**Давайте подробно разберем, как интегрировать AI21 Studio в ваш проект Laravel для генерации литературных предложений из указанных слов с использованием бесплатного уровня.**

### Шаг 1: Регистрация и получение API ключа

1. Зарегистрируйтесь на [AI21 Studio](https://www.ai21.com/studio).
2. Получите ваш API ключ после регистрации.

### Шаг 2: Установка библиотеки Guzzle

Для отправки HTTP запросов мы будем использовать библиотеку Guzzle. Установите её с помощью Composer:

composer require guzzlehttp/guzzle

для перевода на русский

composer require stichoza/google-translate-php

### Шаг 3: Создание контроллера для генерации текста

php artisan make:controller TextGenerationController

**Шаг 4: Реализация контроллера**

<?php  
namespace App\Http\Controllers;  
  
use App\Http\Requests\AIStudio\GenerateSentencesRequest;  
use App\Http\Responses\ApiResponse;  
use GuzzleHttp\Client;  
use GuzzleHttp\Exception\GuzzleException;  
use Illuminate\Http\Request;  
use Stichoza\GoogleTranslate\GoogleTranslate;  
  
class GeneratingSentencesAiController extends Controller  
{  
  
 */\*\*  
 \* Генерирует предложения  
 \** ***@param*** *Request $request  
 \** ***@return*** *ApiResponse  
 \** ***@throws*** *GuzzleException  
 \*/* public function generateSentence(GenerateSentencesRequest $request): ApiResponse  
 {  
 // масив слов которые будут в предложениях  
 $words = $request->arr\_words;  
 $sentences = [];  
  
 foreach ($words as $word) {  
 $response = $this->getSentencesAIStudio($word);  
 $individualSentences = $this->processResponse($response);  
  
 foreach ($individualSentences as $sentence) {  
 if (strpos($sentence, 'Error') === false && strpos($sentence, 'Failed') === false) {  
 // Переводим предложение на русский язык  
 $translatedSentence = $this->translateToRussian($sentence);  
 // Добавляем в массив результатов  
 $sentences[] = [  
 'original' => $sentence,  
 'translated' => $translatedSentence,  
 ];  
 }  
 }  
 }  
  
 return new ApiResponse(compact('sentences'));  
 }  
  
 */\*\*  
 \* Запрос на AI21 Studio и получение ответа  
 \** ***@param*** *$word  
 \** ***@return*** *string  
 \** ***@throws*** *GuzzleException  
 \*/* private function getSentencesAIStudio($word): string  
 {  
 $client = new Client();  
 // Поместите ваш API ключ в файл .env  
 $apiKey = env('STUDIO\_AI\_API\_KEY');  
 $prompt = 'Create five simple literary sentences that are not similar to each other in English and separate them with a vertical symbol | using the given word: ' . $word;  
  
 try {  
 // Используем правильный URL-адрес для AI21 Studio API  
 $response = $client->post('https://api.ai21.com/studio/v1/j2-ultra/complete', [  
 'headers' => [  
 'Authorization' => 'Bearer ' . $apiKey,  
 'Content-Type' => 'application/json',  
 ],  
 'json' => [  
 'prompt' => $prompt,  
 'numResults' => 1,  
 'maxTokens' => 100, // Ограничиваем количество токенов  
 'temperature' => 0.7,  
 'topP' => 0.9,  
 ],  
 ]);  
  
 $data = json\_decode($response->getBody(), true);  
 // Убираем лишние пробелы и переносы строк  
 return trim($data['completions'][0]['data']['text']) ?? 'Failed to generate sentence.';  
 } catch (\Exception $e) {  
 // Обрабатываем ошибки  
 return 'Error: ' . $e->getMessage();  
 }  
 }  
  
 */\*\*  
 \* Обрабатывает ответ API и извлекает предложения  
 \** ***@param*** *$response  
 \** ***@return*** *array  
 \*/* private function processResponse(string $response): array  
 {  
 // Разделяем предложения по разделителю "|"  
 $sentencesArray = explode('|', $response);  
  
 // Очищаем каждое предложение от лишних символов и цифр  
 $cleanedSentences = [];  
 foreach ($sentencesArray as $sentence) {  
 // Удаляем цифры и лишние пробелы в начале и конце  
 $cleanedSentence = trim(preg\_replace('/\d+\.\s\*/', '', $sentence));  
 if (!empty($cleanedSentence)) {  
 $cleanedSentences[] = $cleanedSentence;  
 }  
 }  
  
 return $cleanedSentences;  
 }  
  
 */\*\*  
 \* Переводит текст на русский язык  
 \** ***@param*** *string $text  
 \** ***@return*** *string  
 \*/* private function translateToRussian(string $text): string  
 {  
 // Инициализируем объект для перевода  
 $translator = new GoogleTranslate();  
 // Устанавливаем исходный и целевой языки  
 $translator->setSource('en'); // Исходный язык - английский  
 $translator->setTarget('ru'); // Целевой язык - русский  
  
 // Переводим текст  
 return $translator->translate($text);  
 }  
}

async loadGenerateSentences(){  
 let data = {  
 arr\_words: ***Array***.isArray(this.new\_word) ? this.new\_word : [this.new\_word],  
 };  
  
 this.arrGenerateSentences = [  
 { original: "The happy couple married on the beach.", translated: "Счастливая пара поженилась на пляже." },  
 { original: "The prince married a beautiful princess.", translated: "Принц женился на прекрасной принцессе." },  
 { original: "They married in secret, away from prying eyes.", translated: "Они поженились тайно, вдали от посторонних глаз." },  
 { original: "The lonely man married his cat, but no one else knew.", translated: "Одинокий мужчина женился на своей кошке, но больше никто об этом не знал." },  
 { original: "The wizard married his broomstick, and everyone thought it was strange.", translated: "Волшебник женился на своей метле, и всем показалось это странным." }  
 ];  
  
 // try {  
 // const response = await this.$http.post(`${this.$http.apiUrl()}ai/generate-sentences`, data);  
 // if(this.checkSuccess(response)){  
 // console.log(response.data.data.sentences)  
 // this.arrGenerateSentences = response.data.data.sentences  
 //  
 // // Выводим массив объектов в цикле  
 // this.arrGenerateSentences.forEach((sentence, index) => {  
 // console.log(`Sentence ${index + 1}:`);  
 // console.log(`Original: ${sentence.original}`);  
 // console.log(`Translated: ${sentence.translated}`);  
 // console.log('---');  
 // });  
 //  
 // }  
 // } catch (e) {  
 // console.log(e);  
 // }  
},

### Шаг 5: Настройка маршрутов

Добавьте маршрут для вашего нового контроллера в файле routes/web.php:

use App\Http\Controllers\TextGenerationController;  
  
Route::post('/generate', [TextGenerationController::class, 'generate']);

### Шаг 6: Настройка .env файла

Добавьте ваш API ключ в файл .env:

STUDIO\_AI\_API\_KEY=your\_ai21\_api\_key