how to manual data send laravel api to facebook pixel simple project create

This is a simple Laravel project to demonstrate how to manually send data to Facebook Pixel using the Conversions API (CAPI) from an API endpoint

.

Project overview

You will create a small API with one endpoint. When this endpoint is triggered, it will send a Purchase event to your Facebook Pixel via a server-to-server request. This simulates a back-end process, like an order confirmation, triggering a conversion event

1. composer require guzzlehttp/guzzle

2. FACEBOOK_PIXEL_ID=your_pixel_id
FACEBOOK_CONVERSIONS_API_TOKEN=your_access_token
FACEBOOK_TEST_EVENT_CODE=your_test_event_code

Step 1: Set up Facebook Conversions API

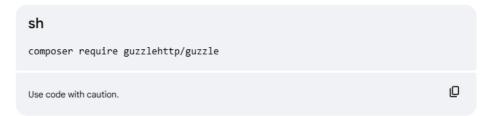
- 1. Generate an access token:
 - · Go to your Meta Events Manager.
 - Select your Pixel, navigate to the Settings tab, and find the Conversions API section.
 - · Click Generate access token and copy the value.
- 2. Generate a test event code:
 - · Still in Events Manager, go to the Test Events tab.
 - Copy the test_event_code to test your integration without affecting your production data.

Step 2: Set up a new Laravel project

1. Create a new Laravel application:



2. Install Guzzle HTTP client: This library will be used to make the API request to Facebook.



3. Configure environment variables: Add your Facebook credentials to your .env file.

```
ini

FACEBOOK_PIXEL_ID="your_pixel_id"

FACEBOOK_CONVERSIONS_API_TOKEN="your_access_token"

FACEBOOK_TEST_EVENT_CODE="your_test_event_code"

Use code with caution.
```

4. **Update** config/services.php: Add a new entry to make your Facebook credentials available in your application.

```
php

return [
    // ... other services
    'facebook' => [
        'pixel_id' => env('FACEBOOK_PIXEL_ID'),
        'conversions_api_token' => env('FACEBOOK_CONVERSIONS_API_TOKEN'),
        'test_event_code' => env('FACEBOOK_TEST_EVENT_CODE'),
    ],
];

Use code with caution.
```

Step 3: Create the Facebook service class

Create a new service class to encapsulate the API logic. This keeps your controller clean and the logic reusable.

1. Create the directory:



Create the FacebookApiService.php file: app/Services/FacebookApiService.php

```
php
<?php
namespace App\Services;
use GuzzleHttp\Client;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Str;
class FacebookApiService
    protected $client;
    protected $pixelId;
    protected $accessToken;
    protected $testEventCode;
    public function __construct()
       $this->client = new Client([
            'base_uri' => 'https://graph.facebook.com/v19.0/',
       ]);
       $this->pixelId = config('services.facebook.pixel_id');
       $this->accessToken = config('services.facebook.conversions api token');
        $this->testEventCode = config('services.facebook.test_event_code');
```

Step 4: Create the API controller and route

1. Create the controller:

```
sh

php artisan make:controller Api/FacebookEventController

Use code with caution.
```

2. Define the controller logic: Inside

app/Http/Controllers/Api/FacebookEventController.php, add a method to trigger the event.

```
php
<?php
namespace App\Http\Controllers\Api;
use App\Http\Controllers\Controller;
use App\Services\FacebookApiService;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
class FacebookEventController extends Controller
    protected $facebookApi;
    public function __construct(FacebookApiService $facebookApi)
        $this->facebookApi = $facebookApi;
    }
    public function trackPurchase(Request $request)
    {
       // Dummy data to simulate a purchase
        $userData = [
```

3. Define the API route: Inside routes/api.php, add a new endpoint.



Step 5: Test the project

1. Start your Laravel server:

