

Есть текст вот такого плана

<https://docs.google.com/document/d/1OJVerOFD9ZvWkq4ESGRVI0FkgJLdxScgseCFmXCRlw/edit?usp=sharing>

Синонимы разделены символом “|”

Для каждого языка может быть до 10 таких текстов. Такие текста будут для 40 языков.

Задача - написать скрипт который сможет сгенерировать текст на основании этого шаблона.

Текст генериться путем выбора одного из синонимов в скобках.

Например для вот этого куска:

```
<p>If you {need to|have to|are requested to|are required to} {buy|purchase|acquire} [coin1_long_name] ([coin1_short_name]) with [coin2_long_name] ([coin2_short_name]), the {fastest|quickest} and {simplest|easiest} way to do it is {through|via} {services|platforms|websites} like [Service_name]. With {such|this kind of|this sort of} {services|solutions|resources} you can {purchase|get|buy} [coin1_long_name] and {other|alternative} popular {cryptocurrencies|crypto assets|digital coins} with one of the 80 supported {fiat|traditional} currencies.</p>
```

Получиться в одном случае текст

```
If you need to buy [coin1_long_name] ([coin1_short_name]) with [coin2_long_name] ([coin2_short_name]), the quickest and easiest way to do it is via websites like [Service_name]. With this sort of resources you can buy [coin1_long_name] and alternative popular digital coins} with one of the 80 supported traditional currencies.</p>
```

Либо такой

```
<p>If you are required to purchase [coin1_long_name] ([coin1_short_name]) with [coin2_long_name] ([coin2_short_name]), the fastest and easiest way to do it is through platforms like [Service_name]. With such solutions you can buy [coin1_long_name] and other popular cryptocurrencieswith one of the 80 supported fiat currencies.</p>
```

Пример возможной реализации скрипта:

```
<?php
// Function to generate text from a spin syntax string
function spinText($text) {
```

```

$pattern = '/{([^{]*)}/'; // Match content inside {}

while (preg_match($pattern, $text, $matches)) {
    // Split options by the pipe character |
    $options = explode('|', $matches[1]);
    // Pick a random option
    $replacement = $options[array_rand($options)];
    // Replace the matched spin syntax with the random option
    $text = preg_replace($pattern, $replacement, $text, 1);
}

return $text;
}

// Input text with spin syntax
$inputText = <<<TEXT
<h2>Купить [coin1_long_name] ([coin1_short_name]) за [coin2_long_name]
([coin2_short_name])</h2>
<p>Если вам {нужно|необходимо|требуется|понадобится} {купить|приобрести|закупить}
[coin1_long_name] ([coin1_short_name]) за [coin2_long_name] ([coin2_short_name]), то
{самый быстрый|наиболее быстрый} и {простой|легкий} способ сделать это — {через|с
помощью|используя} {сервисы|платформы|сайты} такие, как [Service_name]. С помощью
{таких|подобных} {сервисов|решений|ресурсов} вы можете {приобрести|купить|получить}
[coin1_long_name] и {другие|альтернативные} популярные {криптовалюты|цифровые
активы|цифровые монеты} с одной из 80 поддерживаемых {фиатных|традиционных}
валют.</p>
TEXT;

// Generate the final text
$generatedText = spinText($inputText);

// Output the generated text
echo $generatedText;
?>

```

Скрипт должен быть реализован в виде API которые мы будем вызывать из другой системы.

Ряд дополнительных требований:

- 1) Сгенерированный текст будет содержать переменные, формат переменных - [variable_name].

Примерный список переменных :

- [asset1_short_name], [asset2_short_name] - короткое название монеты или фиата
- [asset1_long_name], [asset2_long_name] - полное название монеты или фиата
- [asset1_price], [asset2_price] - цена монеты 1 и 2 в \$
- [rate] - текущий курс монеты 1 к монете 2
- [asset1_market_cap], [asset2_market_cap] - market cap монеты 1 и 2
- [asset1_24h_trading_volume], [asset2_24h_trading_volume] - trading_volume монеты 1 и 2
- [asset1_circulating_supply] [asset2_circulating_supply] - circulating supply монеты 1 и 2
- [current_day] - текущая дата
- [current_year] - текущий год

2) Предлагаемый формат API:

Запрос генерации контента - обращение к url-адресу:

`https://service-domain/add/{page-type}/{locale}/{asset1_short_name}-{asset2_short_name}/{ton_of_voice},`

где {page-type} - тип страницы (crypto-crypto, crypto-fiat, fiat-crypto),
{locale} - локаль языковой версии и региона, например en или en-GB, где первая часть локали определяет код языка (в формате [ISO 639-1](#)), вторая необязательная часть определяет код региона (в формате [ISO 3166-1 Alpha 2](#)). {asset1_short_name}, {asset2_short_name} - короткое имя монеты или фиата
{ton_of_voice} (optional) - "тон текста". Если параметр не задан - будет выбран случайный вариант:

- confident – conveys a sense of security and reliability.
- modern – highlights the platform’s innovation and relevance.
- trustworthy – builds user confidence in handling financial transactions.
- transparent – emphasizes honesty and no hidden fees.
- accessible – makes the platform inviting and easy to use.
- fast – showcases the speed of transaction processing.
- user-ariendly – focuses on ease of navigation and simplicity.
- secure – reassures users of data and fund protection.
- efficient – underscores an optimized exchange process.
- flexible – offers a wide range of currencies and exchange options.
- random (by default) - random from above

пример: при добавлении страницы btc-to-eth с локалью en-GB - обращаемся к URL:

`https://service-domain/add/crypto-crypto/en-GB/btc-eth`

при добавлении страницы btc-to-usd с локалью en-AU - обращаемся к URL:
<https://service-domain/add/crypto-fiat/en-AU/btc-usd>

После обращения будет сгенерирован контент для данной пары и данной языковой версии.

Ответ API будет содержать статус (ok или error), сам сгенерированный текст и контент для секции FAQ с метками локали, теги title и description, дату генерации в формате [unix time](#). Между данными секциями будет добавлен разделитель.

Предлагаемый формат JSON-объекта, который будет возвращать система:

```
{
  "status": "ok",
  "error": null,
  "title_en-GB": "seo generated title tag with variables in square brackets ([asset1_long_name] - example of variable).",
  "meta_description_en-GB": "seo generated description metatag with variables in square brackets ([asset1_long_name] - example of variable).",
  "content_en-GB": "seo generated text content in english for the United Kingdom page version will be here with variables in square brackets ([asset1_price] - example of variable).",
  "faq_en-GB": [
    {
      "question": "here will be question 1",
      "answer": "here will be answer for question 1"
    },
    {
      "question": "here will be question 2",
      "answer": "here will be answer for question 2"
    },
    ... ,
    {
      "question": "here will be question n",
      "answer": "here will be answer for question n"
    }
  ],
  "last-modified": "2024-11-07T16:45:15.44895+00:00"
}
```

3) Генерация по спин-шаблону

На вход подаем шаблоны генерации спин-текста, состоящие из:

- синонимов, помещенных в фигурные скобки и разделенных символом | (пример - {buy|purchase|acquire}). При генерации синонимы из каждого множества должны выбираться случайным образом;
- переменных, помещенных в квадратные скобки (пример - [asset1_long_name], [asset1_short_name], [asset1_price_usd]). Данные переменные в ходе генерации текстов не меняются и передаются на сайт в исходном виде). При отображении контента на страницы переменные будут заменяться конкретными значениями.

При генерации текста из шаблона, синонимы из каждого имеющегося в шаблоне множества должны выбираться случайно. То есть, тексты, полученные в результате генерации, должны быть максимально уникальными.

Для каждого типа страниц и языковой версии должен быть задан свой спин-шаблон, по которому будет происходить генерация. Также должны быть отдельные шаблоны для генерации title и description.

Примеры:

- [Пример шаблонов спин-текстов](#);
- [Пример результата генерации](#);

4). Предусмотреть возможность задать свой кастомный текст для конкретной пары (на конкретном языке) (а также возможность его удаления). В случае, если для пары и данного языка задан кастомный текст - система должна возвращать кастомный текст.

5) Ограничения по частоте обращений:

- Система должна позволять отправлять запросы на генерацию в многопоточном режиме до 10 запросов одновременно.
- Система в течении 60 секунд должна вернуть результат. В качестве результата будет успешно выполненное задание, либо сообщение об ошибке.
- В случае успешного ответа, тексты сохраняются на сайт и обращаемся к API для генерации контента для следующей пары или языковой версии.
- В случае получения сообщения об ошибке повторяем запрос на генерацию контента.

6) Объем данных. Нужно понимать что будет 100к текстов на 40 языках. Т.е 4 миллиона текстов объемом 15к символов

7) Реализовать интерфейс где можно будет массово или одиночно для заданного текста выгрузить сгенерированные тексты в текстовом формате