L'intégration entre **TP4 (Backend Laravel API)** et **TP2 (Frontend statique HTML/CSS/JS)** se fait **exclusivement via des requêtes HTTP** — le frontend consomme l'API Laravel comme un service externe.

Voici comment cela fonctionne **pratiquement**, **étape par étape**, **sans modifier le code du TP2** (vanilla JS).

---

1. **Architecture globale**

```
[TP2 - Frontend statique]  <--->  Requêtes HTTP (fetch)  <--->  [TP4 - API Laravel]
       (index.html, panier.js, etc.)
(routes/api.php)
```

- **TP2** : Pages HTML + JS dans `/tp2/` ou `/public/`
- **TP4** : API Laravel dans `/tp4/`, serveur sur `http://localhost:8000`
- **CORS activé** sur Laravel → autorise les requêtes depuis `http://localhost:5500` (Live Server)

---

2. **Configuration CORS (Laravel)**

Dans `app/Http/Kernel.php` :

```
protected $middleware = [
    // ...
    \Fruitcake\Cors\HandleCors::class,
];
```

Dans `config/cors.php` :

```
'paths' => ['api/*'],
'allowed_methods' => ['*'],
'allowed_origins' => ['http://localhost:5500'], // Live Server TP2
'allowed_headers' => ['*'],
'supports_credentials' => true,
```

---

3. **Authentification : Connexion depuis TP2**

TP2 (`login.html` + `auth.js`)

```
const login = async (email, password) => {
  const res = await fetch('http://localhost:8000/api/v1/login', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
```

```
    body: JSON.stringify({ email, password })
  });

  const data = await res.json();
  if (data.token) {
    localStorage.setItem('token', data.token);
    window.location.href = 'index.html';
  }
};
```

Laravel (AuthController.php)

```
public function login(Request $request)
{
    $credentials = $request->only('email', 'password');
    if (Auth::attempt($credentials)) {
        $user = Auth::user();
        $token = $user->createToken('auth_token')->plainTextToken;
        return response()->json(['token' => $token, 'user' => $user]);
    }
    return response()->json(['error' => 'Unauthorized'], 401);
}
```

---

4. **Panier : Synchronisation avec l'API**

TP2 (panier.js)

```
// Ajouter au panier
const addToCart = async (variantId, qty = 1) => {
  const token = localStorage.getItem('token');
  await fetch('http://localhost:8000/api/v1/cart/items', {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${token}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({ product_variant_id: variantId, quantity: qty })
  });
  updateCartUI();
};
```

Laravel (CartController.php)

```
public function addItem(Request $request)
{
```

```
    $user = $request->user();
    $cart = $user->cart ?? $user->cart()->create([]);

    $cart->items()->updateOrCreate(
        ['product_variant_id' => $request->product_variant_id],
        ['quantity' => DB::raw('quantity + ' . $request->quantity)]
    );

    return response()->json($cart->load('items.variant'));
}
```

---

5. **Chargement des produits dynamiquement**

TP2 (`products.html`)

```
<div id="product-grid"></div>

<script>
  fetch('http://localhost:8000/api/v1/products')
    .then(r => r.json())
    .then(products => {
      document.getElementById('product-grid').innerHTML =
products.data.map(p => `
        <div class="product-card">
          <img src="http://localhost:8000/storage/${p.variant.image}">
          <h3>${p.name}</h3>
          <p>${p.variant.price} FCFA</p>
          <button onclick="addToCart(${p.variant.id})">Ajouter</button>
        </div>
      `).join('');
    });
</script>
```

---

6. **Checkout & Paiement**

TP2 (`checkout.js`)

```
const checkout = async () => {
  const token = localStorage.getItem('token');
  const res = await fetch('http://localhost:8000/api/v1/payments', {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${token}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      card_number: '4242424242424242',
```

```
        exp_month: 12,
        exp_year: 2026,
        cvc: '123'
      })
    });

    const result = await res.json();
    if (result.success) {
      alert('Paiement réussi !');
      window.location.href = 'confirmation.html';
    }
  };
```

---

7. **Lancement en parallèle**

Terminal 1 — TP4 (API)

```
cd tp4
php artisan serve
# → http://localhost:8000
```

Terminal 2 — TP2 (Frontend)

```
cd tp2
npx live-server
# → http://localhost:5500
```

---

8. **Résumé des flux**

Action TP2	Requête API	Endpoint Laravel
Connexion	`POST /api/v1/login`	`AuthController@login`
Voir produits	`GET /api/v1/products`	`ProductController@index`
Ajouter au panier	`POST /api/v1/cart/items`	`CartController@addItem`
Voir panier	`GET /api/v1/cart`	`CartController@show`
Paiement	`POST /api/v1/payments`	`PaymentController@process`
Créer commande	`POST /api/v1/orders`	`OrderController@store`

---

9. **Bonus : Postman pour tester l'API**

- Importez la **collection Postman** générée par **K**

- Testez tous les endpoints
- Copiez les `cURL` → collez dans TP2 JS

En résumé

Élément	TP2	TP4
Code	HTML + JS vanilla	Laravel API
Stockage panier	`localStorage` → synchronisé avec API	Table `cart` + `cart_item`
Auth	Token dans `localStorage`	Sanctum + `personal_access_tokens`
Images	`http://localhost:8000/storage/...`	`php artisan storage:link`
Communication	`fetch()`	`json()` responses

Aucun changement dans le HTML/CSS du TP2
Tout le dynamisme vient de l'API TP4