

# ACID-Compliant CRUD Operations in Laravel

Implementing ACID-compliant CRUD operations in Laravel ensures database transactions are reliable.

## 1. Set Up Migration & Model

-----

Run the following command to create a model and migration:

```
php artisan make:model Product -m
```

Edit the migration file:

```
public function up() {  
    Schema::create('products', function (Blueprint $table) {  
        $table->id();  
        $table->string('name');  
        $table->decimal('price', 10, 2);  
        $table->timestamps();  
    });  
}
```

Run the migration:

```
php artisan migrate
```

Edit the model:

```
class Product extends Model {  
    protected $fillable = ['name', 'price'];  
}
```

## 2. Implement CRUD Operations with Transactions

-----

Create a controller:

```
php artisan make:controller ProductController
```

Create a Product (Atomicity & Consistency):

```
public function store(Request $request) {  
    DB::beginTransaction();  
    try {
```

```

$product = Product::create([
    'name' => $request->name,
    'price' => $request->price
]);
DB::commit();

return response()->json(['message' => 'Product created', 'product' => $product]);
} catch (\Exception $e) {
    DB::rollBack();
    return response()->json(['error' => 'Failed to create product'], 500);
}
}

```

Read Products:

```

public function index() {
    return response()->json(Product::all());
}

```

Update a Product (Ensuring Data Integrity):

```

public function update(Request $request, $id) {
    DB::beginTransaction();
    try {
        $product = Product::findOrFail($id);
        $product->update([
            'name' => $request->name,
            'price' => $request->price
        ]);
        DB::commit();
        return response()->json(['message' => 'Product updated', 'product' => $product]);
    } catch (\Exception $e) {
        DB::rollBack();
        return response()->json(['error' => 'Failed to update product'], 500);
    }
}

```

Delete a Product (Rollback on Failure):

```

public function destroy($id) {
    DB::beginTransaction();
    try {

```

```
$product = Product::findOrFail($id);  
$product->delete();  
DB::commit();  
return response()->json(['message' => 'Product deleted']);  
} catch (\Exception $e) {  
    DB::rollBack();  
    return response()->json(['error' => 'Failed to delete product'], 500);  
}  
}
```

### 3. Routes (web.php)

-----

```
use App\Http\Controllers\ProductController;  
Route::post('/products', [ProductController::class, 'store']);  
Route::get('/products', [ProductController::class, 'index']);  
Route::put('/products/{id}', [ProductController::class, 'update']);  
Route::delete('/products/{id}', [ProductController::class, 'destroy']);
```