



1506
UNIVERSITÀ
DEGLI STUDI
DI URBINO
CARLO BO

CORSO DI LAUREA IN
INFORMATICA APPLICATA
SCUOLA DI
SCIENZE TECNOLOGIE E FILOSOFIA DELL'INFORMAZIONE

Reti di Calcolatori

Progetto a.a. 2021/2022 – Prima sessione riservata

“Mashup tra dati Strava e Cycling Analytics”

Sviluppatori:

Simone Cossi, mat. 290796

INDICE

Specifica del progetto	3
<i>Fonti</i>	3
<i>Realizzazione del progetto</i>	3
Scelte progettuali	4
<i>Definizione di API</i>	4
<i>Scelta delle richieste</i>	4
Struttura delle risposte	5
<i>Strava</i>	5
Problemi:	5
<i>Cycling Analytics</i>	6
Difficoltà incontrate	7
Inziali	7
Strava	7
Conclusioni	7
Informazioni sull'utilizzo	8
<i>Testing</i>	8
<i>N.B</i>	9
Soluzione 1: Chiedere direttamente a me	9
Soluzione 2: Seguire la guida alla sostituzione del codice	10
Sitografia	12

Specifica del progetto

Scrivere un programma, con interfaccia web, che metta in relazione i dati di un utente Strava con i dati di un utente CyclingAnalytics.

Per fare ciò possono venir create due identità fittizie (una per app) e quindi procedere con il mashup.

Fonti

Strava API: <https://developers.strava.com/docs/reference/>

Cycling Analytics API: <https://www.cyclinganalytics.com/developer/api#api-endpoints>

Realizzazione del progetto

Nell'atto pratico ciò che il programma va ad eseguire sono due richieste http.

Le richieste vengono effettuate come da specifica a due API differenti:

- API Strava
- API Cycling Analytics

Fatto ciò, il programma andrà ad inserire i dati delle due API in maniera ordinata su un file html, la nostra cosiddetta interfaccia web.

Scelte progettuali

Ho scelto innanzitutto di sviluppare il programma utilizzando:

- JavaScript per eseguire le richieste http delle API e per l'elaborazione dei dati ottenuti
- HTML per la struttura dell'interfaccia web e per la visualizzazione dei dati ottenuti dalle API
- CSS per l'abbellimento dell'interfaccia web

Altra scelta progettuale è quella di avere un'interfaccia di sola visualizzazione, questo è stato fatto poiché da specifica il progetto non necessitava di altre funzionalità.

Successivamente è stato scelto di non visualizzare solamente il mashup dei dati delle due API, ma, di visualizzare anche i dati delle API separatamente le une dalle altre e dal mashup.

L'ultima scelta progettuale è stata concordata con il professore ed è quella di eseguire il mashup dei dati in una maniera più manuale da parte mia in fase di sviluppo, ovvero andando a scegliere con cura i dati che debbano venir visualizzati nel mashup. Altra scelta possibile era quella di cercare di automatizzare la generazione del mashup ma che avrebbe causato la presenza di informazioni superflue e ridondanza di dati.

Definizione di API

Le API (acronimo di Application Programming Interface, ovvero Interfaccia di programmazione delle applicazioni) sono set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi. Consentono ai tuoi prodotti o servizi di comunicare con altri prodotti o servizi senza sapere come vengono implementati, semplificando così lo sviluppo delle app e consentendo un netto risparmio di tempo e denaro.

Scelta delle richieste

Le due API presentano diversi tipi di richieste da poter effettuare e così diversi tipi di dati di ritorno.

Ad esempio, da Strava Sarebbe possibile ottenere informazioni riguardanti dei club e degli equipaggiamenti mentre da Cycling Analytics sarebbe possibile ottenere i dati di alcune attività o dei piani di allenamento.

Tuttavia, come da concordato, era stato deciso di andare a leggere le informazioni riguardanti due utenti fittizi, così facendo le possibili richieste a disposizione andavano a diminuire drasticamente, la scelta è stata infine effettuata sulla base dei dati di ritorno, così da poter avere più dati interessanti.

Struttura delle risposte

L'API Strava fornisce diversi metodi di utilizzo in base al linguaggio utilizzato mentre Cycling Analytics fornisce solo la struttura delle richieste http.

Strava

Nonostante Strava fornisca diversi metodi, tra cui uno anche per il linguaggio da me utilizzato (JavaScript) ho scelto di utilizzare la semplice richiesta http, in modo da poter interagire con le due API allo stesso modo.

Get Authenticated Athlete (getLoggedInAthlete)

Returns the currently authenticated athlete. Tokens with profile:read_all scope will receive a detailed athlete representation; all others will receive a summary representation.

GET /athlete

Responses

HTTP code 200 Profile information for the authenticated athlete. An instance of [DetailedAthlete](#).

HTTP code 4xx, 5xx A [Fault](#) describing the reason for the error.

```
HTTPIe  Java  Obj-C  JavaScript  C#  Python

$ http GET "https://www.strava.com/api/v3/athlete" "Authorization: Bearer [[token]]"

Sample Response

{
  "id" : 12345678987654321,
  "username" : "marianne_t",
  "resource_state" : 3,
  "firstname" : "Marianne",
  "lastname" : "Teutenberg",
  "city" : "San Francisco",
  "state" : "CA",
  "country" : "US",
  "sex" : "F",
  "premium" : true,
  "created_at" : "2017-11-14T02:30:05Z",
  "updated_at" : "2018-02-06T19:32:26Z",
  "badge_type_id" : 4,
  "profile_medium" : "https://xxxxx.cloudfront.net/pictures/athletes/123456789/123456789.jpg",
  "profile" : "https://xxxxx.cloudfront.net/pictures/athletes/123456789/123456789/2/123456789.jpg",
  "friend" : null,
  "follower" : null,
  "follower_count" : 5,
  "friend_count" : 5,
  "mutual_friend_count" : 0,
  "athlete_type" : 1,
  "date_preference" : "%m/%d/%Y",
  "measurement_preference" : "feet",
  "clubs" : [ ],
  "ftp" : null,
  "weight" : 0,
  "bikes" : [ {
    "id" : "b12345678987655",
    "primary" : true,
    "name" : "EMC",
    "resource_state" : 2,
    "distance" : 0
  } ],
  "shoes" : [ {
    "id" : "s12345678987655",
    "primary" : true,
    "name" : "adidas",
    "resource_state" : 2,
    "distance" : 4984
  } ]
}
```

Si può notare nella richiesta http che per funzionare bisogna immettere un codice di autenticazione. Il codice può essere recuperato seguendo la guida al seguente indirizzo:

<https://developers.strava.com/docs/getting-started/>

Oppure seguendo i primi step del tutorial da me scritto a pagina 9 -> [Vai al Tutorial](#)

Problemi:

Questo metodo, come altri, presenta un problema non indifferente:

Per ottenere i dati come appena detto necessitiamo di un codice, questo codice viene cambiato automaticamente ogni sei ore, tuttavia, non sono stato in grado di implementare un'automazione per non dover sostituire manualmente il codice

Altro fattore è che questo codice viene dato solamente se nelle impostazioni dell'account Strava si decide di creare un'applicazione e, una volta generata e approvata, sarà possibile accedere a questo codice.

Cycling Analytics

A differenza del caso precedente qui non si poteva scegliere un metodo piuttosto che un altro, perciò, sono andato solamente a scegliere la richiesta http che più mi sembrava adatta.

/user/<id>

/me

GET

Gets user information.

Request

All are optional.

- `weight` — If `true`, also returns the user's weight history.
- `power` — If `true`, also returns the user's FTP history and power zones.
- `heartrate` — If `true`, also returns the user's resting and maximum heart rate history, and heart rate zones.

Response

- `id`
- `name`
- `email`
- `timezone` — A timezone name in the `pytz common_timezones_set`.
- `units` — Currently either `metric` or `us`.
- `sex` — Either `male` or `female`.
- `weight` — An array of `{date: <date>, value: <weight>}` objects.
- `power` — An object containing:
 - `ftp` — An array of `{date: <date>, value: <ftp>}` objects.
 - `zones` — An array of `{value: <minimum power FTP%>, name: <name>}` objects.
The maximum power for a zone is based on the minimum power for the next zone. All values are specified as a percentage of FTP.
- `heartrate` — An object containing:
 - `resting` — An array of `{date: <date>, value: <resting heart rate>}` objects.
 - `maximum` — An array of `{date: <date>, value: <maximum heart rate>}` objects.
 - `zones` — An array of `{value: <minimum heart rate>, name: <name>}` objects.
The maximum heart rate for a zone is the minimum heart rate for the next zone minus one.

Anche in questo caso bisogna autenticarsi con un codice, ma fortunatamente questo non cambia ogni tot tempo.

Per la struttura della richiesta rimando alla pagina ufficiale che spiega al meglio la struttura della richiesta e dei parametri da inserire.

<https://www.cyclinganalytics.com/developer/api>

Anche in questo caso il codice di accesso viene dato nella sezione app delle impostazioni profilo dell'utente.

Difficoltà incontrate

Inziali

Innanzitutto, ho dovuto studiare le basi dei linguaggi che ho utilizzato dato che non li avevo mai utilizzati in precedenza.

Successivamente ho avuto difficoltà a comprendere come poter iniziare ad utilizzare le api, dato che ad esempio su Strava è possibile creare una app solo tramite un link diretto e non è possibile arrivarci tramite le impostazioni profilo come una volta che ne è stata creata una.

Perciò dopo un lungo studio di tutte le pagine di documentazione fornite dalle due API ho proceduto con delle prove fino a che non ho raggiunto una fase di stabilità e di funzionamento del programma nonostante i suoi difetti.

Strava

Qua parlo del problema brevemente descritto in precedenza.

L'API Strava necessita di un codice con scadenza di sei ore, ciò è un grande problema che non sono riuscito a risolvere.

Seguendo la guida data da Strava per l'autenticazione "automatica" ho riscontrato discordanze da quanto detto dalla guida e quanto necessitava il programma per la corretta esecuzione.

Ad esempio:

Web applications

Redirect the user to `GET https://www.strava.com/oauth/authorize` .

La guida per l'autenticazione per una web app dice solamente di reindirizzare la pagina come indicato, eseguito il reindirizzamento come indicato si arriva alla seguente pagina:

```
{"message":"Bad Request","errors":[{"resource":"Application","field":"client_id","code":"invalid"}]}
```

Si comprende facilmente che necessita di un 'id' di un utente registra, procedendo con le prove e passando come parametro il client_id del mio utente fittizio arriviamo alla seguente pagina:

```
{"message":"Bad Request","errors":[{"resource":"Application","field":"redirect_uri","code":"invalid"}]}
```

Si comprende anche qua che basta aggiungere un altro parametro per risolvere il problema, tuttavia nonostante molti test dove provavo il reindirizzamento sia alla pagina html (offline) sia a altre pagine web (online, ho utilizzato per le prove un sito online funzionante a cui ho accesso) non sono mai riuscito a procedere...

Perciò ho abbandonato questa via e facendo un passo indietro ho continuato lo sviluppo dell'app lasciando la sostituzione del codice di autenticazione in maniera manuale.

Conclusioni

Posso concludere con l'ultima problematica: questa comportava la sovrapposizione di elementi nell'interfaccia web al rimpicciolirsi della schermata di visualizzazione, problema che è andato a risolversi da solo con dei miglioramenti del codice CSS.

Informazioni sull'utilizzo

Per eseguire/visualizzare l'app basta scaricare tutto il progetto così come lo si trova su GitHub e aprire con un qualsiasi browser il file chiamato Progetto.html.

Testing

Inserisci qui il token Strava

STRAVA

id	95420162
username	null
resource_state	2
firstname	Simone
lastname	Cossi
bio	IT Student
city	Urbino
state	Pesaro and Urbino
country	Italy
sex	M
premium	false
summit	false
created_at	2021-11-16T17
updated_at	2021-11-16T17
badge_type_id	0
weight	79.0
profile_medium	https
profile	https
friend	null
follower	null

MASHUP

firstname	Simone
lastname	Cossi
bio	IT Student
city	Urbino
state	Pesaro and Urbino
country	Italy
email	ilsimo99@live.it
sex	male
timezone	Europe/Rome
units	metric

CYCLING ANALYTICS

email	ilsimo99@live.it
id	8232966
name	Simone
sex	male
timezone	Europe/Rome
units	metric

Studente: Simone Cossi
Matricola: 290796

Professore: Antonio Della Selva
Corso: Reti di Calcolatori

Come potete vedere questo è il programma aperto su una pagina di un browser

- a sinistra verranno visualizzate le informazioni dell'account Strava e una casella di inserimento per inserire un eventuale token se il programma non stesse funzionando
- a destra verranno visualizzate le informazioni dell'account Cycling Analytics
- al centro verrà visualizzato il mashup

N.B

Il programma potrebbe non funzionare a causa della scadenza del codice dell'API Strava; perciò, lascerò due vie per verificare il corretto funzionamento del programma

Inserisci qui il token Strava

STRAVA

message	Authorization Error
errors	[resource
field	access_token
code	invalid]

MASHUP

code	invalid]
email	ilsimo99@live.it
sex	male
timezone	Europe/Rome
units	metric

CYCLING ANALYTICS

email	ilsimo99@live.it
id	8232966
name	Simone
sex	male
timezone	Europe/Rome
units	metric

Studente: Simone Cossi

Matricola: 290796

Professore: Antonio Della Selva

Corso: Reti di Calcolatori

Esempio del codice scaduto

Soluzione 1: Chiedere direttamente a me

Mandare una mail alla e-mail seguente e provvederò il prima possibile alla sostituzione del codice.

s.cossi2@campus.uniurb.it

Soluzione 2: Seguire la guida alla sostituzione del codice

Aprire il link seguente:

<https://www.strava.com/login>

Eseguire l'accesso tramite le seguenti credenziali:

- e-mail: s.cossi2@campus.uniurb.it
- password: ProgettoReti

Andare su impostazioni

STRAVA Dashboard Training Explore Challenges [Subscribe](#)

My API Application

Category Visualizer

Club

Client ID 73791 [API Documentation](#)

Client Secret ***** [show](#)

Your Access Token (?) ***** [show](#) **2**

scope: read expires at 2021-11-16T19:53:18+00:00 **3**

Your Refresh Token (?) ***** [show](#)

scope: read

Rate Limits (?) 100 requests every 15 minutes, 1000 daily

Daily requests (?)

Date	Requests
Nov 10, 2021	40
Nov 11, 2021	30
Nov 12, 2021	100
Nov 13, 2021	70
Nov 14, 2021	60
Nov 15, 2021	0
Nov 16, 2021	0

Requests every 15 minutes

Time	Requests
7 PM	0
10 PM	0
1 AM	0
4 AM	0
7 AM	0
10 AM	0
1 PM	0
4 PM	1

My Account

Email
ilsimo99@live.it

Membership Status
Free Account [Subscribe](#)

Social Connections

- [Connect with Garmin](#)
Automatic uploads and Strava Live Segments
- [Connected as Simone Cossi](#)
[Disconnect](#)
- [Connect with Google](#)
Authenticate with your Google account
- [Connect with MyFitnessPal](#)
Share your Strava activities on MyFitnessPal

Andare su My API Application (1)

Trovare "Your Access Token" (2)

Premere Show (3)

Your Access Token (?) 6df187e5fe5c37536a4eb23a9312b8cd65dccc19 [hide](#)
scope: read
expires at 2021-11-16T19:53:18+00:00

Copiare il codice che verrà mostrato

Inserirlo come nel box di input sopra alla tabella **Strava**

3e962dc7e4fd0db8b23882a035ea9424360761dc



STRAVA

id	95420162
username	null
resource_state	2
firstname	Simone
lastname	Cossi
bio	IT Student
city	Urbino
state	Pesaro and Urbino
country	Italy
sex	M
premium	false
summit	false
created_at	2021-11-16T17
updated_at	2021-11-16T17
badge_type_id	0
weight	79.0
profile_medium	https
profile	https
friend	null
follower	null

MASHUP

firstname	Simone
lastname	Cossi
bio	IT Student
city	Urbino
state	Pesaro and Urbino
country	Italy
email	ilsimo99@live.it
sex	male
timezone	Europe/Rome
units	metric

CYCLING ANALYTICS

email	ilsimo99@live.it
id	8232966
name	Simone
sex	male
timezone	Europe/Rome
units	metric

Studente: Simone Cossi

Matricola: 290796

Professore: Antonio Della Selva

Corso: Reti di Calcolatori

Sitografia

- Per le informazioni base e lo studio dei linguaggi
<https://www.w3schools.com/>
- Definizione di API
<https://www.redhat.com/it/topics/api/what-are-application-programming-interfaces>
- API Strava
<https://developers.strava.com/docs/reference/>
- API Cycling Analytics
<https://www.cyclinganalytics.com/developer/api#api-endpoints>
- Capire come eseguire richieste http con dei parametri
<https://stackoverflow.com/>
- Per eseguire rapidamente dei test
<https://web.postman.co/>
- Repository progetto
<https://github.com/SimoneCossi/ProgettoRetiDiCalcolatori-SessioneStraordinaria.git>