REST API & PASSPORT

Certainly! Creating a REST API with Laravel and integrating Passport for authentication involves several steps. I'll guide you through the process with a detailed tutorial.

Step 1: Install Laravel

Make sure you have Composer installed, and then run the following command to create a new Laravel project:

```bash
composer create-project --prefer-dist laravel/laravel your-api-project
cd your-api-project
```

Step 2: Install Passport

Install Laravel Passport using Composer:

```bash
composer require laravel/passport
```

```

**Step 3: Set up Database**

Configure your database connection in the `.env` file and run migrations:

```bash
php artisan migrate
```

Step 4: Passport Installation

Run the following commands to set up Passport:

```bash
php artisan passport:install
```

This will create encryption keys and necessary database tables.

Step 5: Configuration
```

In your `config/auth.php` file, make sure the 'api' guard uses the 'passport' driver:

```php
'guards' => [
    'api' => [
        'driver' => 'passport',
        'provider' => 'users',
    ],
],
```

Step 6: User Model

Make sure your User model uses the `Laravel\Passport\HasApiTokens` trait:

```php
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Passport\HasApiTokens;
```

```
class User extends Authenticatable
{
    use HasApiTokens, Notifiable;
    // ...
}
```

Step 7: API Routes

Define your API routes in the `routes/api.php` file:

```php
Route::middleware('auth:api')->group(function () {
    // Your authenticated routes go here
});

Route::post('/login', 'AuthController@login');
Route::post('/register', 'AuthController@register');
```

Step 8: AuthController

Create a controller for authentication:

```bash
php artisan make:controller AuthController
```

In `AuthController.php`:

```php
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    public function register(Request $request)
    {
        // Your registration logic here
    }

    public function login(Request $request)
    {
        // Your login logic here
```

```
    }
}
```

Step 9: Implement Authentication Logic

Implement your registration and login logic in the `AuthController`. Use Passport's `createToken` method for token generation.

Step 10: Testing

Use a tool like Postman to test your API. Register a user, obtain a token, and make authenticated requests.

Additional Tips:

- Include error handling and validation in your controller methods.
- You may need to set up CORS if your frontend is on a different domain.
- Secure your API by using HTTPS.

This is a basic setup. Depending on your project requirements, you might need to implement additional features like user roles, scopes, etc. Refer to the official Laravel and Passport documentation for more details: [Laravel Documentation](https://laravel.com/docs) and [Passport Documentation](https://laravel.com/docs/passport).