

Creating project and configuring database:

At first we have to create the project by following command in the CLI:

```
composer create-project --prefer-dist laravel/laravel todo-api
```

Now we need to configure the database from '.env' file of the project.

JWT installation, publishing the configuration file and creating secret key:

Now we have to install JWT package for our project by running following command in the terminal:

```
composer require tyson/jwt-auth
```

Now need to publish the configuration file by following command in the terminal:

```
php artisan vendor:publish --provider="Tyson\JWTAuth\Providers\LaravelServiceProvider"
```

Now need to generate secret key by following command in the terminal:

```
php artisan jwt:secret
```

Now need to edit the config/auth.php file to use the JWT guard and provider as follow:

```
'guards' => [  
    'api' => [  
        'driver' => 'jwt',  
        'provider' => 'users',  
    ],  
],  
  
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\User::class,  
    ],  
],
```

Generating migration and updating migration files:

Now we need to create a migration file by following command:

```
php artisan make:migration create_todos_table
```

Now we need to update the 'up' method of the table '{timestamp}_create_users_table' and '{timestamp}_create_todos_table' as follow:

For '{timestamp}_create_users_table':

```
public function up()  
  
{  
  
Schema::create('users', function (Blueprint $table) {  
  
    $table->id();  
  
    $table->string('name');  
  
    $table->string('email')->unique();  
  
    $table->string('password');  
  
    $table->timestamps();  
  
});  
  
}
```

For '{timestamp}_create_todos_table':

```
public function up()  
  
{  
  
Schema::create('todos', function (Blueprint $table) {  
  
    $table->id();  
  
    $table->unsignedBigInteger('user_id');  
  
    $table->string('title');  
  
    $table->text('description');  
  
    $table->boolean('completed')->default(false);
```

```
$table->timestamps();
```

```
$table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
```

```
});
```

```
}
```

Now we need to migrate the files by running following command in the terminal:

```
php artisan migrate
```

Creating models for users and todos:

Now we need to create models for users and todos by running following command in the terminal:

```
php artisan make:model User
```

```
php artisan make:model Todo
```

Now we have to edit the generated model files User.php and Todo.php as follow:

For User.php:

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Foundation\Auth\User as Authenticatable;
```

```
use Tymon\JWTAuth\Contracts\JWTSubject;
```

```
class User extends Authenticatable implements JWTSubject
```

```
{
```

```
    public function getJWTIdentifier()
```

```
    {
```

```
        return $this->getKey();
```

```
}  
  
public function getJWTCustomClaims()  
  
{  
  
    return [];  
  
}  
  
}
```

For Todo.php:

```
<?php
```

```
app/Models/Todo.php
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Todo extends Model
```

```
{  
  
    public function user()  
  
    {  
  
        return $this->belongsTo(User::class);  
  
    }  
  
}
```

Generating controllers for users and todos:

Now we have to generate the controllers for users and todos by running following command in the terminal:

```
php artisan make:controller UserController
```

```
php artisan make:controller TodoController
```

Now we need to edit the UserController and TodoController as follow:

For UserController:

```
<?php
```

```
use App\Models\User;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Routing\Controller;
```

```
use Illuminate\Support\Facades\Hash;
```

```
use Illuminate\Validation\ValidationException;
```

```
use Tymon\JWTAuth\Facades\JWTAuth;
```

```
class UserController extends Controller
```

```
{
```

```
    public function login(Request $request)
```

```
    {
```

```
        $credentials = $request->validate([
```

```
            'email' => 'required|email',
```

```
            'password' => 'required|string',
```

```
        ]);
```

```
        if (!$token = JWTAuth::attempt($credentials)) {
```

```
            throw ValidationException::withMessages([
```

```
                'email' => 'Invalid credentials',
```

```

        ]);
    }

    return response()->json(['token' => $token]);
}
}

```

For TodoController:

```

<?php

namespace App\Http\Controllers\TodoControllers;

use App\Models\Todo;
use Illuminate\Http\Request;
use Illuminate\Routing\Controller;

class TodoController extends Controller
{
    public function index()
    {
        $user = auth()->user();

        $todos = $user->todos;

        return response()->json($todos);
    }

    public function store(Request $request)
    {
        $user = auth()->user();
    }
}

```

```

        $todo = $user->todos()->create($request->all());

        return response()->json($todo, 201);
    }

    public function update(Request $request, \Todo $todo)
    {
        $user = auth()->user();

        $this->authorize('update', $todo);

        $todo->update($request->all());

        return response()->json($todo);
    }

    public function destroy(\Todo $todo)
    {
        $user = auth()->user();

        $this->authorize('delete', $todo);

        $todo->delete();

        return response()->json(null, 204);
    }
}

```

Define routes:

Now we need to define the routes in the 'routes/api.php' file as follow:

```

Route::post('login', 'UserController@login');

Route::middleware('auth:api')->group(function () {
    Route::resource('todos', 'TodoController')->except(['create', 'edit']);
});

```