Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Trevano |
| Data | 06.02.2020 |

|  |
| --- |
| Lavori svolti |
| Oggi ho iniziato testando il token JWT, per farlo ho utilizzato una semplice funzione che andavo a richiamare e dovevo poterlo fare solo se il token fosse valido, dopodiché sono andato a modificare il codice facendo si che essa fosse eseguibile solo se nel token fosse scritto che l’utente è un admin. Per verificare il ruolo nella funzione ho dovuto aggiungerlo alla richiesta così da passarlo dal middleware al metodo:  ...  //Decodifico il token  $credentials = JWT::decode($token, env('APP\_KEY'), ['HS256']);  //Aggiungo alla richiesta il tipo di ruolo così da riutilizzarlo in seguito  $request->request->add(['id\_role' => $credentials->id\_role]);  ...  return $next($request);  E nel metodo lo utilizzo così:  if($request->all()['id\_role'] == 2){  //Se è ammministratore gli mostro la pagina  return view('admin/index');  }else{  //Se non lo è ritorno l'errore  return response()->json(['Unauthorized' => 'Non hai i permessi'], 401);  }  Dopodiché ho fatto la validazione frontend delle due pagine (login e registrazione) portandole così a termine. Essa è fatta tramite HTML5 con i metodi a disposizione ma personalizzando i messaggi di errore tramite JQuery.  $("#email")[0].oninvalid = function () {  this.setCustomValidity("Inserire un'email valida");  }; |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Oggi non ho riscontrato grossi problemi se non con alcune Regex che non ricordavo bene ma cercando su internet ho risolto tutto piuttosto in fretta. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Sono in orario. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| La prossima volta voglio iniziare la pagina di amministrazione. |