



Progetto Tecnologie Web

Authorize Me

**Realizzazione di una piattaforma per il rilascio di autorizzazioni
in modo trasparente**

Andrea Pergetti 133288

2021-07-20

Contents

1	Overview	3
2	Traccia	3
3	Tecnologie utilizzate	3
4	Database	3
5	Tipologie di utenti	4
6	Organizzazione logica	5
6.1	Accounts	5
6.2	Authorization	6
6.3	Api	6
7	Funzionalità	6
7.1	User registration, Login/Logout, Reset password	6
7.1.1	User registration	6
7.1.2	Reset password	6
7.2	Profile page, statistics e setting profile	6
7.2.1	Profile page	6
7.2.2	Profile statistics	7
7.2.3	Profile settings	7
7.3	Creazione, eliminazione, aggiornamento autorizzazione	7
7.3.1	Creare una autorizzazione	7
7.3.2	Eliminare un'autorizzazione	7
7.3.3	Aggiornare un'autorizzazione	8
7.4	Admin page	8
7.5	API REST	8
7.5.1	List authorization	8
7.5.2	Create authorization	8
7.5.3	Delete authorization	9
7.5.4	Update authorization	9
7.5.5	Create authorization with JWT	9
8	Test	11
9	Risultati	12
10	Informazioni aggiuntive	14

1 Overview

Lo sviluppo di questo sito Web si pone l'obiettivo di creare un applicativo che permetta a server autorizzativi di rilasciare a dei client dei permessi di accesso ad altri server rendendo questo processo trasparente e verificabile in modo da avere la possibilità di controllare e osservare il comportamento dei server autorizzativi per individuare eventualmente quelli che agiscono maliziosamente o che rilasciano permessi sbagliati. Le informazioni dei permessi dovrebbero successivamente essere loggate in una struttura dati pubblicamente verificabile e non modificabile. Quest'ultima parte però non sarà inclusa nel progetto oggetto della relazione.

2 Traccia

Le funzionalità implementate sono:

- Gli utenti che si possono registrare al sito sono i server autorizzativi, ovvero quelli che rilasciano le autorizzazioni, che in fase di registrazione devono caricare la loro chiave pubblica, necessaria per decodificare i token, ed eventualmente altre informazioni.
- Nel database vengono salvate le informazioni riguardo le autorizzazioni rilasciate, in particolare identificativo client, identificativo server, data di inizio validità autorizzazione e data di fine validità autorizzazione.
- Nella piattaforma admin viene data la possibilità di fare query e ispezionare le varie autorizzazioni. Nella piattaforma degli utenti (server) invece, sarà presente la lista delle autorizzazioni che essi hanno rilasciato, con possibilità di ordinamento e ricerca, ed inoltre qualche informazione statistica come l'andamento temporale del numero di richieste inviate.
- Utilizzo di API REST per permettere ai server di rilasciare una autorizzazione, cioè inviare un token, e ai client di verificare se un'autorizzazione è valida in un certo istante di tempo.

3 Tecnologie utilizzate

Come framework principale per il progetto si è utilizzato Django.

Per quanto riguarda il database si è utilizzato SQLite, principalmente per questioni di comodità e praticità.

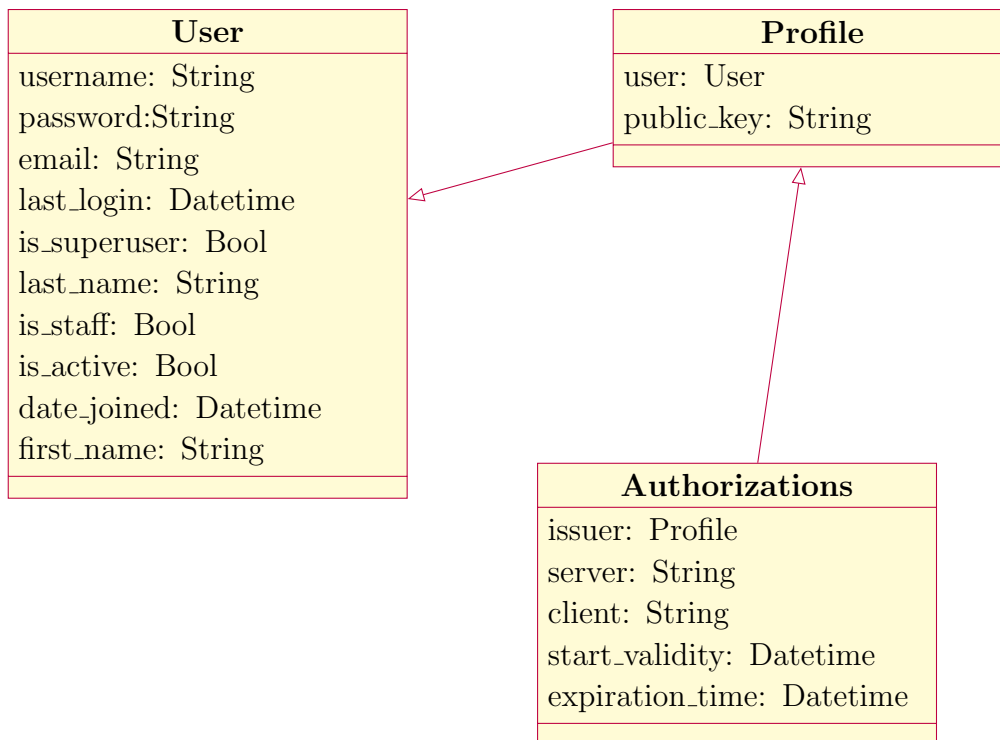
4 Database

Sul database vengono salvate le informazioni sugli utenti e sulle autorizzazioni.

Per quanto riguarda gli utenti sono stati considerati i campi username, password, email e chiave pubblica, caricata come testo (opzionale). Dato che la maggior parte

dei campi è già presente nel modello User di default di Django si è pensato di estenderlo aggiungendo il campo per l'inserimento della chiave pubblica. Quindi il modello degli utenti (Profile model) è composto da un campo user, che è una relazione one-to-one con il modello User default, dato che ogni profilo può far riferimento solamente ad un unico utente e viceversa, e il campo public key, che deve essere in formato PEM e viene inserito come testo. Per quanto riguarda le autorizzazioni, invece, vengono salvati:

- il campo issuer, cioè chi rilascia l'autorizzazione che è una relazione Foreign key con la tabella profile (quella degli utenti) in quanto un utente può rilasciare più autorizzazioni;
- il campo server, che indica il nome del server a cui l'autorizzazione dà accesso;
- il campo client, che è il client che riceve il permesso di accesso;
- il campo start validity, che si riferisce alla data temporale in cui comincia la validità dell'autorizzazione;
- il campo expiration time, che indica la data temporale di fine validità.



5 Tipologie di utenti

Per quanto riguarda gli utenti si è sviluppato l'applicativo web pensando a tre entità diverse con diversi permessi:

- Admin: questi sono gli utenti con i massimi permessi e gli unici che possono accedere all'area Admin. Hanno il controllo sugli utenti e le autorizzazioni oltre alle statistiche relative ad esse;
- Utenti normali: questi sono gli utenti che possono registrarsi al sito. Hanno il permesso di creare, eliminare e aggiornare le proprie autorizzazioni, e l'accesso alle statistiche relative;
- Utenti anonimi: essi non hanno una parte attiva nell'applicativo ma attraverso le API REST hanno il permesso di verificare le autorizzazioni rilasciate per poter verificare, ad esempio, se in un determinato tempo una data autorizzazione è valida

6 Organizzazione logica

Si è deciso di sviluppare 3 applicazioni, di cui 2 sono legate allo sviluppo delle principali entità dell'applicativo, cioè gli utenti e le autorizzazioni, e la terza è legata allo sviluppo delle API REST.

Quindi le applicazioni implementate sono:

- accounts
- authorizations
- api

6.1 Accounts

Questa applicazione è dedicata alle funzionalità che riguardano gli utenti. È stata implementata la funzione di registrazione utenti, di login e logout e di reset della password. Inoltre sono state implementate:

- la pagina del profilo utente, all'interno della quale troviamo le autorizzazioni rilasciate,
- la pagina dei setting dell'utente, in cui si possono cambiare alcuni campi del profilo (password, email e chiave pubblica)
- la pagina riguardante alcune statistiche dell'utente (grafico temporale delle autorizzazioni rilasciate e autorizzazioni attualmente attive).

Gli utenti che si possono registrare sono i server autorizzativi che possono rilasciare le autorizzazioni. In questo documento per semplicità verranno sempre riferiti come utenti e non come server autorizzativi.

6.2 Authorization

Questa applicazione è dedicata alle funzionalità che riguardano le autorizzazioni. È stata implementata la funzione di creazione, eliminazione e aggiornamento di un'autorizzazione. Inoltre accedendo con il proprio account è possibile visualizzare la lista delle autorizzazioni che quell'utente ha rilasciato.

6.3 Api

In questa applicazione sono presenti le API REST. Viene data la possibilità di recupero delle autorizzazioni, andando eventualmente a specificare parametri di selezione; è inoltre permessa la creazione, modifica e cancellazione di un'autorizzazione. Viene anche permesso, infine, di rilasciare un'autorizzazione tramite token (utilizzando un JWT).

7 Funzionalità

7.1 User registration, Login/Logout, Reset password

7.1.1 User registration

Nella fase di registrazione l'utente deve fornire uno username, una password, una email (opzionale) e la chiave pubblica come testo (opzionale). Quest'ultima serve per poter utilizzare la funzionalità di rilascio di un'autorizzazione utilizzando le API REST tramite un Json Web Token, di cui parleremo successivamente.

7.1.2 Reset password

Per il reset della password viene mostrato un form di richiesta dell'indirizzo email. Se l'indirizzo email non è trovato tra gli indirizzi registrati sul database non viene dato errore ma non viene inviata la mail. Se invece l'indirizzo email viene trovato si procede all'invio della mail con il link per fare il reset della password. Per convenienza si è scelto di non inviare mail reali ma di farle scrivere su standard output. Il link presente nella mail reindirizza su un form in cui si può inserire la nuova password. Impostata la nuova password si viene reindirizzati alla pagina che conferma il successo del procedimento.

7.2 Profile page, statistics e setting profile

7.2.1 Profile page

Nella pagina profilo utente si può accedere solo dopo aver fatto il login. All'interno di essa vengono visualizzate in una tabella tutte le autorizzazioni da lui rilasciate. È possibile ordinare le righe per i vari campi presenti nella tabella (server, client, inizio validità, fine validità) cliccando sul nome del campo per il quale la si vuole

ordinare (nell'header della tabella). É possibile ricercare un valore specifico nella tabella attraverso l'apposita casella *Search* ed inoltre si può modificare il numero di entries da visualizzare contemporaneamente cambiando il valore del campo *Show entries*.

Le funzioni di ricerca e ordinamento sono state ottenute usando un plugin di jQuery chiamato DataTables.

7.2.2 Profile statistics

Dalla pagina profilo si può passare alla pagina statistics in cui vengono evidenziate alcune informazioni sull'attività dell'utente. In particolare viene mostrato un grafico con l'andamento temporale delle nuove richieste che sono state fatte dall'utente corrente. Sull'asse delle x ci sono le date e su quello delle y ci sono il numero delle nuove autorizzazioni rilasciate. Inoltre viene indicato il numero delle autorizzazioni, rilasciate dall'utente, attualmente attive.

7.2.3 Profile settings

Dalla pagina di profilo si può passare alla pagina dei settings dell'utente, in cui vengono presentate alcune informazioni sull'account (email e chiave pubblica) e da cui si possono modificare alcuni campi del profilo utente come la password, l'email e la chiave pubblica. Se si clicca su una di queste opzioni si viene reindirizzati ad un form in cui è possibile inserire il nuovo valore per il campo selezionato che, se supera i rispettivi controlli (ad esempio deve essere un valore diverso da quello precedente) viene sostituito nel database.

7.3 Creazione, eliminazione, aggiornamento autorizzazione

7.3.1 Creare una autorizzazione

Si può accedere alla pagina di creazione di un'autorizzazione dalla pagina profilo utente oppure direttamente con l'url, dopo aver fatto l'accesso. Nella pagina di creazione è presente un form con le varie informazioni da inserire ovvero server, client, inizio e fine validità. Quando si fa la richiesta POST vengono fatti dei controlli sulla presenza (i campi sono tutti richiesti) e sulla correttezza dei campi (le due date non possono essere prima della data corrente e la fine validità non può precedere l'inizio validità). Se va a buon fine l'autorizzazione viene creata e si viene ridirezionati sulla pagina profilo.

7.3.2 Eliminare un'autorizzazione

Si può accedere alla pagina di eliminazione di un'autorizzazione dalla pagina profilo utente oppure direttamente con l'url, ma bisogna aver fatto l'accesso. Se si

accede alla funzione tramite la pagina profilo si viene prima indirizzati su una pagina con una tabella simile alla pagina profilo in cui si può scegliere, selezionando il radio button corrispondente, un'autorizzazione che si vuole eliminare. Scelta l'autorizzazione si viene direzionati nella pagina in cui si può confermare definitivamente l'eliminazione.

7.3.3 Aggiornare un'autorizzazione

Si può accedere alla pagina di aggiornamento di un'autorizzazione dalla pagina profilo utente oppure direttamente con l'url, dopo l'accesso. Accedendo dalla pagina profilo si fanno gli stessi passaggi che si sono spiegati per l'eliminazione, con la differenza che selezionata l'autorizzazione che si vuole aggiornare si viene direzionati ad un form già compilato con le informazioni del permesso selezionato, che possono essere cambiate.

7.4 Admin page

Nella pagina di admin sono state aggiunte la tabella riguardante i profili e la tabella delle autorizzazioni. In entrambe le tabelle è possibile ordinare le entries per i vari campi e fare ricerca di uno specifico valore inserendolo nell'apposito Search box. Nella pagina riguardante le autorizzazioni si è aggiunto anche il grafico temporale che indica le autorizzazioni rilasciate da tutti gli utenti e il numero totale delle autorizzazioni attualmente attive.

7.5 API REST

7.5.1 List authorization

È stata creata una API per recuperare la lista di autorizzazioni. Se usato l'url senza parametri viene restituita l'intera lista di autorizzazioni. In alternativa si possono specificare i parametri *server* e *client* (entrambi o uno solo, es.

`api/v1/authorizations?server=s&client=c` o `api/v1/authorizations?server=s`) che restituiscono le autorizzazioni filtrate per i campi *server* e/o *client* specificati.

Questa API permette richieste con i metodi GET, HEAD, OPTIONS e non ha restrizione dei permessi in quanto proprio per il concetto di trasparenza che è alla base dell'applicativo viene permesso a qualunque client di verificare le autorizzazioni rilasciate e le autorizzazioni attive per un determinato client e server.

7.5.2 Create authorization

Viene data la possibilità di creare una autorizzazione indicando i vari campi escluso il campo *issuer* che viene compilato automaticamente con il nome dell'utente che si autentica.

Questa API permette richieste con i metodi POST e OPTIONS ed è usufruibile solo per gli utenti autenticati con le proprie credenziali.

7.5.3 Delete authorization

Viene data la possibilità di eliminare un'autorizzazione. L'autorizzazione che si vuole eliminare e di cui si indica l'ID nell'url deve, ovviamente, essere stata rilasciata dall'utente che fa la richiesta, se no essa fallisce.

Questa API permette richieste con i metodi GET, DELETE, HEAD e OPTIONS ed è usufruibile solo per gli utenti autenticati con le proprie credenziali.

7.5.4 Update authorization

Viene data la possibilità di aggiornare un'autorizzazione. Come per il punto precedente per poter aggiornare un'autorizzazione essa deve essere stata rilasciata dall'utente che fa la richiesta, altrimenti essa fallisce.

Questa API permette richieste con i metodi GET, PUT, PATCH, HEAD e OPTIONS ed è usufruibile solo per gli utenti autenticati con le proprie credenziali.

7.5.5 Create authorization with JWT

Viene data la possibilità di rilasciare un'autorizzazione tramite JWT quindi utilizzando un token che identifica un'autorizzazione e il server autorizzativo. Per fare ciò si è utilizzato la libreria PyJWT.

Il server che vuole eseguire questa operazione deve fare una richiesta POST all'apposito url(*/api/v1/authorizations/jwt/create*) inserendo nel payload un JSON con il campo *token* e il campo *user*:

- nel campo *user* deve inserire l'username utilizzato per registrarsi al sito, che serve per recuperare la chiave pubblica corrispondente;
- nel campo *token* deve inserire il token generato con il metodo *encode* della libreria *PyJWT* in cui inserire:
 - la chiave privata
 - l'algoritmo di cifratura utilizzato, che deve essere RS256(RSA Signature con SHA-256) in quanto per semplicità è stato implementato solo con questo algoritmo
 - il campo *payload* che contiene le informazioni che identificano l'autorizzazione (*client*, *server*, *exp* (deve essere di tipo *NumericDate*) che corrisponde a *expiration time* e *nbf* (deve essere di tipo *NumericDate*) che corrisponde a *start validity*)

Questa API non ha nessuna restrizione dei permessi perchè l'autenticazione viene fatta tramite la corrispondenza tra la chiave pubblica registrata nel database e la chiave privata usata per crittografare il token; nel caso questa corrispondenza venisse a mancare la decodifica del token non andrebbe a buon fine e quindi la richiesta fallirebbe.

In questo modo si riesce con un'unica richiesta a fare:

- l'autenticazione, in quanto come spiegato prima per avere successo ci deve essere corrispondenza fra le due chiavi (delle quali la chiave segreta è in possesso solo dell'utente che le ha generate)
- la creazione dell'autorizzazione, in quanto se la fase precedente va a buon fine si hanno già a disposizione tutte le informazioni per creare il permesso.

8 Test

Per eseguire i test è stata utilizzata la classe `TestCase` fornita da Django.

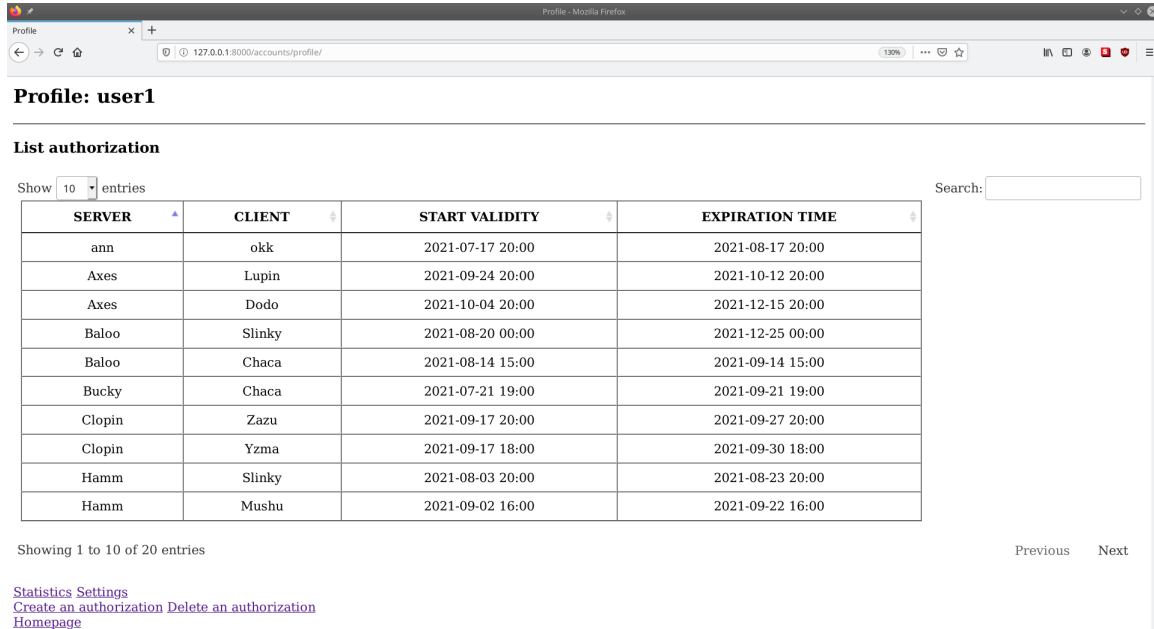
Il primo test è stato fatto sulla view del profilo utente. Il secondo è invece stato fatto sul form di creazione di una autorizzazione.

I test creati sono i seguenti:

- `ProfileViewTest`
 - ☐ `test_profile_page_view_with_user_logged`: Controlla se lo username e il queryset(lista di autorizzazioni) corrispondono alle informazioni dell'utente loggato
 - ☐ `test_profile_page_view_with_no_authorizations`: Controlla quando non è stata rilasciata nessuna autorizzazione(Il messaggio "No authorization released" dovrebbe essere mostrato)
 - ☐ `test_profile_page_view_with_three_authorizations`: Rilascia tre autorizzazioni e verifica che siano tutte mostrate nella view
 - ☐ `test_profile_page_view_link`: Controlla se tutti i link nella view siano funzionanti e ritornino lo status code 200
 - ☐ `test_profile_page_view_with_no_user_logged`: Controlla se provando ad entrare senza essersi autenticati non fa accedere e ti redireziona alla pagina del login
- `AuthorizationCreateTest`
 - ☐ `test_have_expiration_date_before_start_validity`: Controlla che inserendo una data di fine validità precedente alla data di inizio validità dia errore
 - ☐ `test_have_start_validity_before_now`: Controlla che inserendo una data di inizio validità precedente alla data attuale dia errore
 - ☐ `test_have_expiration_time_before_now`: Controlla che inserendo una data di fine validità precedente alla data attuale dia errore
 - ☐ `test_without_field`: Controlla che togliendo uno qualunque dei campi da errore in quanto tutti i campi sono richiesti

9 Risultati

Screenshot applicazione più interessanti



Profile: user1

List authorization

Show entries

Search:

SERVER	CLIENT	START VALIDITY	EXPIRATION TIME
ann	okk	2021-07-17 20:00	2021-08-17 20:00
Axes	Lupin	2021-09-24 20:00	2021-10-12 20:00
Axes	Dodo	2021-10-04 20:00	2021-12-15 20:00
Baloo	Slinky	2021-08-20 00:00	2021-12-25 00:00
Baloo	Chaca	2021-08-14 15:00	2021-09-14 15:00
Bucky	Chaca	2021-07-21 19:00	2021-09-21 19:00
Clopin	Zazu	2021-09-17 20:00	2021-09-27 20:00
Clopin	Yzma	2021-09-17 18:00	2021-09-30 18:00
Hamm	Slinky	2021-08-03 20:00	2021-08-23 20:00
Hamm	Mushu	2021-09-02 16:00	2021-09-22 16:00

Showing 1 to 10 of 20 entries

[Statistics](#) [Settings](#)
[Create an authorization](#) [Delete an authorization](#)
[Homepage](#)

Previous Next

Figure 1: This is an image of the user profile page.

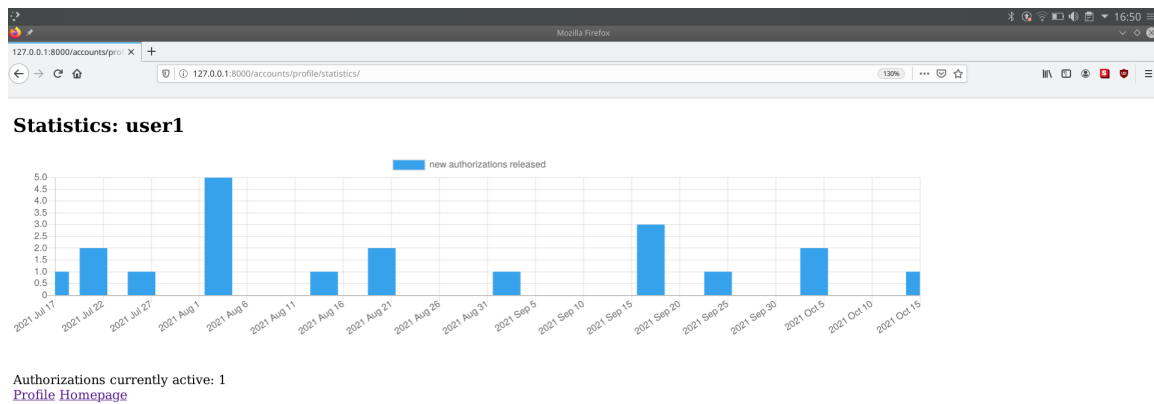


Figure 2: This is an image of the user profile statistics.

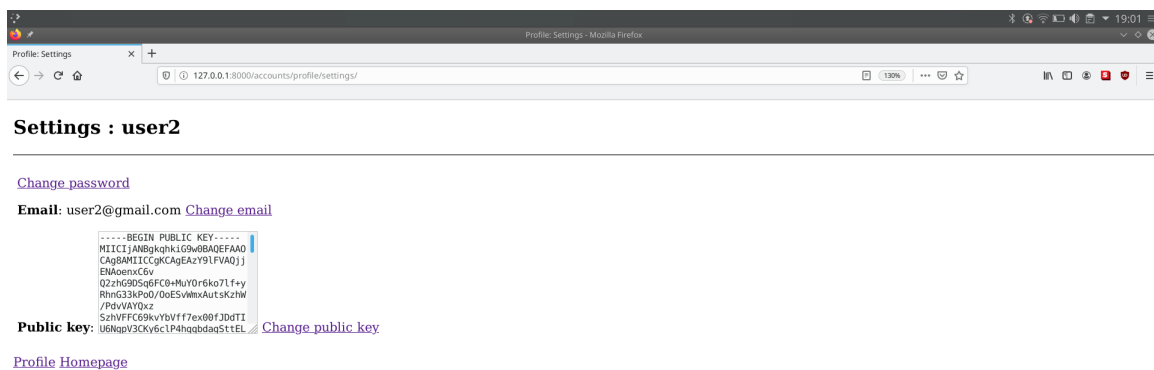


Figure 3: This is an image of the user profile settings.

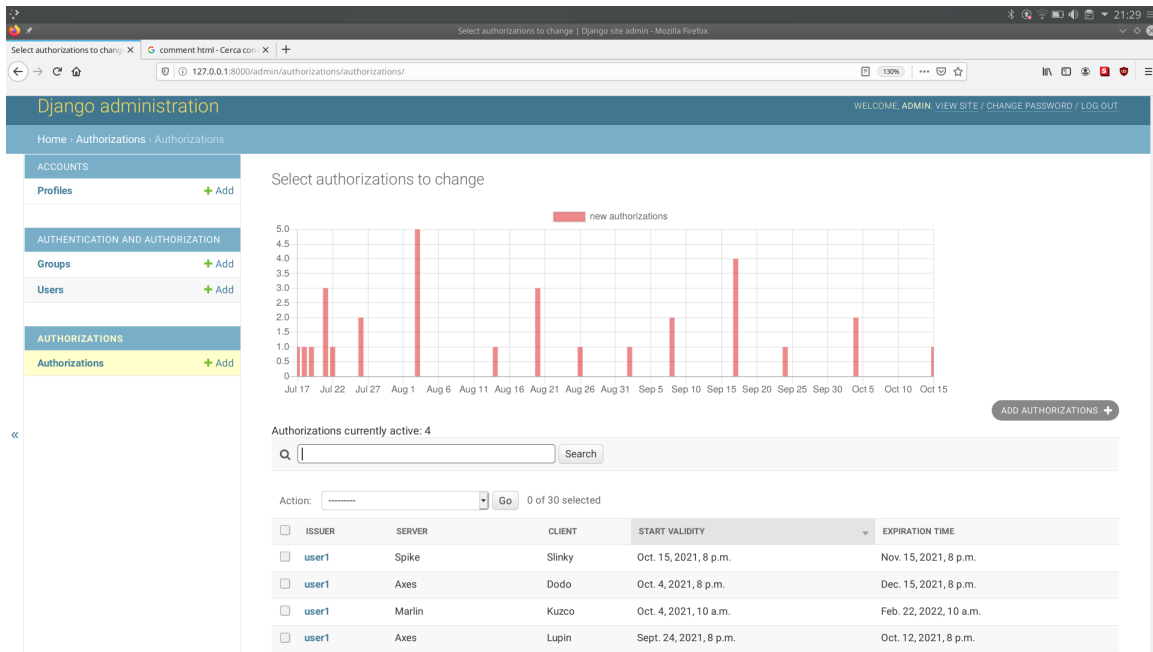


Figure 4: This is an image of the admin page for the authorizations model.

10 Informazioni aggiuntive

Nel database fornito sono presenti alcuni utenti di cui qui vengono fornite le credenziali:

Utenti admin:

username: admin password: admin

Utenti normali:

username: user1 password: utente11

username: user2 password: utente222