

Diario di lavoro

Luogo	Trevano
Data	05.02.2020

Lavori svolti

Oggi inizialmente ho aggiunto nel controller di registrazione la parte che esegue l'hash della password così come in quello di login la parte che la compara. Dopodiché ho iniziato ad implementare il login con l'autenticazione tramite JWT (Json Web Token), per utilizzare i JWT ho dovuto aggiungere a composer *firebase-jwt*. Inizialmente ho generato il metodo `authenticate` nel controller della pagina di login che verifica che i campi del form di login siano corretti e che l'utente esista. Dopodiché se esiste gli ritorno il codice di successo (200) e il token che andrà a salvare in un cookie.

...

```
return response()->json(['token' => $this->jwt($user)], 200);
```

...

```
protected function jwt(User $user) {
    //Imposto i dati del token
    $payload = [
        'iss' => config('ISS'), //Emittente del token
        'sub' => $user->id, //Soggetto del token
        'iat' => time(), //Data e orario dell'emissione
        'exp' => time() + 60*60 //Data e orario dello scadere del token
    ];

    //Genero il token crittografato con la chiave generata contenente i dati
    return JWT::encode($payload, config('KEY'));
}
```

Dopodiché ho aggiunto un Middleware che verifica che, quando si fa una richiesta che necessita l'autorizzazione, il token sia impostato e valido. Il token verrà passato nell'header della richiesta. La richiesta conterrà quindi nell'header:

```
'Authorization: Bearer xxxxxx.yyyyyy.zzzzzz'
```

Problemi riscontrati e soluzioni adottate

Oggi ho avuto parecchi problemi creando il middleware, inizialmente quando facevo dei test mi ritornava sempre un *error 500* che ho poi capito che era dovuto ad una sbagliata lettura della chiave di codifica. Questo generava quindi problemi nella decodifica dove la chiave era corretta.

Inoltre, una volta risolto il primo errore, tutte le volte al *decode* del token veniva ritornato il messaggio di errore *Impossibile decodificare il token*. Dopo diverso tempo perso cercando le più disparate soluzioni ho scoperto che io cercavo di decodificare il token senza rimuovere la stringa *Bearer* (mostrata sopra) e quindi il token era sempre sbagliato. L'errore è quindi stato risolto così:

```
$token = $request->header('Authorization');
```

```
//Rimuovo la stringa 'Bearer' iniziale
$token = explode(" ", $token)[1];
```

Punto della situazione rispetto alla pianificazione

Sono in orario.

Programma di massima per la prossima giornata di lavoro

La prossima volta voglio testare la validazione con JWT e, se ho tempo, creare i validatori frontend.