

Git Workflow With Laravel Team Examples

===== 1. Laravel Controller Example ===== Example: UserController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index()
    {
        return User::all();
    }

    public function store(Request $request)
    {
        $request->validate([
            'name' => 'required',
            'email' => 'required|email'
        ]);

        return User::create($request->all());
    }
}
?>
```

===== 2. Git Workflow Using Laravel Tasks

===== Example scenario inside a team: Task: Add new user endpoint
Steps: 1. Update your main branch: git checkout main git pull origin main 2. Create feature branch: git checkout -b feature/add-user-endpoint 3. Add your Laravel code in controller, routes, models, etc. 4. Stage and commit: git add . git commit -m "add user endpoint with validation" 5. Push branch: git push origin feature/add-user-endpoint 6. Open Pull Request on GitHub: - Title: "Add user creation endpoint" - Review with team - When approved → merge to main

===== 3. Example: Migration Update

===== Suppose one developer works on users table migration. Laravel migration example:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up()
    {
        Schema::table('users', function (Blueprint $table) {
            $table->string('phone')->nullable();
        });
    }

    public function down()
    {
        Schema::table('users', function (Blueprint $table) {
            $table->dropColumn('phone');
        });
    }
};
?>
```

===== 4. How Conflicts Happen in Laravel Project

===== Example: - You modify routes/api.php - Teammate modifies same line Git will show: CONFLICT (content): Merge conflict in routes/api.php You fix it manually, then: git add routes/api.php git commit -m "fix merge conflict" git push