

Authentication.

Methods, which require authentication, must have 2 additional parameters:

- **apiEmail**: string, authenticating user email
- **apiSig**: string, request signature

User roles.

Upon successful authentication request received one of following roles:

- **USER** - standard user, controls only his own data. This access level is used by client application.
- **ADMIN** - super user, can control other users' data(profiles, locations, events, etc), also can make locations and events public. This access level is used by public site.
- **OEM USER** - special user, who only can create new users, which will have OEM user as parent. This access can be used by other sites or applications.

Formats.

Each request can have parameter for forcing response format:

- **format**: string. Can be set to "xml" or "json". Default value is "json"

Errors.

Errors are returned as HTTP errors with detailed messages.

API methods.

Method	URL	Parameters	Description	Errors
Find keywords	/keywords/find	<ul style="list-style-type: none">● latitude*: double, -90..+90● longitude*: double, -180..+180● scope: list of strings, separated with comma.	<p>Authentication: optional (required for searching for private locations and events)</p> <p>Required role: none or USER for private locations,</p>	<ul style="list-style-type: none">● invalid arguments● invalid signature● invalid user ID● permission denied

		<p>Allowed values: COUNTRY, STATE, COUNTY, CITY, NEIGHBOURHOOD, POINT_OF_INTEREST. All locations are returned if parameter is not specified</p> <ul style="list-style-type: none"> ● date: date in format of “YYYY-MM-DD HH:mm:ss”. Used for finding events ● withEvents: boolean. Event keywords are returned if set to “true”. Default value: “true” ● userEmail: string. If specified and request has valid signature, then private locations and events owned by this user will be included in search. Otherwise only public locations and events will be scanned. ● appId: string, identifies client application or 3rd party developer. Can be used for tracking activity. 	<p>events</p> <p>Returns: List of keywords for locations, which cover specified point. If event keywords requested - they are matched by location keywords and date and appended to the returned list. If “userEmail” parameter is specified, then request must have valid signature. If “appId” is not specified, keyword www.ipictdb.com test” will be appended to resulting keywords.</p>	
Create location	/locations/create	<ul style="list-style-type: none"> ● top*: double, Northern boundary latitude, - 90..+90 ● bottom*: double, Southern boundary latitude, -90..+90 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Creates a private location for the user. Location type</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● permission denied

		<ul style="list-style-type: none"> ● left*: double, Western boundary longitude, -180..+180 ● right*: double, Eastern boundary longitude, -180..+180 ● keywords*: list of keywords, separated with comma ● userEmail*: location will be owned by specified user. ● makePublic: boolean. Default value - "false". If set to "true", will request the created location to become public 	<p>is "POINT OF INTEREST". A location can be submitted to review to become public with flag "makePublic". If location with the same boundaries already exists (private for this user) - new location keywords are added to the existing location keywords. New location won't be created in this case, flag "makePublic" will be ignored.</p> <p>Returns: created or existing location</p>	
Update location	/locations/update	<ul style="list-style-type: none"> ● locationId*: integer, location ID ● top*: double, Northern boundary latitude, -90..+90 ● bottom*: double, Southern boundary latitude, -90..+90 ● left*: double, Western boundary longitude, -180..+180 ● right*: double, Eastern boundary longitude, -180..+180 ● keywords*: list of keywords, separated with comma 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Updates existing user private location. If new location boundaries match another existing location (private for this user), keywords will be appended to that location. Location with specified ID will be removed. Flag "makePublic" will be ignored in this case.</p> <p>Returns: updated or existing location</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● invalid location ID ● permission denied

		<ul style="list-style-type: none"> ● userEmail*: location owner. ● makePublic: boolean. Default value - "false". If set to "true", will request the created location to become public 		
Delete location	/locations/delete	<ul style="list-style-type: none"> ● locationId*: integer, location ID ● userEmail*: location owner. Parameter used only with ADMIN access 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Deletes existing user private location</p> <p>Returns: removed location</p>	<ul style="list-style-type: none"> ● invalid signature ● invalid user ID ● invalid location ID ● permission denied
List locations	/locations/list	<ul style="list-style-type: none"> ● pageNumber*: integer. 1..N. ● pageSize*: integer, 1..N. ● userEmail*: locations owner. Parameter used only with ADMIN access 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Finds user's private locations. Parameters "pageNumber" and "pageSize" control number of returned locations.</p> <p>Returns: list of matching locations and information about total number of available locations.</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● permission denied

Create event	/events/create	<ul style="list-style-type: none"> ● startDate*: date in format, YYYY-MM-DD or YYYY-MM-DD HH:mm:ss ● endDate*: date in format, YYYY-MM-DD or YYYY-MM-DD HH:mm:ss ● locationKeywords*: list of strings, comma-separated ● eventKeywords*: list of strings, comma-separated ● userEmail*: event owner. ● makePublic: boolean. Default value - "false". If set to "true", will request the created event to become public 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Creates a private event for the user. Every event has location keywords to bind events to locations. If time is not specified, it is considered to be 00:00:00 for start date and 23:59:59 for end date. If private event with the same dates and location keywords exists, event keywords are appended to existing event. Flag "makePublic" will be ignored in this case.</p> <p>Returns: created or existing event</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● permission denied
Update event	/events/update	<ul style="list-style-type: none"> ● eventID*: event ID ● startDate*: date in format, YYYY-MM-DD or YYYY-MM-DD HH:mm:ss ● endDate*: date in format, YYYY-MM-DD or YYYY-MM-DD HH:mm:ss ● locationKeywords*: list of strings, comma-separated ● eventKeywords*: list of strings, comma-separated ● userEmail*: event 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Updates existing private event. If updated events matches another event by dates and location keywords, event keywords will be appended to the existing event. Old event with ID eventID will be removed. Flag "makePublic" will be ignored in this case.</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● invalid event ID ● permission denied

		<p>owner.</p> <ul style="list-style-type: none"> ● makePublic: boolean. If set to “true”, will request the created event to become public 	<p>Returns: updated or existing event</p>	
Delete event	/events/delete	<ul style="list-style-type: none"> ● eventID*: event ID ● userEmail*: event owner. Parameter used only with ADMIN access 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Deletes user private event.</p> <p>Returns: removed event</p>	<ul style="list-style-type: none"> ● invalid signature ● invalid user ID ● invalid event ID ● permission denied
List events	/events/list	<ul style="list-style-type: none"> ● pageNumber*: integer. 1..N. ● pageSize*: integer, 1..N. ● userEmail*: events owner. Parameter used only with ADMIN access 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Finds private events. Parameters “pageNumber” and “pageSize” control number of returned events.</p> <p>Returns: list of matching events and information about total number of available events.</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● permission denied
Create user account	/users/create	<ul style="list-style-type: none"> ● email*: user email. Must be unique ● password*: user password. Must match specific password rules 	<p>Authentication: not required</p> <p>Required role: none</p>	<ul style="list-style-type: none"> ● invalid arguments ● email is not unique ● password doesn't match rules

			<p>Creates new user. Rate limiting will be implemented for requests from the same IP to avoid spamming</p> <p>Returns: created user email</p>	<ul style="list-style-type: none"> ● permission denied
Get user profile	/users/get	<ul style="list-style-type: none"> ● email*: user email. 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Returns: user profile data.</p>	<ul style="list-style-type: none"> ● invalid signature ● invalid user ID ● permission denied
Update user profile	/users/update	<ul style="list-style-type: none"> ● email*: string, user email. ● name*: string, user name ● phone: string, user phone number ● canCreateUsers: boolean flag. Default value - "false" 	<p>Authentication: required</p> <p>Required role: ADMIN</p> <p>Returns: updated user profile data.</p>	<ul style="list-style-type: none"> ● invalid arguments ● invalid signature ● invalid user ID ● password doesn't match rules ● permission denied