1.7

6/23/2015

## Schemas

### Account:

<b>id:</b> String,	<i>unique for each user</i>
<b>firstName:</b> String,	
<b>lastName:</b> String,	
<b>email:</b> String,	<i>functions as a username (unique)</i>
<b>password:</b> String,	<i>the value is hashed for security</i>
<b>phone:</b> String,	
<b>photo:</b> Object,	
<b>paid:</b> boolean,	
<b>billingAddress:</b> {	<u>Address class</u>
<b>street:</b> String,	
<b>city:</b> String,	
<b>state:</b> String,	
<b>zipcode:</b> String	
},	
<b>creditCard:</b> String,	<i>TBD</i>
<b>lots:</b> Map<String, String>,	<i>maps a lot id to the corresponding role, including:</i>
	<i>owner, manager, patrol, etc.</i>
<b>roles:</b> Set<String>	<i>set of systems admin roles, including:</i>
	<i>super, analytics, onboarding, etc.</i>

### Lot:

<b>id:</b> String,	<i>unique for each lot</i>
<b>name:</b> String,	
<b>type:</b> String,	<i>`residential` vs. `parking lot`</i>
<b>address:</b> {	<u>Address class</u>
<b>street:</b> String,	
<b>city:</b> String,	
<b>state:</b> String,	
<b>zipcode:</b> String	
},	
<b>location:</b> double[],	<i>(lng, lat) pair</i>
<b>photo:</b> Object,	

<b>rate:</b> Map<String, double>,	<i>maps the rate type (hourly, daily, etc.) to the value of the rate</i>
<b>rateRatio:</b> double,	<i>scale multiplier of rate (default 1.0)</i>
<b>capacity:</b> int,	
<b>occupied:</b> int,	
<b>available:</b> boolean,	<i>if the lot is available (does not include reservable spaces)</i>
<b>reservable:</b> boolean,	
<b>reserveMax:</b> int,	<i>the maximum number of reservable spaces</i>
<b>calendar:</b> long[]	<i>a list of the weekly reservable times</i>

#### Reservation:

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

#### LotHistory:

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

#### SearchHistory:

<b>accountId:</b> String,	<i>the id of the associated account</i>
<b>locations:</b> List<double[]>	<i>list of locations of searches</i>

## Endpoints

*Required fields are listed under the endpoint.*

*All `date` fields are represented in milliseconds.*

### **GET**

#### Account:

<code>/listAccounts</code>	<i>return all accounts</i>
<code>/accountById</code> <code>accountId</code>	<i>returns account with corresponding id</i>
<code>/accountByEmail</code> <code>email</code>	<i>returns account with corresponding email</i>
<code>/checkAccount</code> <code>email</code>	<i>returns a boolean of whether the account exists</i>
<code>/getLots</code> <code>accountId</code>	<i>returns all of the lots of the user</i>

#### Lot:

<code>/listLots</code>	<i>returns all lots</i>
<code>/lotById</code> <code>lotId</code>	<i>returns lot with corresponding id</i>
<code>/searchNear</code> <code>latitude</code> <code>longitude</code>	<i>returns lots within 1 mile of the location</i>
<code>/getMembers</code> <code>lotId</code>	<i>returns the account ids of members of the lot</i>

#### Reservation:

<code>/listReservations</code>	<i>returns all reservations</i>
<code>/reservationById</code> <code>reservationId</code>	<i>returns reservation with corresponding id</i>
<code>/reservationsByAccountId</code> <code>accountId</code>	<i>returns reservations for corresponding account</i>
<code>/reservationsByLotId</code> <code>lotId</code>	<i>returns reservations for corresponding lot</i>
<code>/activeReservationsByAccountId</code> <code>accountId</code>	<i>returns active reservations for corresponding account</i>
<code>/activeReservationsByLotId</code> <code>lotId</code>	<i>returns active reservations for corresponding lot</i>

#### LotHistory:

<code>/listLotHistory</code>	<i>returns all lot history</i>
<code>/lotHistory</code> <code>lotId</code>	<i>returns lot history for corresponding lot</i>

## SearchHistory:

/listSearchHistory

*returns all search history*

/searchHistory  
accountId

*returns search history for corresponding account*

## POST

### Account:

/login

email

password

*handles login*

/signup

firstName

lastName

email

password

*handles signup and creates a new account*

/addRole

accountId

role

*adds a systems admin role to the account*

/addLot

accountId

lotId

role

*adds a lot and corresponding role to the account*

### Lot:

/newLot

accountId

name

type

address

latitude

longitude

capacity

reserveMax

calendar

*creates a new lot*

*a list of the times for the calendar*

/entered

lotId

*increases the occupancy of the lot by 1*

/exited

lotId

*decreases the occupancy of the lot by 1*

/changeReserveMax

lotId

reserveMax

*changes the number of reservable spots*

/setCalendar

lotId

*adds a member to the lot*

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