

# Embedding Process

- [Embedding dashboard in laravel](#)
- [RLS in Embeeding](#)

# Embedding dashboard in laravel

## Setup env

```
SUPERSET_URL=domain_name
SUPERSET_USERNAME=admin_username
SUPERSET_PASSWORD=admin_password
SUPERSET_DASHBOARD_ID=id_of_dashboard
```

## Create a SupersetService in `app/Services/SupersetService.php`:

```
<?php

namespace App\Services;

use Illuminate\Support\Facades\Http;

class SupersetService
{
    protected $baseUrl;
    protected $username;
    protected $password;
    protected $dashboardId;

    public function __construct()
    {
        $this->baseUrl = env('SUPERSET_URL');
        $this->username = env('SUPERSET_USERNAME');
        $this->password = env('SUPERSET_PASSWORD');
        $this->dashboardId = env('SUPERSET_DASHBOARD_ID');
    }

    public function login()
    {

```

```

$response = Http::log()->post("{ $this->baseUrl}/api/v1/security/login", [
    'username' => $this->username,
    'password' => $this->password,
    'provider' => 'db',
]);

return $response->json('access_token');
}

public function getGuestToken()
{
    $token = $this->login();

    $response = Http::withHeaders([
        'Content-Type' => 'application/json'
    ])->log()
    ->withToken($token)
    ->post("{ $this->baseUrl}/api/v1/security/guest_token", [
        'user' => [
            'username' => "",
            'first_name' => "",
            'last_name' => ""
        ],
        'resources' => [
            [
                'type' => 'dashboard',
                'id' => $this->dashboardId
            ]
        ],
        'rls' => []
    ]);

    if ($response->successful()) {
        return $response->json('token');
    }

    throw new \Exception('Failed to get guest token: ' . $response->body());
}
}

```

## in controller setup the process to get guest token

```
public function info(Request $request)
{
    $supersetService = new SupersetService();
    $token = $supersetService->getGuestToken();
    return view('info', [
        'token' => $token,
    ]);
}
```

## in the blade setup the js

```
<script src="https://unpkg.com/@superset-ui/embedded-sdk"></script>

<div id="dashboard" width="100%" height="500vh" class="p-6"> </div>

<script>
    supersetEmbeddedSdk.embedDashboard({
        id: '{{ env('SUPERSET_DASHBOARD_ID') }}',
        supersetDomain: '{{ env('SUPERSET_URL') }}',
        mountPoint: document.getElementById('dashboard'),
        fetchGuestToken: () => '{{ $token }}',
        dashboardUiConfig: {
            hideTitle: true,
            hideTab: true,
            hideChartControls: true,
        },
    });
    document.getElementById("dashboard").children[0].width="100%";
    document.getElementById("dashboard").children[0].height="1200vh";
    document.getElementById("dashboard").children[0].style.border="none";
    document.getElementById("dashboard").children[0].style.overflow ="hidden";
</script>
```

# RLS in Embedding

we can directly define RLS when creating guesttoken

```
public function getGuestToken()
{
    $token = $this->login();

    $response = Http::withHeaders([
        'Content-Type' => 'application/json'
    ]->log()
    ->withToken($token)
    ->post("{ $this->baseUrl}/api/v1/security/guest_token", [
        'user' => [
            'username' => "",
            'first_name' => "",
            'last_name' => ""
        ],
        'resources' => [
            [
                'type' => 'dashboard',
                'id' => $this->dashboardId
            ]
        ],
        'rls' => [
            ['clause'=> 'fakultas = "Fakultas Kedokteran"']
        ],
    ]);

    if ($response->successful()) {
        return $response->json('token');
    }

    throw new \Exception('Failed to get guest token: ' . $response->body());
}
```