**SoundPlanes API**

**GET /user/:id**

Returns information about an user.

**Path Parameters**

Id: An unique user identifier

**Response Format**

On success the HTTP status code in the response header is 200 OK.

{

Id: {id},

name: {name},

country: {country},

state: {state}

}

On error the HTTP status code is 404 Not Found if the given id doesn’t exist.

---------------------------------------------------------------------------------------------------

**POST /user**

Creates an user.

**Body Parameters**

{

name: {name},

country: {country},

state: {state}

}

**Response** **Format**

On success the HTTP status code in the response header is 201 Created.

{

Id: {id},

location: {

lat: {latitude},

lon: {longitude}

}

}

On error the HTTP status code is 400 Bad Request if the body parameters aren’t correct.

---------------------------------------------------------------------------------------------------

**PUT /user/:id**

Modifies information of a given user.

**Header Fields**

Authorization: Required. A valid access token to modify the given user.

**Path Parameters**

Id: An unique user identifier

**Body Parameters**

{

name: {name}

}

**Response** **Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given id doesn’t exist, 401 Unauthorized if Access token isn’t valid.

---------------------------------------------------------------------------------------------------

**DELETE /user/:id**

Deletes a given user.

**Header Fields**

Authorization: Required. A valid access token to delete the given user.

**Path Parameters**

Id: An unique user identifier

**Response** **Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given id doesn’t exist, 401 Unauthorized if Access token isn’t valid.

---------------------------------------------------------------------------------------------------

**GET /user/:id/playlist**

Returns the playlist of the given user.

**Path Parameters**

Id: An unique user identifier

**Response** **Format**

On success the HTTP status code in the response header is 200 OK.

An array of songs

{

Playlist: [

{

Id: {songId},

name: {songName}

artist: {artistName}

},

…

]

}

On error the HTTP status code is 404 Not Found if the given id doesn’t exist, or the player doesn’t have a playlist.

---------------------------------------------------------------------------------------------------

**POST /user/:id/playlist**

Creates a playlist for the user

**Header Fields**

Authorization: Required. A valid access token to create a playlist for the given user.

**Path Parameters**

Id: An unique user identifier

**Body Parameters**

An array of songs.

{

Playlist: [

{

Id: {songId},

name: {songName}

artist: {artistName}

},

…

]

}

**Response Format**

On success the HTTP status code in the response header is 201 Created.

On error the HTTP status code is 404 Not Found if the given id doesn’t exist, or 400 Bad Request if the body parameters are not correct.

---------------------------------------------------------------------------------------------------

**PUT /user/:id/playlist**

Add songs to an user playlist

**Header Fields**

Authorization: Required. A valid access token to modify a playlist for the given user.

**Path Parameters**

Id: An unique user identifier

**Body Parameters**

An array of songs.

{

Playlist: [

{

Id: {songId},

name: {songName}

artist: {artistName}

},

…

]

}

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given id or playlist doesn’t exist, or 400 Bad Request if the body parameters are not correct.

---------------------------------------------------------------------------------------------------

**DELETE /user/:id/playlist**

Removes an user’s playlist

**Header Fields**

Authorization: Required. A valid access token to delete a playlist for the given user.

**Path Parameters**

Id: An unique user identifier

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given id or playlist doesn’t exist, or 401 Unauthorized if the Access token isn’t valid.

---------------------------------------------------------------------------------------------------

**GET /user/:id/airport**

Returns information of an user’s airport.

**Path Parameters**

Id: An unique user identifier

**Response Format**

On success the HTTP status code in the response header is 200 OK.

{

name: {airportName},

location: {

lat: {latitude},

lon: {longitude{

}

}

On error the HTTP status code is 404 Not Found if the given id or airport doesn’t exist.

**POST /user/:id/airport**

Creates an airport for the given user.

**Header Fields**

Authorization: Required. A valid access token to create an airport for the given user.

**Path Parameters**

Id: An unique user identifier

**Response Format**

On success the HTTP status code in the response header is 201 Created.

On error the HTTP status code is 404 Not Found if the given id doesn’t exist, or 401 Unauthorized if the Access token isn’t valid.

---------------------------------------------------------------------------------------------------

**PUT /user/:id/airport**

Change an airport information for the given user.

**Header Fields**

Authorization: Required. A valid access token to modify an airport for the given user.

**Path Parameters**

Id: An unique user identifier

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given id or airport doesn’t exist, or 401 Unauthorized if the Access token isn’t valid.

---------------------------------------------------------------------------------------------------

**DELETE /user/:id/airport**

Deletes an airport for the given user.

**Header Fields**

Authorization: Required. A valid access token to delete an airport for the given user.

**Path Parameters**

Id: An unique user identifier

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given id or airport doesn’t exist, or 401 Unauthorized if the Access token isn’t valid

---------------------------------------------------------------------------------------------------

**GET /country/:name**

Returns information about the users, songs playing…

**Path Parameters**

name: An unique country identifier in Alpha-3 code (ISO 3166)

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given name doesn’t exist.

---------------------------------------------------------------------------------------------------

**GET /country/:name/top**

Returns top songs of a given country.

**Path Parameters**

name: An unique country identifier in Alpha-3 code (ISO 3166)

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given name doesn’t exist.

---------------------------------------------------------------------------------------------------

**GET /country/:name/airport**

Returns information about the international airport of a country.

**Path Parameters**

name: An unique country identifier in Alpha-3 code (ISO 3166)

**Response Format**

On success the HTTP status code in the response header is 200 OK.

On error the HTTP status code is 404 Not Found if the given name doesn’t exist.

---------------------------------------------------------------------------------------------------