SAML autentikacija za Laravel 8.x

SAML autentikacija za Laravel 8.x

Sve radnje radimo kao običan korisnik. U ovom opisu SAML autentikacija je preko komponente aacotroneo/laravel-saml2(link is external) i povezana je sa standardnom laravel autentikacijom, radi se instalacija "od nule".

Kreiranje novog laravel projekta

- `composer create-project laravel/laravel larvel_projekt`
- `cd larvel_projekt`

Promjena ovlasti na ključnim direktorijima da bi apache mogao pisati u njih (ovo je potrebno ako se datoteke nalaze na poslužitelju i uređujemo preko ssh ili sftp, naravno promijenite usera)

- `sudo chown -R nekiuser:www-data storage`
- `sudo chown -R nekiuser:www-data bootstrap/cache`
- `sudo chmod 775 -R storage`
- `sudo chmod 775 -R bootstrap/cache`

Instaliranje laravel debug bar-a (ovo nije nužno, ali pomaže prilikom razvoja)

- `composer require barryvdh/laravel-debugbar --dev`

Instalacija bootstrap-a i laravel autentikacije

- `composer require laravel/ui`
- `php artisan ui bootstrap`
- `npm install`
- `npm run dev`

Umjesto ovog gore, može se odmah dodati "*--auth*", ili ponoviti sa tim parametrom, učinak je isti

- `php artisan ui bootstrap --auth`
- `npm install`
- `npm run dev`

- Podesiti bazu u .env datoteci

```
DB_CONNECTION=mysql
DB_HOST=posluzitelj.domena
DB_PORT=3306
DB_DATABASE=baza
DB_USERNAME=korisnik_baze
DB_PASSWORD=lozinka_za_bazu
```

Instalacija SAML komponente

- `composer require aacotroneo/laravel-saml2`
- `php artisan vendor:publish`

Odabrati **Provider: Aacotroneo\Saml2\Saml2ServiceProvider**

Ovo kreira datoteke: config/saml2_settings.php i config/saml2/test_idp_settings.php

- u datoteci config/saml2_settings.php treba promijeniti slijedeće (*'test' nam ne treba, ali 'AAI' je nužan*)
  red 10:

```
'idpNames' => ['test', 'AAI'],
```

red 35:

```
//ovo je potrebno da bi SAML mogao spremati podatke u session
'routesMiddleware' => ['web'],
```

- Datoteku config/saml2/test_idp_settings.php prekopirati u config/saml2/AAI_idp_settings.php i urediti:

red 5:

```
$this_idp_env_id = 'AAI';
```

red 32:

```
//ovo je oblik u kojem AAI SAML oekuje i dobija podatke
'NameIDFormat' => 'urn:oasis:names:tc:SAMLurn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
```

- u datoteci .env dodati slijedeće:

```
SAML2_AAI_IDP_HOST=https://login.aaiedu.hr/sso/saml2/idp/SSOService.php
SAML2_AAI_IDP_ENTITYID=https://login.aaiedu.hr/sso/saml2/idp/metadata.php
SAML2_AAI_IDP_SSO_URL=https://login.aaiedu.hr/sso/saml2/idp/SSOService.php
SAML2_AAI_IDP_SL_URL=https://login.aaiedu.hr/sso/saml2/idp/SingleLogoutService.php
#SAML2_AAI_IDP_x509='MIIHrD........bla...bla...ostatak_certifikata'
```

certifikat se nalazi u datoteci metadata/saml20-idp-remote.php alata SimpleSamlPHP koji je na poslužitelju .... nije nužan za autentikaciju

- da bi autentikacija ispravno radila potrebno je našu aplikaciju registrirati u Registru resursa sa slijedećim podacima:

entityId: URL_APLIKACIJE/saml2/AAI/metadata
assertionConsumerService URL: URL_APLIKACIJE/saml2/AAI/acs
singleLogoutService URL: URL_APLIKACIJE/saml2/AAI/sls

Nakon što su nam odobreni resursi možemo dalje raditi na razvoju aplikacije, ... autentikacija bez ovog koraka neće ispravno raditi

Potrebno je kreirati listenere, oni služe da se pokrenu u slučaju nekog događaja (u našem slučaju prilikom SAML prijave ili odjave):

- `php artisan make:listener -e Saml2LoginEvent Saml2LoginListener`
- `php artisan make:listener -e Saml2LogoutEvent Saml2LogoutListener`

Ove naredbe stvorit će datoteke app/Listeners/Saml2LoginListener.php i app/Listeners/Saml2LogoutListener.php. Listenere moramo povezati s njihovim eventima u EventServiceProvideru.
U datoteci app/Providers/EventServiceProvider.php dodati:
(prikazani redci 17-27):

```
protected $listen = [
    Registered::class => [
        SendEmailVerificationNotification::class,
    ],
    'Aacotroneo\Saml2\Events\Saml2LoginEvent' => [
        'App\Listeners\Saml2LoginListener',
    ],
    'Aacotroneo\Saml2\Events\Saml2LogoutEvent' => [
        'App\Listeners\Saml2LogoutListener',
    ],
];
```

Zatim je potrebno urediti same listenere:
datoteka app/Listeners/Saml2LoginListener.php funkcija handle:

```
//nakon SAML prijave pokupi podatke o korisniku i spremi ih u session
public function handle(Saml2LoginEvent $event)
{
     $saml2User = $event->getSaml2User();
     $samlAttributes = $saml2User->getAttributes();
     $saml2User = array(
          'aaiedu' => $samlAttributes['hrEduPersonUniqueID'][0],
          'oib' => $samlAttributes['hrEduPersonOIB'][0],
          'name' => $samlAttributes['givenName'][0],
          'surname' => $samlAttributes['sn'][0],
          'email' => $samlAttributes['mail'][0],
     );
     session()->put('saml2user', $saml2User);
}
```

datoteka app/Listeners/Saml2LogoutListener.php funkcija handle:

```
//u sluaju SAML odjave, briše se podatak o korisniku iz sessiona, te se radi odjava i iz laravel autentikacije
public function handle(Saml2LogoutEvent $event)
{
     session()->forget('saml2User');
     Auth::logout();
     Session::save();
}
```

Da bi sama autentikacija mogla raditi bez problema potrebno je još urediti datoteku app/Http/Middleware/VerifyCsrfToken.php:

```
//obzirom da je autentikacija na vanjskoj stranici treba za SAML autentikaciju iskljuiti provjeru CSRF tokena
protected $except = [
     'saml2/AAI/acs',
];
```

Ovime je gotova SAML konfiguracija, autentikacija putem SAML-a je sada moguća, no za potrebe ove upute taj dio preskačem jer istu integrirama sa laravel autentikacijom radi veće kontrole. Laravel autentikacija sprema korisnike u bazu, te je moguće na nju dodati npr. role ili aktivaciju/deaktivaciju i sl.

Integracija sa laravel autentikacijom

Prvo je potrebno kreirati migraciju sa potrebnim parametrima u datoteci database/migrations/2014_10_12_000000_create_users_table.php

```
public function up()
{
     Schema::create('users', function (Blueprint $table) {
          $table->increments('id');
          $table->string('aaiedu')->unique();
          $table->char('oib', 11)->unique();
          $table->string('name');
          $table->string('surname');
          $table->string('email');
          $table->string('password');
          $table->boolean('is_active')->default(true);
          $table->rememberToken();
          $table->timestamps();
     });
}
```

Također je potrebno urediti datoteku app/Models/User.php da bi podatke mogli zapisivati u bazu:

```
protected $fillable = [
     'aaiedu', 'oib', 'name', 'surname', 'email', 'password', 'is_active',
];
```

Nakon provedene pripreme možemo napraviti migraciju:

- `php artisan migrate`

Dodavanje laravel-ide-helper(link is external) komponente

Ova komponenta je čisto opcionalna, i nije nužna, ali olakšava programiranje, te je u niže opisanim radnjama ista korištena u modulu "User" te je bez nje potrebno modificirati source u datoteci app/Http/Controllers/Auth/LoginController.php

- *`composer require --dev barryvdh/laravel-ide-helper`*
- *`php artisan vendor:publish`*

*Odabrati **Provider: Barryvdh\LaravelIdeHelper\IdeHelperServiceProvider***

- *`php artisan config:cache`*
- *`php artisan ide-helper:generate`*
- *`php artisan ide-helper:models -W`*
- *`php artisan ide-helper:meta`*

Gornje četiri naredbe dobro je pokrenuti nakon kreiranja bilo kojeg modela jer nam kreira eloquent "shortcate" za lakše baratanje objektima i database upitima.

Sada je potrebno kreirati custom login kontroler sa funkcijama **login()** i **logout()**, a za to će nam poslužiti datoteka app/Http/Controllers/Auth/LoginController.php koju je laravel već prije kreirao:

```
public function login()
{
     //ukoliko podatak u sessionu postoji SAML autentikacija je obavljena, treba nastaviti na laravel
autentikaciju
     if (!session()->has('saml2user')) {
          try {
               //iniciranje i provedba SAML autentikacije, te povratak na željenu stranicu
               $saml2Auth = new Saml2Auth(Saml2Auth::loadOneLoginAuthFromIpdConfig('AAI'));
               return $saml2Auth->login(session()->pull('url.intended'));
          } catch (Exception $e) {
               //greška u sluaju sa SAML ne radi
               return abort(503);
          }
     } else {
          //uitavanje podataka kreiranih SAML autentikacijom
          $saml2user = session()->pull('saml2user');
          //provjera da li je korisnik ve u bazi
          //----funkcija whereAaiedu je generirana preko laravel-ide-helper komponente
          //----bez nje treba raditi klasian querry npr. DB::table('users')->where('aaiedu',$saml2user['aaiedu'])-
>first();
          $localUser = User::whereAaiedu($saml2user['aaiedu'])->first();
          //ukoliko korisnik nije u bazi, kreiraj ga
          if (!$localUser) {
               $password = bcrypt(Str::random(40));
               $localUser = User::create([
                    'aaiedu' => $saml2user['aaiedu'],
                    'oib' => $saml2user['oib'],
                    'name' => $saml2user['name'],
                    'surname' => $saml2user['surname'],
                    'email' => $saml2user['email'],
                    'password' => $password,
                    'is_active' => true
               ]);
          }
          //obavi laravel autentikaciju i odi na željenu stranicu, kao autenticirani laravel user
          Auth::login($localUser);
          return redirect(redirect()->intended('home'));
     }
}

public function logout()
{
     try {
          //pokreni SAML2 odjavu, listener e obaviti i laravel odjavu
          $saml2Auth = new Saml2Auth(Saml2Auth::loadOneLoginAuthFromIpdConfig('AAI'));
          $saml2Auth->logout(route('welcome'));
     } catch (Exception $e) {
          //greška u sluaju sa SAML ne radi
          return abort(500);
     }
}
```

I zadnji korak: kreiranje web ruta za login i logout na našem projektu u datoteci routes/web.php:

```
use App\Http\Controllers\Auth\LoginController;

...

//Auth::routes();
Route::get('/login', [LoginController::class, 'login'])->name('login');
Route::any('/logout', [LoginController::class, 'logout'])->name('logout');
```

Ovim korakom autentikacija je implementirana te funkcionira u standardnim pogledima koji dolaze sa laravelom.