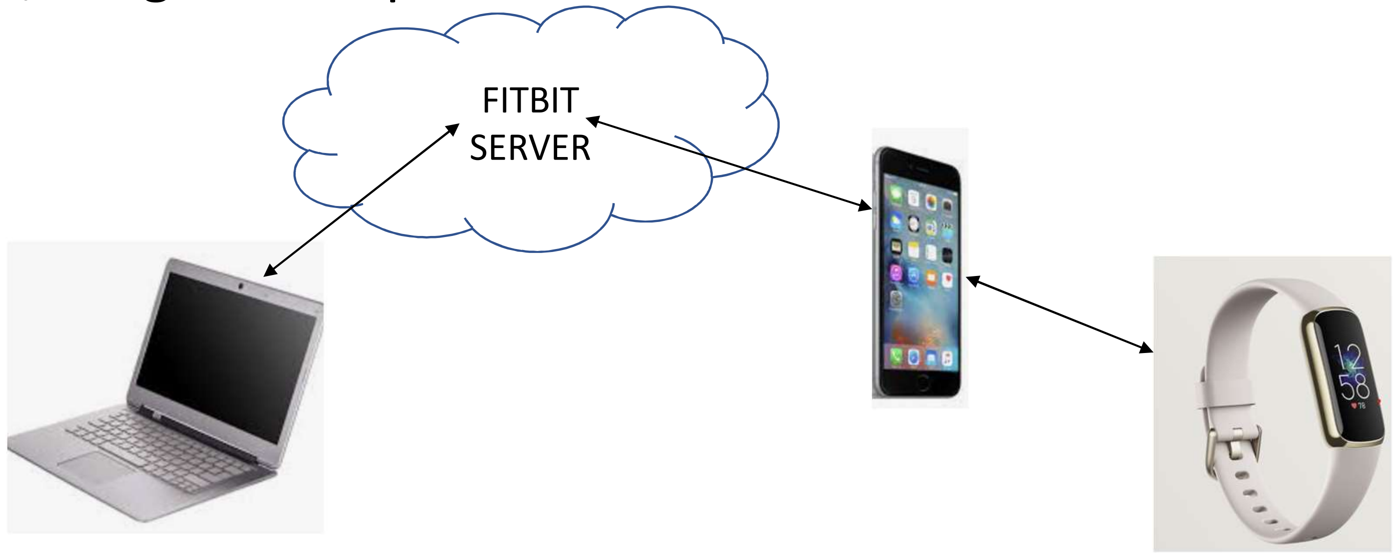
UNIVERSITÀ DEL PIEMONTE ORIENTALE

Autenticazione e autorizzazione con Oauth2 un esempio d’uso: FITBIT

Uso di JSON Web Token per gestire l’accesso alle risorse

Un esempio di applicazione: accedere ai dati del profilo fitbit

- La fitness band Fitbit si collega via bluetooth al cellulare, ma le informazioni vengono mandate con una certa periodicità al cloud di fitbit. Le informazioni pur essendo presenti nella fitness band e sul cellulare, vengono sempre lette dal server su cloud

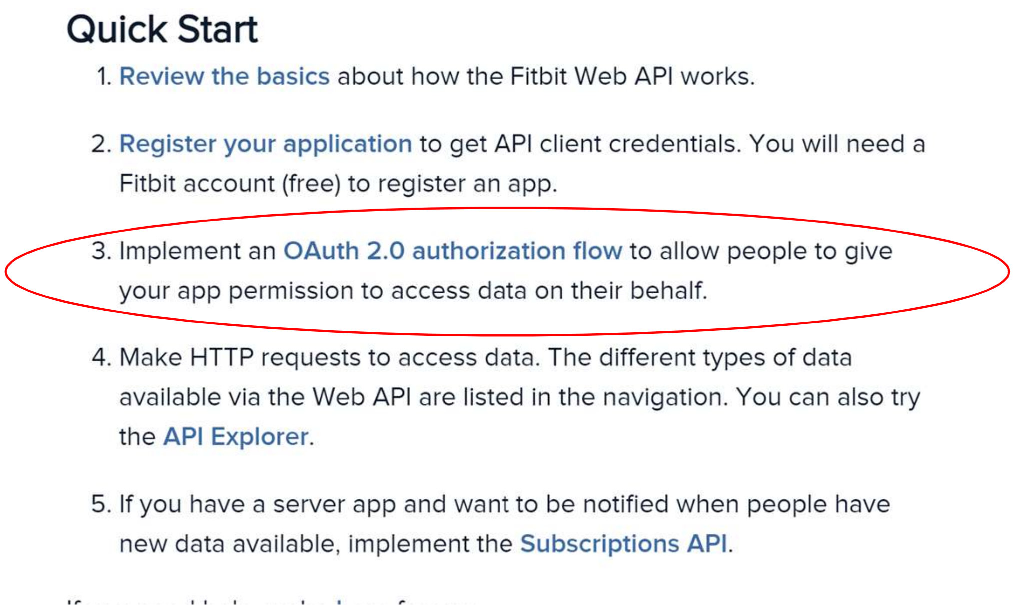


UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

Lettura dei dati dell'utente da programma

Si possono sviluppare applicazioni che si connettono al server fitbit utilizzando le API che mette a disposizione

https://dev.fitbit.com/build/reference/web-api



Quick Start

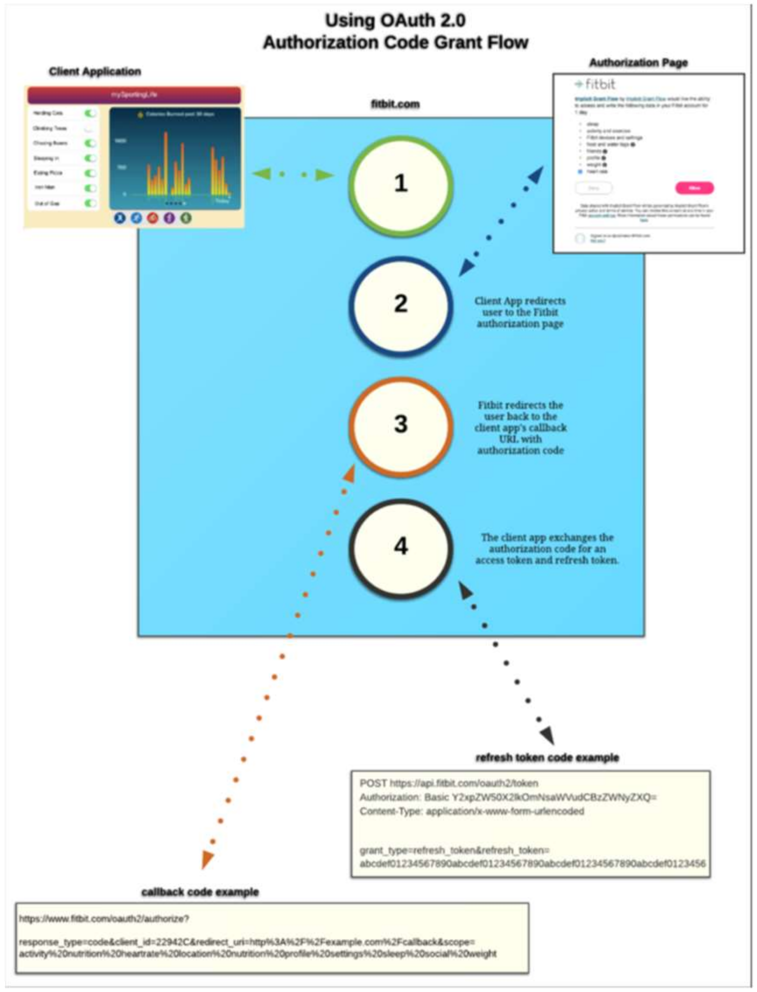
1. Review the basics about how the Fitbit Web API works.
2. Register your application to get API client credentials. You will need a Fitbit account (free) to register an app.
3. Implement an OAuth 2.0 authorization flow to allow people to give your app permission to access data on their behalf.
4. Make HTTP requests to access data. The different types of data available via the Web API are listed in the navigation. You can also try the API Explorer.
5. If you have a server app and want to be notified when people have new data available, implement the Subscriptions API.

UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

The Authorization Code Grant Flow has the following steps:

1. Your application presents the Fitbit authorization page to the user by calling the "authorize" endpoint. See Authorization Page below.
2. The user consents to share their Fitbit data with your application by enabling some or all scopes. Upon the user's initial consent, Fitbit redirects the user back to your application using the redirect or callback URL with an authorization code as a URL parameter.
3. Your application exchanges the authorization code it receives from step 2 for an access token and refresh token pair. See Access Token Request below. Your application should store the access token and refresh token.
4. The application uses the access token to execute API calls. The refresh token is used to renew the access token when it expires without having to re-prompt the user.

Dal sito https://dev.fitbit.com/build/reference/web-api/oauth2/#authorization-code-grant-flow


Using OAuth 2.0
Authorization Code Grant Flow
Client Application
fitbit.com
Authorization Page
fitbit
1
2
Client App redirects user to the Fitbit authorization page
3
Fitbit redirects the user back to the client app's callback URL with authorization code
4
The client app exchanges the authorization code for an access token and refresh token.
refresh token code example
callback code example

refresh token code example

```
POST https://api.fitbit.com/oauth2/token
Authorization: Basic Y2xpZW50X2lkOmNsaWVudCBzZWNyZXQ=
Content-Type: application/x-www-form-urlencoded


grant_type=refresh_token&refresh_token=
abcdef01234567890abcdef01234567890abcdef01234567890abcdef0123456
```

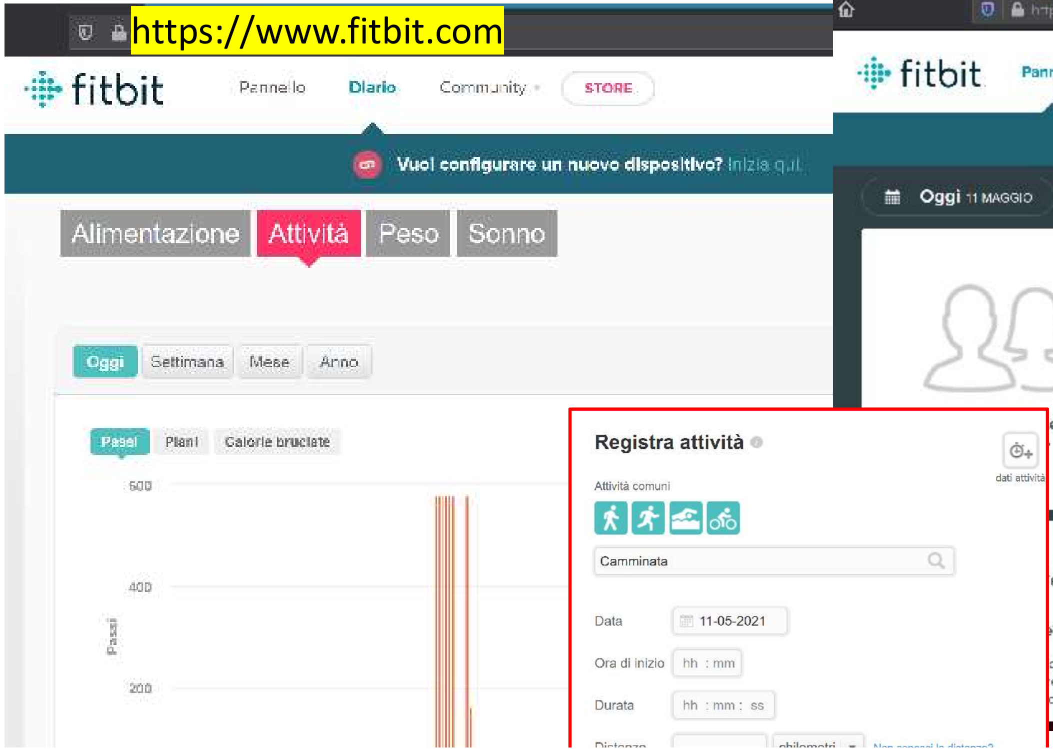
callback code example

```
https://www.fitbit.com/oauth2/authorize?

response_type=code&client_id=22942C&redirect_uri=http%3A%2F%2Fexample.com%2Fcallback&scope=
activity%20nutrition%20heartrate%20location%20nutrition%20profile%20settings%20sleep%20social%20weight
```

UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

Prima di registrare la nuova app (la client application) …

Occorre disporre di un account su fitbit.com (potete crearne uno locale oppure registrarvi con un utente google). Una volta creato l'account potete accedere alla vostra dashboard dove sono visualizzabili i dati provenienti dal vostro fitness tracker/smartwatch o dati inseriti manualmente.


https://www.fitbit.com

Registra attività

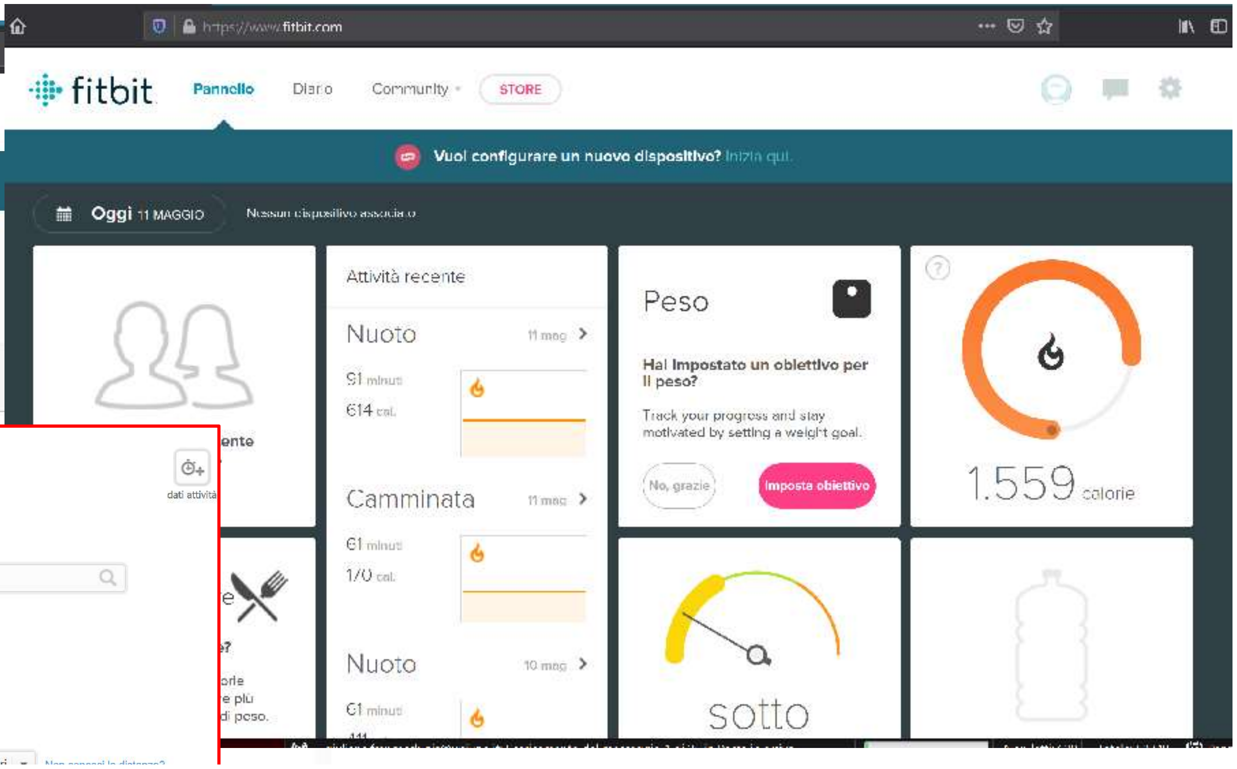
Attività comuni

Camminata

Data 11-05-2021

Ora di inizio hh : mm

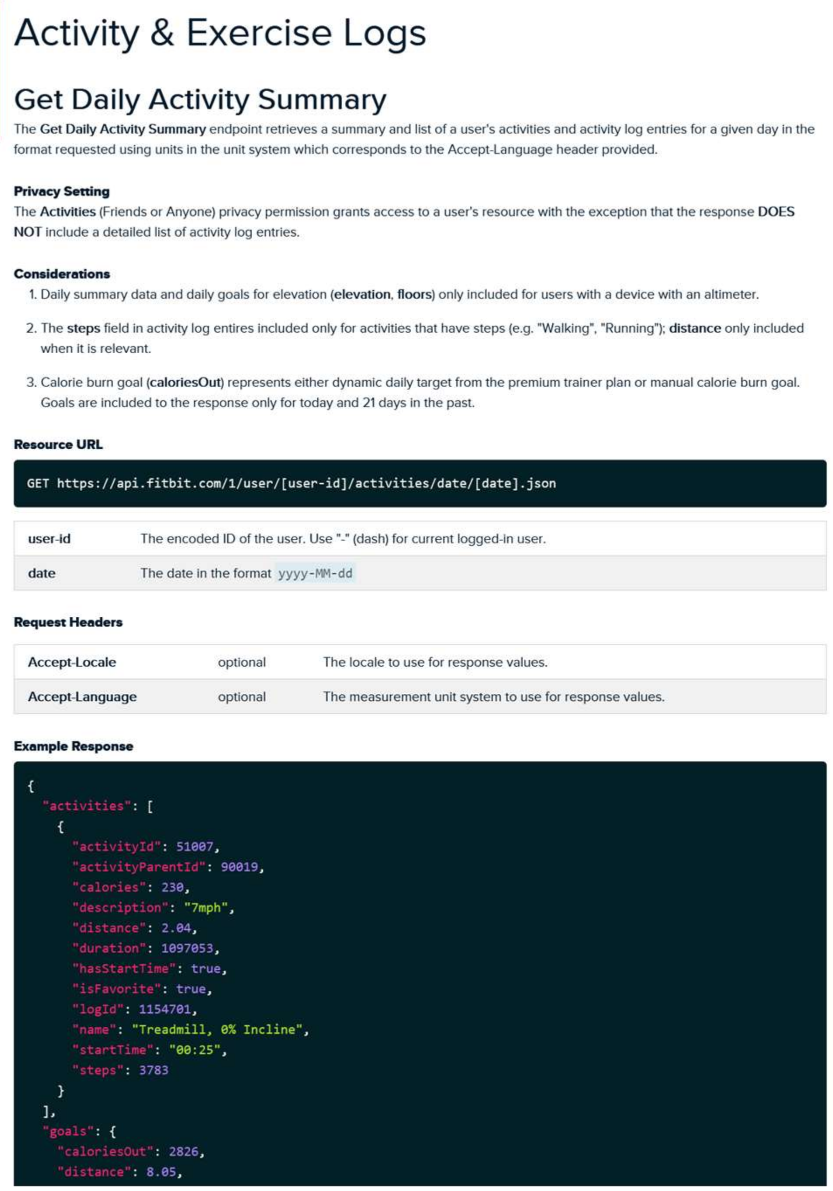
Durata hh : mm : ss



Le Web-API di fitbit.com

Invieremo al server fitbit.com delle GET per prelevare le attività dell'utente [user-id] ad una certa data (nel formato AAAA-MM-GG)

Dovremo inviare nell'header un token dopo averlo prelevato dall'authorization server di fitbit (presentando l'ID di una Client App registrata sullo stesso, e un «secret» condiviso tra Client App e l'authorization server)



Activity & Exercise Logs

Get Daily Activity Summary

The **Get Daily Activity Summary** endpoint retrieves a summary and list of a user's activities and activity log entries for a given day in the format requested using units in the unit system which corresponds to the Accept-Language header provided.

Privacy Setting

The **Activities** (Friends or Anyone) privacy permission grants access to a user's resource with the exception that the response **DOES NOT** include a detailed list of activity log entries.

Considerations

1. Daily summary data and daily goals for elevation (**elevation**, **floors**) only included for users with a device with an altimeter.
2. The **steps** field in activity log entires included only for activities that have steps (e.g. "Walking", "Running"); **distance** only included when it is relevant.
3. Calorie burn goal (**caloriesOut**) represents either dynamic daily target from the premium trainer plan or manual calorie burn goal. Goals are included to the response only for today and 21 days in the past.

Resource URL

```
GET https://api.fitbit.com/1/user/[user-id]/activities/date/[date].json
```

user-id	The encoded ID of the user. Use "-" (dash) for current logged-in user.
date	The date in the format `yyyy-MM-dd`

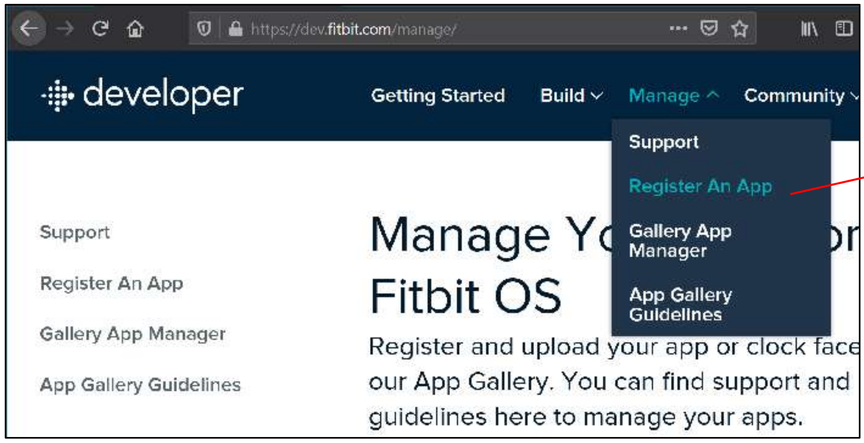
Request Headers

Accept-Locale	optional	The locale to use for response values.
Accept-Language	optional	The measurement unit system to use for response values.

Example Response

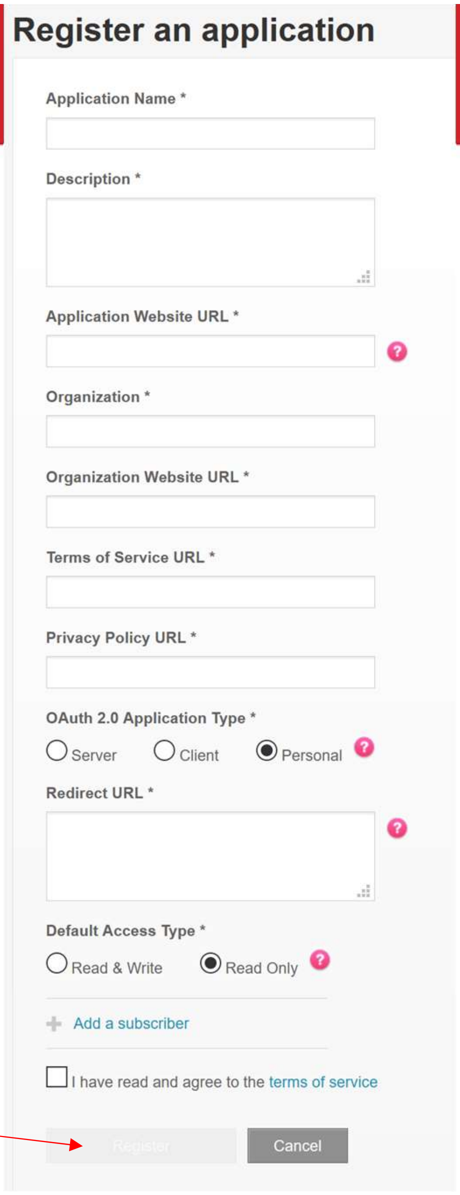
```
{
  "activities": [
    {
      "activityId": 51007,
      "activityParentId": 90019,
      "calories": 230,
      "description": "7mph",
      "distance": 2.04,
      "duration": 1097053,
      "hasStartTime": true,
      "isFavorite": true,
      "logId": 1154701,
      "name": "Treadmill, 0% Incline",
      "startTime": "00:25",
      "steps": 3783
    }
  ],
  "goals": {
    "caloriesOut": 2826,
    "distance": 8.05,
```

Registriamo una nuova app (dev.fitbit.com)



Manage Y… Fitbit OS

Register and upload your app or clock face… our App Gallery. You can find support and guidelines here to manage your apps.



Register an application

Application Name *

Description *

Application Website URL *

Organization *

Organization Website URL *

Terms of Service URL *

Privacy Policy URL *

OAuth 2.0 Application Type *: Server Client [✓ Personal]

Redirect URL *

Default Access Type *: Read & Write [✓ Read Only]

Application name / Description: a vostra scelta

Application Website, Terms of Service, Privacy policy URL: http://localhost

Organization + Website URL: a vostra scelta (io ho inserito DISIT, UPO + sito)

Oauth2 Application Type: Personal (potrà accedere solo l’utente che ha creato l’app)

Redirect URL: http://localhost:6789/Callback

Access Type: Read Only; flaggare «I have read …» poi Click su Register

UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

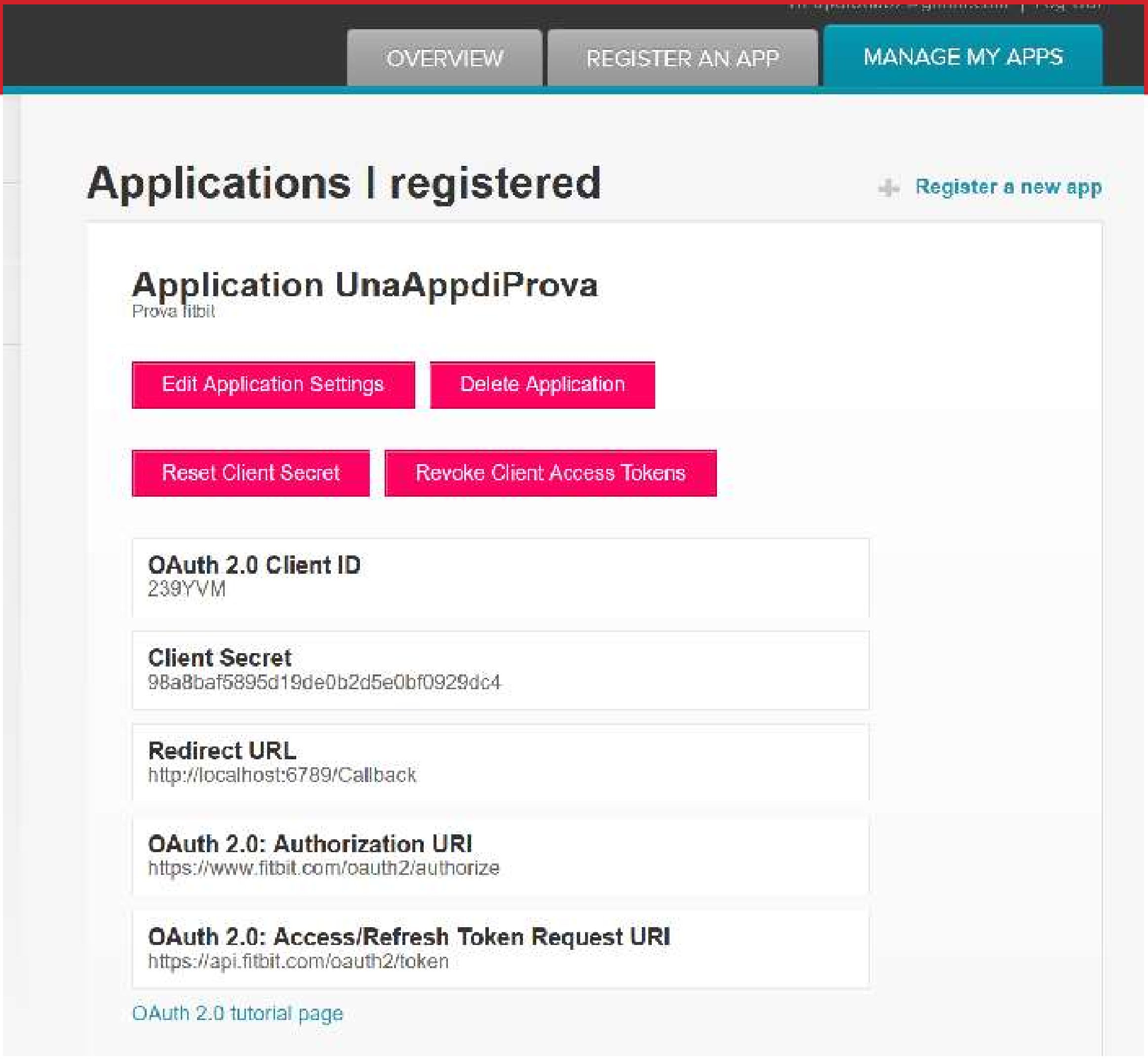
Annotare i campi:

- Client ID
- Client Secret
- Redirect URL (definita da noi)
- Oauth2.0 Authorization URI

https://www.fitbit.com/oauth2/authorize

- Oauth 2.0 Access/Refresh Token Request URI

https://www.fitbit.com/oauth2/token



Applications I registered

Application UnaAppdiProva
Prova fitbit

OAuth 2.0 Client ID
239YVM

Client Secret
98a8baf5895d19de0b2d5e0bf0929dc4

Redirect URL
http://localhost:6789/Callback

OAuth 2.0: Authorization URI
https://www.fitbit.com/oauth2/authorize

OAuth 2.0: Access/Refresh Token Request URI
https://api.fitbit.com/oauth2/token

OAuth 2.0 tutorial page

Sperimentiamo (vedere … fitbit-oauth-java sul dir)

Utilizziamo le librerie di google che implementano il protocollo Oauth2 con «accessori» per avviare un web-server jetty quando si esegue l'applicazione, e per parsificare il json ricevuto dal server.

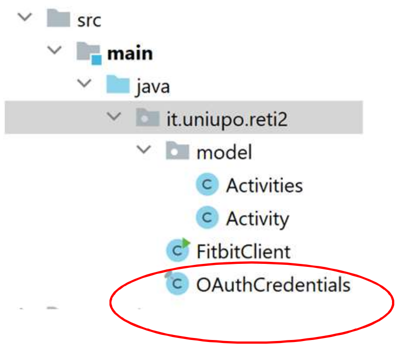
In build.gradle

```
dependencies {
   ….
   // OAuth client library
   implementation 'com.google.oauth-client:google-oauth-client:1.28.0'
   // Jetty extension for the OAuth client library (as local webserver)
   implementation 'com.google.oauth-client:google-oauth-client-jetty:1.28.0'
   // gson support for the OAuth client library
   implementation 'com.google.http-client:google-http-client-gson:1.28.0'
}
```

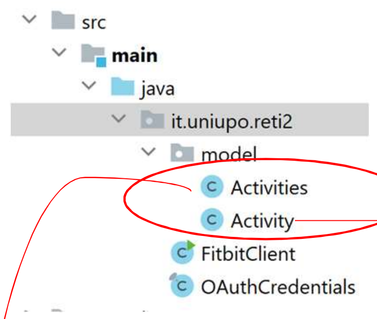
Il programma (un web server che risponderà su *localhost:6789*)

Tutti i dati ricavati al momento della registrazione della «client application» sul sito fitbit si devono inserire in OauthCredentials.java

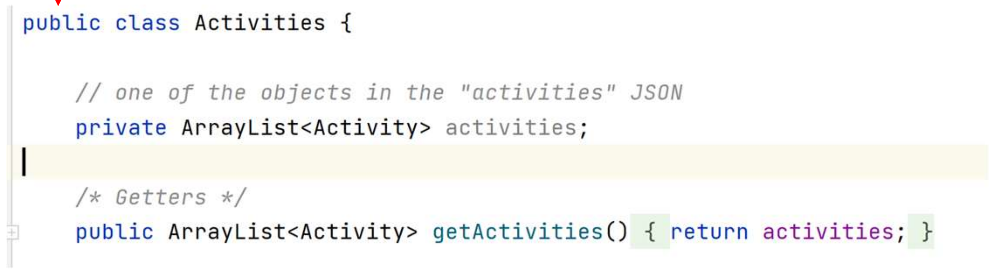
```
 // sample client id, as provided by Fitbit
   private static final String CLIENT_ID = "239YWF";
   // sample client secret, as provided by Fitbit
   private static final String CLIENT_SECRET = "821b9097c207ffa914c985a8a51bb8a3";
   // domain and port of the local server to complete the OAuth 2.0 authorization flow
   private static final int PORT = 6789;
   private static final String DOMAIN = "localhost";

   // server URLs, as provided by Fitbit
   private static final String TOKEN_SERVER_URL = "https://api.fitbit.com/oauth2/token";
   private static final String AUTHORIZATION_SERVER_URL = "https://www.fitbit.com/oauth2/authorize";
```



Il modello dei dati (per estrarre i dati dal json ottenuto dal server fitbit)



https://dev.fitbit.com/build/reference/web-api/activity/



```java
public class Activities {

    // one of the objects in the "activities" JSON
    private ArrayList<Activity> activities;

    /* Getters */
    public ArrayList<Activity> getActivities() { return activities; }
```

```java
/**
 * Represents an activity, according to the Fitbit Web API
 */
public class Activity {
    // private String name
    private String activityParentName;
    // calories
    private int calories;
    // activity description
    private String description;
    // distance (km)
    private float distance;
    // steps
    private int steps;
...
... }
```

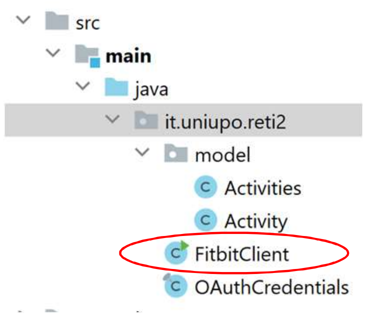
array di attività

```json
"activities": [
  {
    "activityId": 51007,
    "activityParentId": 90019,
    "activityParentName " : " Run",
    "calories": 230,
    "description": "7mph",
    "distance": 2.04,
    "duration": 1097053,
    "hasStartTime": true,
    "isFavorite": true,
    "logId": 1154701,
    "name": "Treadmill, 0% Incline",
    "startTime": "00:25",
    "steps": 3783
  }
```

json restituito dal server quando interrogato dall'applicazione

UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

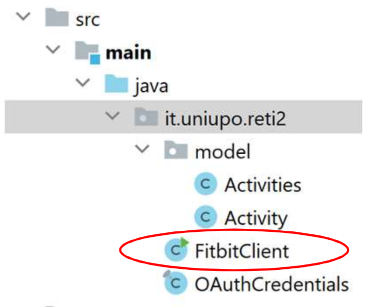
Classe principale (fase di autorizzazione e interrogazione)



```
public static void main(String[] args) {
  try {
    // authorization flow
    final Credential credential = authorize();
    // print JWT access token
    System.out.println(credential.getAccessToken().toString());
    System.out.println("Authorization completed");
    // init a requestFactory
    HttpRequestFactory requestFactory =
        HTTP_TRANSPORT.createRequestFactory((HttpRequest request) -> {
          credential.initialize(request);
          request.setParser(new JsonObjectParser(JSON_FACTORY));
        });
    // execute the API call, backed by OAuth 2.0
    run(requestFactory);
    System.out.println("Authorized");
    // Success!
    return;
  } catch (IOException e) {
    System.err.println(e.getMessage());
  } catch (Throwable t) {
    t.printStackTrace();}
```

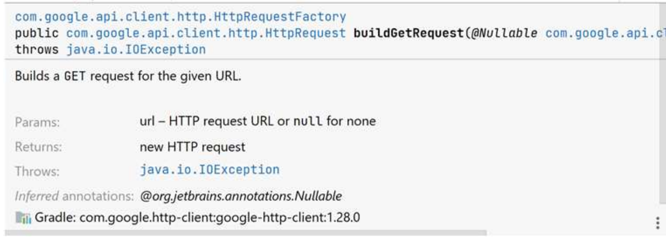
UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

Classe principale (fase di autorizzazione e interrogazione)



```
private static void run(HttpRequestFactory requestFactory) throws IOException {
        // the URL to call
        GenericUrl url = new GenericUrl("https://api.fitbit.com/1/user/-/activities/date/2022-05-03.json");
        // perform a GET request
        HttpRequest request = requestFactory.buildGetRequest(url);

        // get the response as a JSON (and put it in a string)
        String jsonResponse = request.execute().parseAsString();

        // debug
        System.out.println("[DEBUG] " + jsonResponse);

        // parse the JSON string in POJO thanks to gson
        Activities activities = gson.fromJson(jsonResponse, Activities.class);

        // print out the steps and calories of the first activity (e.g., 1000 steps for a walk)
        for (int i = 0; i < activities.getActivities().size(); i++) {
        System.out.println("Name: " + activities.getActivities().get(i).getName());
        System.out.println("Description: " + activities.getActivities().get(i).getDescription());
        System.out.println("Steps: " + activities.getActivities().get(i).getSteps());
        System.out.println("Calories: " + activities.getActivities().get(i).getCalories());}
```

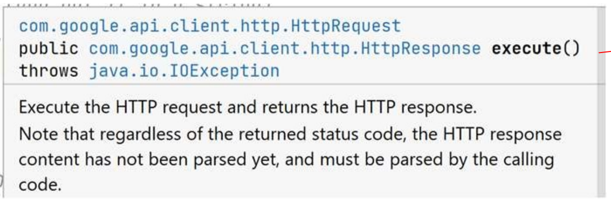
Richiesta relativa all'utente autenticato che deve anche essere il creatore dell'app



```
com.google.api.client.http.HttpRequestFactory
public com.google.api.client.http.HttpRequest buildGetRequest(@Nullable com.google.api.cl
throws java.io.IOException
```

Builds a GET request for the given URL.

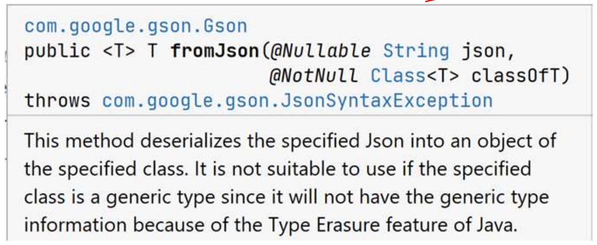
Params: url – HTTP request URL or null for none
Returns: new HTTP request
Throws: java.io.IOException
Inferred annotations: *@org.jetbrains.annotations.Nullable*
Gradle: com.google.http-client:google-http-client:1.28.0



```
com.google.api.client.http.HttpRequest
public com.google.api.client.http.HttpResponse execute()
throws java.io.IOException
```

Execute the HTTP request and returns the HTTP response.
Note that regardless of the returned status code, the HTTP response content has not been parsed yet, and must be parsed by the calling code.

Il json viene parsificato secondo il modello e inserito in un'ArrayList

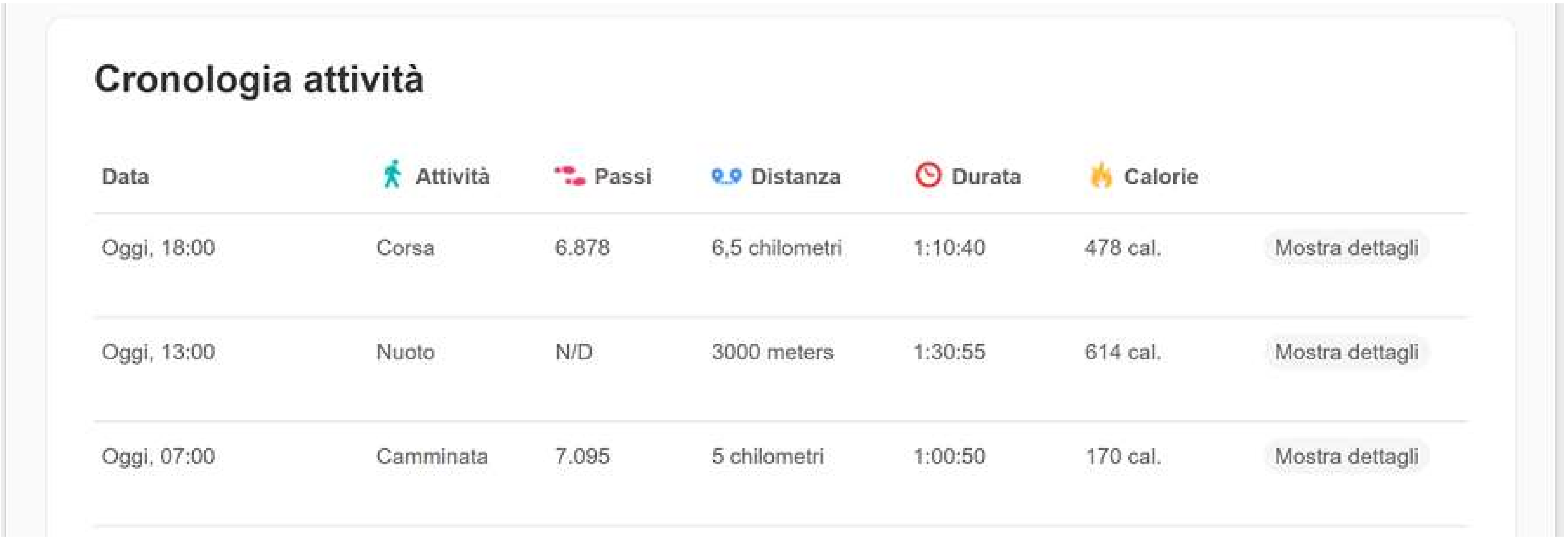


```
com.google.gson.Gson
public <T> T fromJson(@Nullable String json,
                      @NotNull Class<T> classOfT)
throws com.google.gson.JsonSyntaxException
```

This method deserializes the specified Json into an object of the specified class. It is not suitable to use if the specified class is a generic type since it will not have the generic type information because of the Type Erasure feature of Java.

UPO UNIVERSITÀ DEL PIEMONTE ORIENTALE

Eseguiamo l'app sull'utente che ha svolto 3 attività



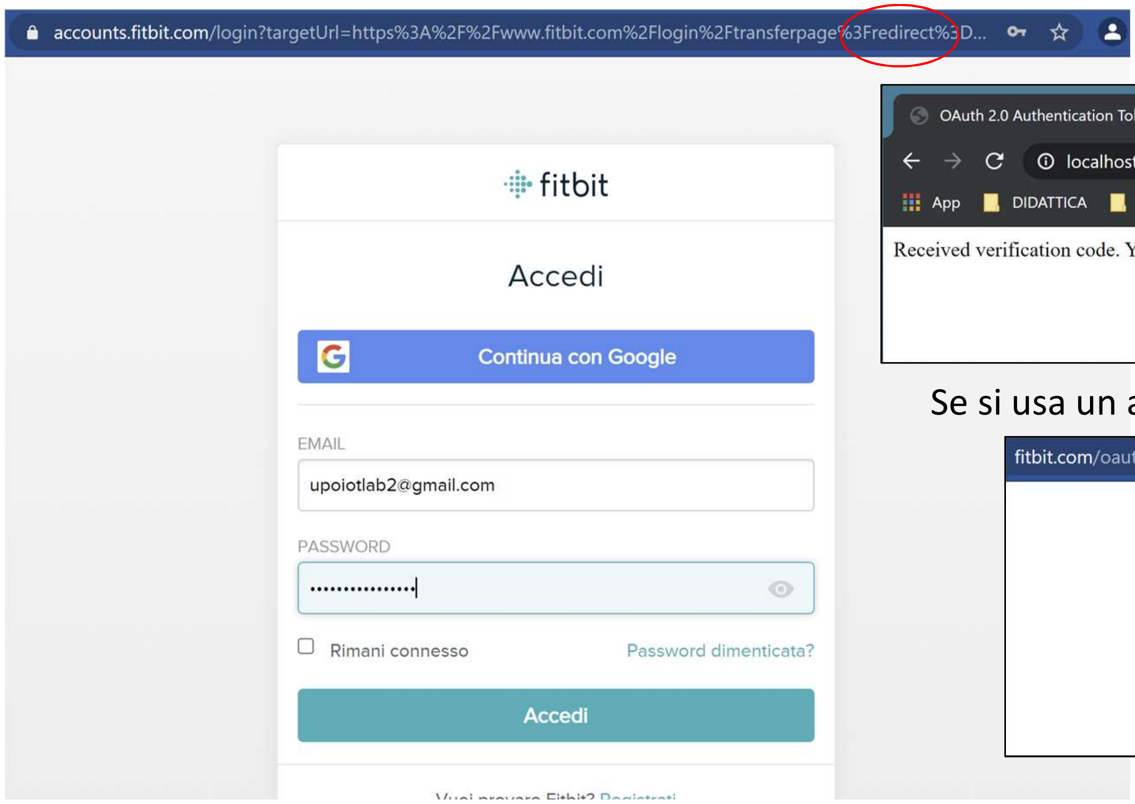
Cronologia attività

Data	Attività	Passi	Distanza	Durata	Calorie	
Oggi, 18:00	Corsa	6.878	6,5 chilometri	1:10:40	478 cal.	Mostra dettagli
Oggi, 13:00	Nuoto	N/D	3000 meters	1:30:55	614 cal.	Mostra dettagli
Oggi, 07:00	Camminata	7.095	5 chilometri	1:00:50	170 cal.	Mostra dettagli

UNIVERSITÀ DEL PIEMONTE ORIENTALE

Avvio della fase di autorizzazione

Viene automaticamente aperto il browser sulla pagina di autenticazione di fitbit, dove ci viene richiesto anche di scegliere QUALI dati mettere a disposizione della client-app.



accounts.fitbit.com/login?targetUrl=https%3A%2F%2Fwww.fitbit.com%2Flogin%2Ftransferpage%3Fredirect%3D...

fitbit

Accedi

Continua con Google

EMAIL

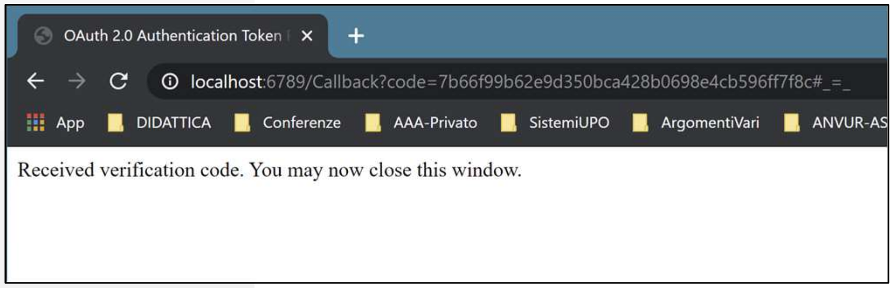
upoiotlab2@gmail.com

PASSWORD

•••••••••••••••

Rimani connesso Password dimenticata?

Accedi

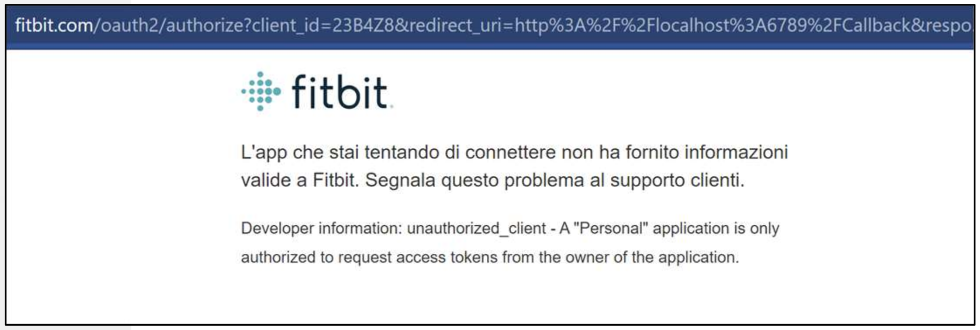


OAuth 2.0 Authentication Token

localhost:6789/Callback?code=7b66f99b62e9d350bca428b0698e4cb596ff7f8c#_=_

Received verification code. You may now close this window.

Se si usa un account diverso da chi ha creato l'app viene bloccato:



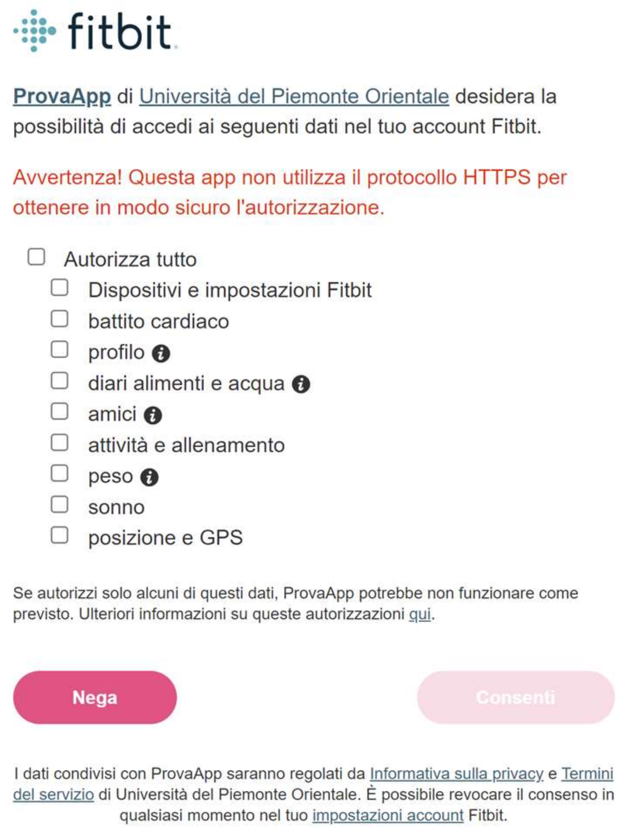
fitbit.com/oauth2/authorize?client_id=23B4Z8&redirect_uri=http%3A%2F%2Flocalhost%3A6789%2FCallback&respo

fitbit

L'app che stai tentando di connettere non ha fornito informazioni valide a Fitbit. Segnala questo problema al supporto clienti.

Developer information: unauthorized_client - A "Personal" application is only authorized to request access tokens from the owner of the application.

La prima volta chiede anche di specificare quali informazioni si autorizza a prelevare



fitbit

ProvaApp di Università del Piemonte Orientale desidera la possibilità di accedi ai seguenti dati nel tuo account Fitbit.

Avvertenza! Questa app non utilizza il protocollo HTTPS per ottenere in modo sicuro l'autorizzazione.

- ☐ Autorizza tutto
  - ☐ Dispositivi e impostazioni Fitbit
  - ☐ battito cardiaco
  - ☐ profilo
  - ☐ diari alimenti e acqua
  - ☐ amici
  - ☐ attività e allenamento
  - ☐ peso
  - ☐ sonno
  - ☐ posizione e GPS

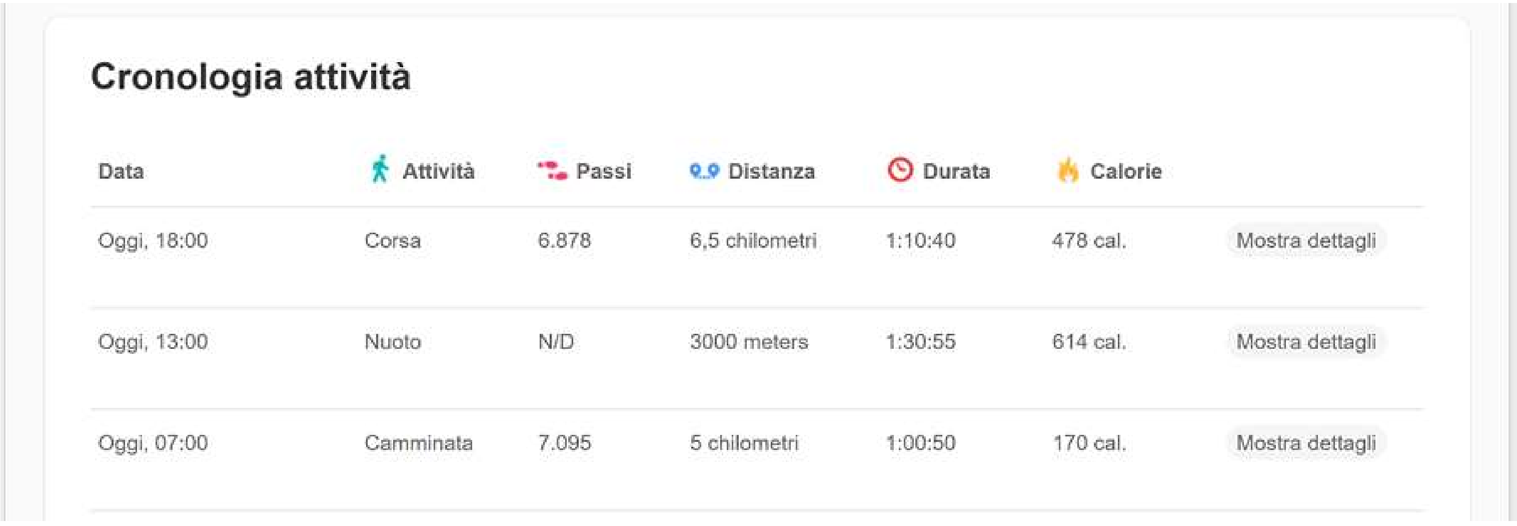
Se autorizzi solo alcuni di questi dati, ProvaApp potrebbe non funzionare come previsto. Ulteriori informazioni su queste autorizzazioni qui.

I dati condivisi con ProvaApp saranno regolati da Informativa sulla privacy e Termini del servizio di Università del Piemonte Orientale. È possibile revocare il consenso in qualsiasi momento nel tuo impostazioni account Fitbit.

UNIVERSITÀ DEL PIEMONTE ORIENTALE

Risultato dell’esecuzione

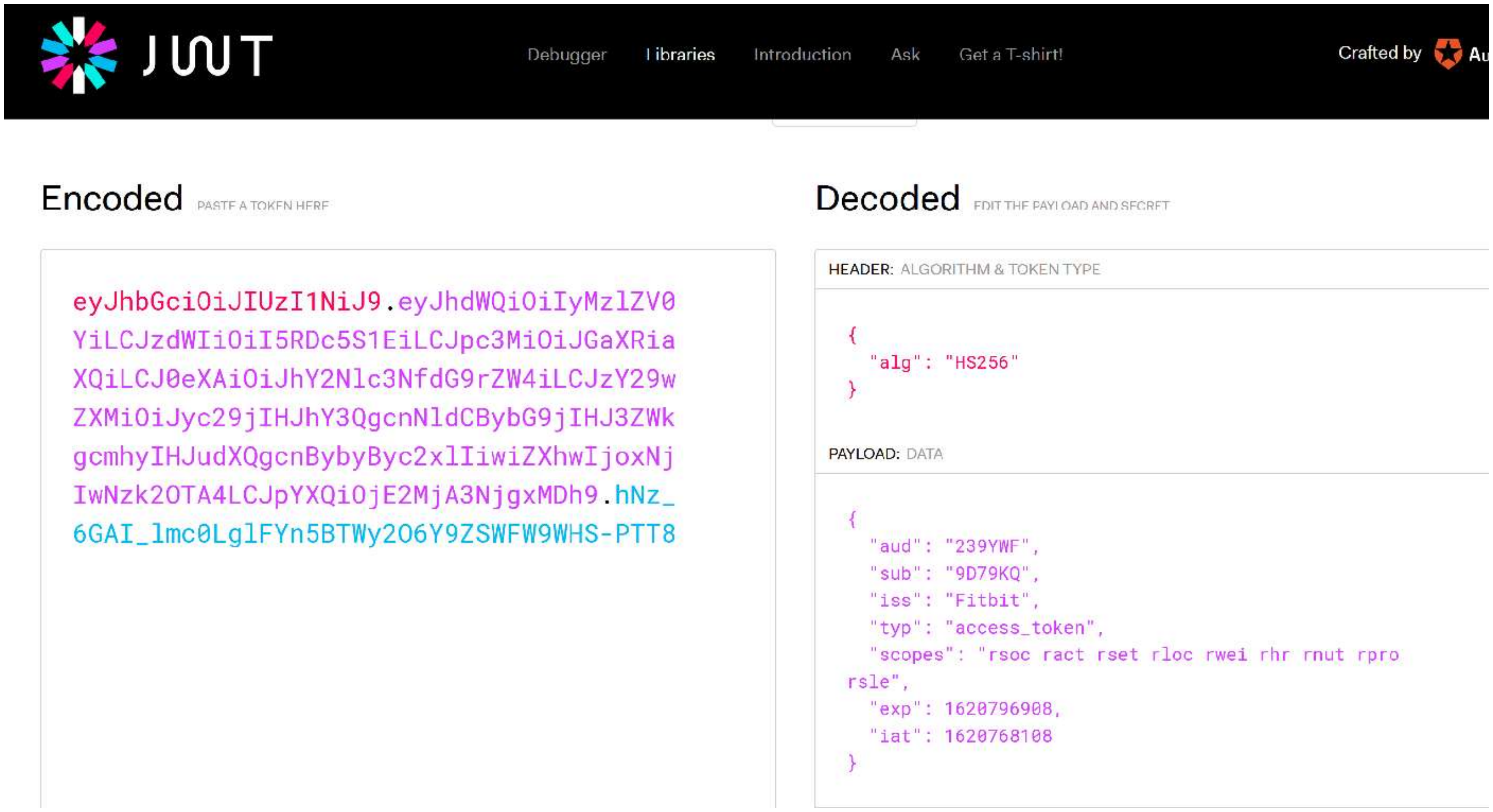
```
Please open the following address in your browser: Authorization endpoint
  https://www.fitbit.com/oauth2/authorize?client_id=239YWF&redirect_uri=http://localhost:6789/Callback&response_type=code&scope=activity%20heartrate%20location%
Attempting to open that address in the default browser now...
2021-05-11 22:52:08.725:INFO::Stopped SocketConnector@localhost:6789                token
eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMzlZV0YiLCJzdWIiOiI5RDc5S1EiLCJpc3MiOiJGaXRiaXQiLCJ0eXAiOiJhY2Nlc3NfdG9rZW4iLCJzY29wZXMiOiJyc29jIHJhY3QgcnNldCBybG9jIHJ3ZWkgcmh
Authorization completed
[DEBUG] {"activities":[{"activityId":17151,"activityParentId":90013,"activityParentName":"Walk","calories":170,"description":"Walking less than 2 mph, strolling
Name: Walk
Description: Walking less than 2 mph, strolling very slowly
Steps: 7095
Calories: 170
Name: Swim
Description: 25-50 yards/min
Steps: 0
Calories: 614
Name: Run
Description: Running - 5 mph (12 min/mile)
Steps: 6878
Calories: 478
Authorized
```



Cronologia attività

Data	Attività	Passi	Distanza	Durata	Calorie	
Oggi, 18:00	Corsa	6.878	6,5 chilometri	1:10:40	478 cal.	Mostra dettagli
Oggi, 13:00	Nuoto	N/D	3000 meters	1:30:55	614 cal.	Mostra dettagli
Oggi, 07:00	Camminata	7.095	5 chilometri	1:00:50	170 cal.	Mostra dettagli

Il token



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyMzlZV0
YiLCJzdWIiOiI5RDc5S1EiLCJpc3MiOiJGaXRia
XQiLCJ0eXAiOiJhY2Nlc3NfdG9rZW4iLCJzY29w
ZXMiOiJyc29jIHJhY3QgcnNldCBybG9jIHJ3ZWk
gcmhyIHJudXQgcnBybyByc2xlIiwiZXhwIjoxNj
IwNzk2OTA4LCJpYXQiOjE2MjA3NjgxMDh9.hNz_
6GAI_lmc0LglFYn5BTWy2O6Y9ZSWFW9WHS-PTT8
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256"
}
```

PAYLOAD: DATA

```
{
  "aud": "239YWF",
  "sub": "9D79KQ",
  "iss": "Fitbit",
  "typ": "access_token",
  "scopes": "rsoc ract rset rloc rwei rhr rnut rpro
rsle",
  "exp": 1620796908,
  "iat": 1620768108
}
```