Users API v1.1

Common provisions

Terminology

The terminology of RFC 2119 (specifically **must**, **should**, **may** and their negatives) applies. The word **will**, when applied to the Hardware Service API ("the API"), has the same meaning as **must**.

Protocol

The API supports communication over HTTPS only.

Encoding

The API supports communication using JSON encoding only. The client **must** submit the headers `Content-Type: application/json` and `Accept: application/json` for all requests. Failure to do so **will** result in a `415 Unsupported Media Type` response. The API **will** include the header `Content-Type: application/json` with its response.

Authentication

Unless otherwise specified, the endpoints in the API are authenticated by a JWT bearer token. Two token sources are accepted:

- Tokens acquired from the `user/login` endpoint;
- Tokens returned from the Website service's `/token` endpoint as the value of the `access_token` property. See the documentation at /Website/README.md#Oauth.

The client **must** submit the header `Authorization: <JWT>` with all requests. Failure to do so, or submitting an invalid or expired JWT, **will** result in a `401 Unauthorized` response.

General responses

In addition to the AWS API Gateway responses and the specific responses for each endpoint, the server **may** respond with one of the following HTTP responses:

- `400 Bad Request` with `Status` header equal to `InvalidSchema`, if the JSON body of the request does not match the requirements of the endpoint.
- `404 Unknown` with `Status` header equal to `UnknownEndpoint`, if an invalid endpoint was requested.

Schema

Simple

The following simple types **may** be used in responses:

- `string`, `number`, `integer`: as defined in the JSON Schema standard.

- `Uuid`: a `string` matching the regular expression `^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$`, that is, the string representation of an RFC 4122 UUID.
- `Date`: a `string` matching the regular expression `/\d{4}-\d{2}-\d{2}/` and representing a date in ISO 8601 'YYYY-MM-DD' format.
- `Datetime`: a `string` matching the regular expression `/\d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}Z/` and representing a date and time in full ISO 8601 format.
- `Gender`: one of the `strings` `male`, `female`, `mixed` or `other`.
- `Phonenumber`: a `string` matching the regular expression `TODO`, being a string representation of a phone number

User

A `User` object **must** have the following schema:

```
{
    "biometric_data": {
        "gender": Gender,
        "height": {
            "ft_in": [integer, integer],
            "m": number
        },
        "mass": {
            "lb": number,
            "kg": number
        }
    },
    "created_date": Datetime,
    "deleted_date": Datetime,
    "id": Uuid,
    "organization_id": Uuid,
    "personal_data": {
        "birth_date": Date,
        "email": string,
        "first_name": string,
        "last_name": string,
        "phone_number": Phonenumber,
        "position": string
    },
    "role": enum,
    "updated_date": Datetime,
    "training_status": enum,
    "teams": [ Team, ... ],
    "training_groups": [ TrainingGroup, ... ]
}
```

The following constraints **will** apply:

- `personal_data.gender` **must not** be `mixed`.
- `role` **must** be one of the strings `athlete`, `manager`, `admin`, `super_admin`, `biometrix_admin`, `subject` or `consumer`.

- `training_status` **must** be one of the strings `full_volume`, `returning_to_full_volume` or `inactive`.

Team

A `Team` object **must** have the following schema:

```
{
    "id": Uuid,
    "name": string,
    "organization_id": Uuid,
    "created_date": Datetime,
    "updated_date": Datetime,
    "athlete_subscriptions": integer,
    "athlete_manager_subscriptions": integer,
    "gender": Gender,
    "sport_id": Uuid
}
```

The following constraints **will** apply:

- `athlete_subscriptions` and `athlete_subscriptions` **must** be a non-negative integer.

TrainingGroup

A `TrainingGroup` object **must** have the following schema:

```
{
    "id": Uuid
}
```

The following constraints **will** apply:

- `device_type` **will** be one of the strings `accessory` or `sensor`.

Endpoints

User

Login

This endpoint can be called by a client, once registered, to acquire credentials with which to access other endpoints. The user **must** have been registered prior to requesting this endpoint.

Query String

The client **must** submit a request to the endpoint `/user/login`.

Request

The client **must** submit a request body containing a JSON object with the following schema:

```
{
    "email": String,
    "password": String
}
```

- `password` **must** be a string containing 8 or more characters, with no leading or trailing spaces.

```
POST /users/1_1/user/login HTTP/1.1
Host: apis.env.fathomai.com
Content-Type: application/json

{
    "email": "fathomai@example.com",
    "password": "ffqkjhrqdkha2"
}
```

Authentication is not required for this endpoint.

Responses

If the authentication was successful, the Service **will** respond with HTTP Status `200 OK`, and with a body with the following syntax:

```
{
    "authorization": {
        "expires": String,
        "jwt": String,
        "session_token": String
    },
    "user": User
}
```

- `authorization.jwt` **will** be a String forming a valid JWT Bearer Token.
- `authorization.expires` **will** be a Datetime, representing the time at which the JWT will expire.
- `user` **will** be the User uniquely identified by the supplied email address.

Example response:

```
{
    "authorization": {
        "expires": "2018-04-06:31:19Z",
        "jwt": "eyJraWQ...ajBc4VQ",
        "session_token":
"bf652a1a90f4a3edc0887716c6bc309939a5bf87d1274ad624de0374e3ad1b1e"
    },
    "user": {
        "biometric_data": {
```

```
            "gender": "male"",
            "height": {
                "ft_in": [5, 11],
                "m": 1.80
            },
            "mass": {
                "lb": 140,
                "kg": 63.50
            }
        },
        "created_date": "2018-04-05T23:49:52Z",
        "deleted_date": null,
        "id": "5b7f234f-d25d-4799-8a14-23bc909b61dd",
        "organization_id": "f62fbd5b-aafc-436f-b358-be2b34e1fe58",
        "personal_data": {
            "birth_date": "2000-01-01",
            "email": "fathomai@example.com",
            "first_name": "John",
            "last_name": "Smith",
            "phone_number": "9191234567",
            "position": "Quarterback"
        },
        "role": "athlete",
        "updated_date": "2018-04-06T23:49:52Z",
        "training_status": "full_volume",
        "teams": [
            {
                "id": "f87e1deb-f022-4223-acaa-4926b6094343",
                "name": "Womens Soccer",
                "organization_id": "f62fbd5b-aafc-436f-b358-be2b34e1fe58",
                "created_at": "2017-08-15T07:55:39Z",
                "updated_at": "2017-10-16T16:11:34Z",
                "athlete_subscriptions": 10,
                "athlete_manager_subscriptions": 10,
                "gender": "female",
                "sport_id": "8534c4ea-4b37-40a0-a037-cad00cf03f74"
            }
        ],
        "training_groups": [
            {
                "id": "7fa6e26c-e2f5-46e4-bf07-79bfb12ac840"
            }
        ]
    }
}
```


If the authentication was not successful, the Service **will** respond with one of the following HTTP Status codes:

- `400 Unauthorized`, if the credentials supplied are not valid
- `404 Not Found`, if no user with those credentials was found and the Service is willing to reveal that fact (note that the Service **may** choose to reply with `400 Unauthorized` in this instance, for security reasons).

Get

This endpoint allows the client to get information about a user, including the current user.

Query String

The client **must** submit a request to the endpoint `/user/{user_id}`, where `user_id` **must** be either a Uuid or the string `me`. If `me` is submitted, the returned results **will** be identified from the authorization header JWT.

The request method **must** be `GET`.

Request

This method takes no request body.

Example request:

```
GET /users/1_1/user/me HTTP/1.1
Host: apis.env.fathomai.com
Content-Type: application/json
Authorization: eyJraWQ...ajBc4VQ
```

Response

The Service **will** respond with an HTTP status of `200 OK` and a body with the following syntax:

```
{
    "user": User
}
```

Authorise

This endpoint can be called by a client, which has previously logged in, renew its JWT authorisation token. The user **must** have been registered prior to requesting this endpoint.

Query String

The client **must** submit a request to the endpoint `/user/{user_id}/authorise`.

Request

The client **must** submit a request body containing a JSON object with the following schema:

```
{
    "session_token": String
}
```

- `session_token` **must** be a session token string previously returned from a call to `login`.

```
POST /users/1_1/user/e8514489-8de9-47e0-b3d5-b15da244783f/authorise HTTP/1.1
Host: apis.env.fathomai.com
Content-Type: application/json

{
    "session_token":
"bf652a1a90f4a3edc0887716c6bc309939a5bf87d1274ad624de0374e3ad1b1e"
}
```

Authentication is not required for this endpoint.

Responses

If the authentication was successful, the Service **will** respond with HTTP Status `200 OK`, and with a body with the following syntax:

```
{
    "authorization": {
        "expires": String,
        "jwt": String,
    }
}
```

- `authorization.jwt` **will** be a String forming a valid JWT Bearer Token.
- `authorization.expires` **will** be a Datetime, representing the time at which the JWT will expire.

Example response:

```
{
    "authorization": {
        "expires": "2018-04-06:31:19Z",
        "jwt": "eyJraWQ...ajBc4VQ"
    }
}
```

If the authentication was not successful, the Service **will** respond with one of the following HTTP Status codes:

- `400 Unauthorized`, if the session token was not (or is no longer) valid
- `404 Not Found`, if no user with that uuid was found.

Logout

This endpoint can be called by a client, which has previously logged in, to log out.

Query String

The client **must** submit a request to the endpoint `/user/{user_id}/logout`.

Request

This endpoint takes no request body.

```
POST /users/1_1/user/e8514489-8de9-47e0-b3d5-b15da244783f/logout HTTP/1.1
Host: apis.env.fathomai.com
Content-Type: application/json
```

Responses

If the logout was successful, the Service **will** respond with HTTP Status `200 OK`, and with an empty body.

Notify

This endpoint can be called by a client to send a push notification to all the devices affiliated to a given user.

Query String

The client **must** submit a request to the endpoint `/user/{user_id}/notify`.

Request

The client **must** submit a request body containing a JSON object with the following schema:

```
{
    "message": String
}
```

- `message` is the text which **will** be displayed to the user in push notification.

```
POST /users/1_1/user/e8514489-8de9-47e0-b3d5-b15da244783f/notify HTTP/1.1
Host: apis.env.fathomai.com
Content-Type: application/json

{
    "message": "Hello world!"
}
```

Responses

If the request was successful, the Service **will** respond with HTTP Status `200 OK`, and with a body with the following syntax:

```
{
    Uuid: {
        "message": String,
        "success": Bool
    },
    ...
}
```

Where each `Uuid` is the id of a `Device` affiliated to the user, and `success` indicates whether the message was successfully sent to that device.

Example response:

```
{
    "0758c0c7-bf2f-4ec7-89a7-926a7eb4ba4c": {
        "message": "Endpoint disabled for device
0758c0c7-bf2f-4ec7-89a7-926a7eb4ba4c",
        "success": false
    },
    "e8514489-8de9-47e0-b3d5-b15da244783f": {
        "message": "Success",
        "success": true
    }
}
```

If the request was not successful, the Service **may** respond with one of the following HTTP Status codes:

- `429 Too Many Requests`, if an identical message has already been sent to the user within the throttling period.

Device

Register

This endpoint can be called by a client to register a new mobile device.

Query String

The client **must** submit a request to the endpoint `/device/{device_id}`. The request method **must** be `POST`.

The `device_id` **must** be a `Uuid`, and **must** be unique to the device.

Request

The client **must** submit a request body containing a JSON object with the following schema:

```
{
    "device_type": String,
    "push_notifications": {
        "token": String,
        "enabled" Boolean
    }
}
```

- `device_type` **must** be either `ios` or `android`.
- `push_notifications` is optional.

```
POST /users/1_1/device/e8514489-8de9-47e0-b3d5-b15da244783f HTTP/1.1
Host: apis.env.fathomai.com
Content-Type: application/json
Authorization: eyJraWQ...ajBc4VQ

{
    "device_type": "ios",
    "push_notifications": {
        "token": "ABCDEF",
        "enabled" true
    }
}
```

Responses

If the registration was successful, the Service **will** respond with HTTP Status `201 Created`, and with a body with the following syntax:

```
{
    "certificate": {
        "id": String,
        "pem": String,
        "private_key": String,
        "public_key": String
    },
    "device": {
        "id": Uuid,
        "thing_id": Uuid,
        "type": String
    }
}
```

- `certificate.pem`, `certificate.private_key` and `certificate.public_key` **will** be Strings representing an RSA keypair in PEM format.

Example response:

```
{
    "authorization": {
```

```
        "expires": "2018-04-06:31:19Z",
        "jwt": "eyJraWQ...ajBc4VQ"
    }
}
```

If the authentication was not successful, the Service **will** respond with one of the following HTTP Status codes:

- `400 Unauthorized`, if the session token was not (or is no longer) valid
- `404 Not Found`, if no user with that uuid was found.