# смотри. мы будем это публиковать на гитхаб. нужна авторизация/синхронизация с гугл таблицей через екшнс - опубликовал в мейн - получил обновления таблицы!

И все сообщения от бота! должны быть в отдельном конфиг листе на ексель таблице!! чтобы админ управлял легко! и мог изменить текст автоматически. скрипт должен каждый раз брать настройки с этого листа! добавь дополнение к этому всем. доп файл

создай доп файл мд

Добавлю отдельный файл-надстройку к ТЗ, чтобы закрыть два новых требования: GitHub Actions ↔ Google Sheets и вынос всех текстов бота в конфиг-таблицу.

Ниже — содержимое нового файла `TZ-GitHub-Sync-and-Config.md`, который можно положить рядом с основным ТЗ в репозитории.

```
# ДОПОЛНЕНИЕ К ТЗ: GITHUB ACTIONS И КОНФИГ ТЕКСТОВ БОТА ЧЕРЕЗ GOOGLE SHEETS

**Проект:** Бот автоматизации аукционов ВКонтакте
**Файл:** Дополнение к ТЗ (GitHub + Config-лист сообщений)
**Дата:** 06.02.2026

---

## 1. ЦЕЛЬ ДОПОЛНЕНИЯ

1. Добавить механизм **синхронизации конфигурации бота** (включая тексты сообщений) через
   - При публикации в `main` GitHub-репозитория — автоматически обновлять конфиг-лист в G
   - Обратно: админ может править тексты в Google Sheets, а скрипт всегда читает актуальн

2. Вынести **все текстовые сообщения бота** (ЛС, комментарии, шаблоны) в отдельный лист `
   - Админ управляет текстами без правок кода.
   - Скрипт **никогда не хардкодит текст**, а берёт его только из `BotConfig`.

---

## 2. АРХИТЕКТУРА И ПОДХОД

### 2.1. Общая схема

```text
GitHub (репозиторий с кодом)
  |
```
```

```
   ├─ push в ветку main → GitHub Actions workflow
   │       │
   │       └─ Обновление конфиг-данных в Google Sheets (через Sheets API)
   │
Google Sheets
   ├─ Лист Settings (общие параметры: шаг ставки, время и т.д.)
   ├─ Лист Config/Bids/Winners (данные аукциона)
   └─ Лист BotConfig (все текстовые сообщения)
           ▲
           └─ Бот при каждом вызове читает нужный ключ → подставляет значения
```

### 2.2. Два источника правды

- **Тексты сообщений и шаблоны**:
  Источник правды → лист `BotConfig` в Google Sheets.
- **Структура и дефолтные значения** (ключи, примеры текстов, комментарии):
  Источник правды → YAML/JSON-файл в репозитории (`config/bot-messages.yml`).

GitHub Actions при пуше в `main` синхронизирует структуру и дефолты из репозитория в лист `BotConfig`, **не затирая пользовательские правки текста**, если они уже есть.

## 3. ЛИСТ `BotConfig` В GOOGLE SHEETS

### 3.1. Структура листа `BotConfig`

Создать новый лист в основной таблице: `BotConfig`.

Структура:

| key | value | desc |
|---|---|---|
| msg_outbid_title | □ Ваша ставка перебита! | Заголов |
| msg_outbid_body | Лот: {LOT_NAME}\nВаша ставка: {OLD_BID}₽ ... | Текст |
| msg_winner_title | □ Поздравляем! Вы выиграли лот. | Заголов |
| msg_winner_body | Фигурка: {LOT_NAME} ... | Тело л |
| msg_auction_no_wins | Вы не выиграли ни одного лота в этом аукционе. | Ответ |
| msg_auction_summary_template | Добрый день!\n\nВаши выигранные лоты:\n... | Шабло |
| msg_accumulate_ok | ✅ Отлично! Ваши фигурки отложены... | Ответ |
| msg_error_generic | Произошла ошибка. Попробуйте позже. | Общая |
| comment_winner_template | □ Победитель: @id{USER_ID} ({USER_NAME}) ... | Коммен |
| report_admin_header | □ Аукцион завершён!\n\n | Заголо |
| report_admin_line | Лот {LOT_ID}: {LOT_NAME} — {PRICE}₽ ({USER}) | Шаблон |

- `key` — уникальный идентификатор сообщения.
- `value` — текущий текст сообщения (может менять админ).
- `description` — пояснение назначение ключа (для админа).
- `enabled` — флаг (TRUE/FALSE) для включения/выключения отдельных сообщений (на будущее).

### 3.2. Плейсхолдеры

Поддерживаем плейсхолдеры в `value`, которые будут заменяться в коде:

- `{LOT_ID}` — номер лота
- `{LOT_NAME}` — название лота
- `{OLD_BID}` — предыдущая ставка
- `{NEW_BID}` — новая ставка
- `{USER_NAME}` — имя пользователя
- `{USER_ID}` — VK id пользователя
- `{POST_LINK}` — ссылка на пост
- `{LOTS_LIST}` — список выигранных лотов
- `{LOTS_TOTAL}` — сумма за лоты
- `{DELIVERY_COST}` — стоимость доставки
- `{TOTAL_COST}` — итого к оплате
- `{ITEM_COUNT}` — количество лотов
- `{PAYMENT_PHONE}` — номер для оплаты
- `{PAYMENT_BANK}` — банк

## 4. ДОПОЛНЕНИЯ В КОД БОТА (APPS SCRIPT)

### 4.1. Класс для работы с конфигом сообщений

Добавить вспомогательный модуль, например `ConfigService.gs`:

```
const CONFIG_SHEET_NAME = 'BotConfig';
const CONFIG_CACHE_TTL = 300; // 5 минут

function getConfigSheet_() {
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  return ss.getSheetByName(CONFIG_SHEET_NAME);
}

/**
 * Загружает все ключи/значения из BotConfig в объект.
 */
function loadBotConfig() {
  const cache = CacheService.getScriptCache();
  const cached = cache.get('bot_config');
  if (cached) {
    return JSON.parse(cached);
  }

  const sheet = getConfigSheet_();
  if (!sheet) {
```

```
    throw new Error('Sheet "BotConfig" not found');
  }

  const values = sheet.getDataRange().getValues();
  const headers = values;
  const keyIndex = headers.indexOf('key');
  const valueIndex = headers.indexOf('value');
  const enabledIndex = headers.indexOf('enabled');

  const config = {};

  for (let i = 1; i < values.length; i++) {
    const row = values[i];
    const key = row[keyIndex];
    const value = row[valueIndex];
    const enabled = (row[enabledIndex] + '').toLowerCase() !== 'false';

    if (key && enabled) {
      config[key] = value;
    }
  }

  cache.put('bot_config', JSON.stringify(config), CONFIG_CACHE_TTL);
  return config;
}

/**
 * Возвращает строку-конфиг по ключу. Если нет — дефолт из кода или бросает ошибку.
 */
function getConfigValue(key, defaultValue) {
  const config = loadBotConfig();
  if (config[key] !== undefined && config[key] !== '') {
    return config[key];
  }
  if (defaultValue !== undefined) {
    return defaultValue;
  }
  throw new Error('Config key not found: ' + key);
}

/**
 * Подстановка плейсхолдеров в шаблон.
 */
function fillTemplate(template, data) {
  let result = template;
  Object.keys(data).forEach(function (placeholder) {
    const re = new RegExp('\\{' + placeholder + '\\}', 'g');
    result = result.replace(re, data[placeholder]);
  });
  return result;
}
```

### 4.2. Использование в основных модулях

**Пример: уведомление о перебитой ставке**

```
function sendOutbidNotification(previousLeaderId, lot, oldBid, newBid, newLeaderName, pos
  const title = getConfigValue('msg_outbid_title', '⚠ Ваша ставка перебита!');
  const bodyTemplate = getConfigValue('msg_outbid_body',
    'Лот: {LOT_NAME}\nВаша ставка: {OLD_BID}₽\nНовая ставка: {NEW_BID}₽\n\nСделайте новук
  );

  const body = fillTemplate(bodyTemplate, {
    LOT_NAME: lot.name,
    OLD_BID: oldBid,
    NEW_BID: newBid,
    POST_LINK: postLink,
  });

  const text = title + '\n\n' + body;
  sendMessage(previousLeaderId, text);
}
```

**Пример: сводка по команде** `АУКЦИОН`

```
function sendAuctionSummary(userId, lotsList, lotsTotal, deliveryCost, totalCost, itemCou
  const template = getConfigValue('msg_auction_summary_template');
  const message = fillTemplate(template, {
    LOTS_LIST: lotsList,
    LOTS_TOTAL: lotsTotal,
    DELIVERY_COST: deliveryCost,
    TOTAL_COST: totalCost,
    ITEM_COUNT: itemCount,
    PAYMENT_PHONE: settings.payment_phone,
    PAYMENT_BANK: settings.payment_bank,
  });

  sendMessage(userId, message);
}
```

**Пример: комментарий с победителем**

```
function buildWinnerComment(lot, winner) {
  const template = getConfigValue(
    'comment_winner_template',
    '🏆 Победитель: @id{USER_ID} ({USER_NAME})\nФинальная ставка: {PRICE}₽\n\nДля оформлен
  );

  return fillTemplate(template, {
    USER_ID: winner.id,
    USER_NAME: winner.first_name + ' ' + winner.last_name,
    PRICE: lot.current_price,
  });
}
```

**КРИТИЧЕСКОЕ ТРЕБОВАНИЕ:**
Все новые/старые тексты бота (комментарии, ЛС, ошибки, подсказки) должны быть реализованы **только через** `getConfigValue` **+ плейсхолдеры**, прямых строк в коде быть не должно, кроме дефолтов.

## 5. СИНХРОНИЗАЦИЯ ЧЕРЕЗ GITHUB ACTIONS

### 5.1. Цель

При каждом пуше в `main`:

- Обновлять структуру и набор ключей в листе `BotConfig` на основе `config/bot-messages.yml` (или `.json`):
  - Если ключ **есть в файле, но нет в таблице** — добавить строку с дефолтным текстом и description.
  - Если ключ **есть в таблице, но нет в файле** — **не удалять**, помечать как `enabled = FALSE` или оставлять (на усмотрение).
  - Если ключ есть и там, и там — **НЕ перезаписывать** `value`, чтобы не терять правки админа; обновлять лишь `description` при необходимости.

Таким образом, GitHub → управляет структурой и дефолтами, а админ в Google Sheets → управляет конкретными текстами.

### 5.2. Файл конфигурации в репозитории

Создать файл: `config/bot-messages.yml`

```
messages:
  msg_outbid_title:
    default: "⚠ Ваша ставка перебита!"
    description: "Заголовок уведомления о перебитой ставке"
  msg_outbid_body:
    default: |
      Лот: {LOT_NAME}
      Ваша ставка: {OLD_BID}₽
      Новая ставка: {NEW_BID}₽

      Сделайте новую ставку здесь: {POST_LINK}
    description: "Текст уведомления о перебитой ставке"
  msg_winner_title:
    default: "🎉 Поздравляем! Вы выиграли лот."
    description: "Заголовок ЛС победителю"
  msg_winner_body:
    default: |
      Фигурка: {LOT_NAME}
      Ваша ставка: {PRICE}₽

      Для оформления заказа напишите "АУКЦИОН" в ЛС группы.
    description: "Текст ЛС победителю"
  msg_auction_summary_template:
```

```
    default: |
      Добрый день!

      Ваши выигранные лоты:
      {LOTS_LIST}

      Сумма за лоты: {LOTS_TOTAL}₽
      Доставка ({ITEM_COUNT} фигурок): {DELIVERY_COST}₽
      ━━━━━━━━━━━━━━━━━━━━
      ИТОГО К ОПЛАТЕ: {TOTAL_COST}₽

      Для оформления отправки пришлите:
      1. ФИО полностью
      2. Город и адрес
      3. Номер телефона
      4. Скриншот оплаты

      Реквизиты: {PAYMENT_BANK} — {PAYMENT_PHONE}
    description: "Шаблон сообщения по команде АУКЦИОН"
```

### 5.3. GitHub Actions: общая идея

1. Создать сервисный аккаунт в Google Cloud с доступом к Google Sheets API.
2. Сохранить JSON ключ как GitHub Secret: `GCP_SERVICE_ACCOUNT_JSON`.
3. Сохранить ID таблицы в Secret: `GOOGLE_SHEET_ID`.

Пример workflow-файла: `.github/workflows/sync-bot-config.yml`

```
name: Sync BotConfig to Google Sheets

on:
  push:
    branches: [ main ]
    paths:
      - "config/bot-messages.yml"

jobs:
  sync-config:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repo
        uses: actions/checkout@v4

      - name: Setup Node.js
        uses: actions/setup-node@v4
        with:
          node-version: "20"

      - name: Install dependencies
        run: npm install js-yaml googleapis

      - name: Sync BotConfig to Google Sheets
        env:
```

```
          GCP_SERVICE_ACCOUNT_JSON: ${{ secrets.GCP_SERVICE_ACCOUNT_JSON }}
          GOOGLE_SHEET_ID: ${{ secrets.GOOGLE_SHEET_ID }}
        run: |
          node ./scripts/sync-bot-config.js
```

### 5.4. Скрипт `scripts/sync-bot-config.js` (концепция)

Псевдокод:

```
const { google } = require('googleapis');
const fs = require('fs');
const yaml = require('js-yaml');

async function main() {
  const cred = JSON.parse(process.env.GCP_SERVICE_ACCOUNT_JSON);
  const sheetId = process.env.GOOGLE_SHEET_ID;
  const auth = new google.auth.JWT(
    cred.client_email,
    null,
    cred.private_key,
    ['https://www.googleapis.com/auth/spreadsheets']
  );
  const sheets = google.sheets({ version: 'v4', auth });

  // 1. Прочитать bot-messages.yml
  const doc = yaml.load(fs.readFileSync('config/bot-messages.yml', 'utf8'));
  const messages = doc.messages || {};

  // 2. Прочитать текущий BotConfig из таблицы
  const res = await sheets.spreadsheets.values.get({
    spreadsheetId: sheetId,
    range: 'BotConfig!A1:D1000',
  });

  const rows = res.data.values || [];
  const header = rows || ['key', 'value', 'description', 'enabled'];
  const keyIndex = header.indexOf('key');
  const valueIndex = header.indexOf('value');
  const descIndex = header.indexOf('description');
  const enabledIndex = header.indexOf('enabled');

  const existing = {};
  for (let i = 1; i < rows.length; i++) {
    const row = rows[i];
    const key = row[keyIndex];
    existing[key] = { rowIndex: i + 1, row };
  }

  const updates = [];
  const appends = [];

  // 3. Для каждого ключа из YAML
  for (const [key, meta] of Object.entries(messages)) {
    const def = meta.default || '';
    const desc = meta.description || '';
```

```
    if (existing[key]) {
      // Уже есть строка: не трогаем value, но обновляем description/ enabled
      const rowIndex = existing[key].rowIndex;
      const row = existing[key].row;

      const newDesc = desc || row[descIndex];
      const newEnabled = row[enabledIndex] || 'TRUE';

      updates.push({
        range: `BotConfig!C${rowIndex}:D${rowIndex}`,
        values: [[newDesc, newEnabled]],
      });
    } else {
      // Нет ключа → добавляем с дефолтным value
      appends.push([
        key,
        def,
        desc,
        'TRUE',
      ]);
    }
  }

  // 4. Обновить description/enabled
  if (updates.length > 0) {
    await sheets.spreadsheets.values.batchUpdate({
      spreadsheetId: sheetId,
      requestBody: {
        valueInputOption: 'RAW',
        data: updates,
      },
    });
  }

  // 5. Добавить новые строки
  if (appends.length > 0) {
    await sheets.spreadsheets.values.append({
      spreadsheetId: sheetId,
      range: 'BotConfig!A1',
      valueInputOption: 'RAW',
      insertDataOption: 'INSERT_ROWS',
      requestBody: {
        values: appends,
      },
    });
  }
}

main().catch(err => {
  console.error(err);
  process.exit(1);
});
```

**Итог:**

- Ты пушишь изменения в `config/bot-messages.yml` → GitHub Actions приводит `BotConfig` к нужной структуре.
- Админ меняет `value` прямо в Google Sheets → бот использует актуальный текст **без нового деплоя кода**.

## 6. ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

1. **Все тексты бота вынести в лист** `BotConfig`**.**
2. Вся логика отправки сообщений и комментариев должна использовать `getConfigValue(...)` + `fillTemplate(...)`.
3. В основном ТЗ (`TZ-VK-Auction-Bot.md`) в разделах про шаблоны сообщений добавить ссылку:
   *«Текст берётся из листа* `BotConfig` *по ключу* `{ключ}`*»* — вместо жёстко прописанных строк.
4. В репозитории создать:
   - `config/bot-messages.yml`
   - `.github/workflows/sync-bot-config.yml`
   - `scripts/sync-bot-config.js`
5. Описать для заказчика в документации (не технической), что:
   - Все формулировки можно менять прямо в таблице.
   - Перезагрузка бота не требуется: новые тексты подхватываются автоматически (с задержкой до 5 минут из-за кеша).

**Конец дополнения к ТЗ**

```
Если хочешь, могу сразу адаптировать основной `TZ-VK-Auction-Bot.md`, чтобы в разделах пр
<span style="display:none">[^1][^10][^11][^12][^13][^14][^15][^2][^3][^4][^5][^6][^7][^8]

<div align="center">⁂</div>

[^1]: https://github.com/marketplace/actions/gsheet-action
[^2]: https://github.com/marketplace/actions/transfer-github-project-issues-into-google-s
[^3]: https://discuss.google.dev/t/google-sheets-credentials-in-github-actions-environmen
[^4]: https://github.com/marketplace/actions/classroom-to-sheets-integration
[^5]: https://www.jaredwhalen.com/blog/deploy-gh-action-from-sheets
[^6]: https://apptivasoftware.com/blog/creating-a-settings-section-in-google-sheets-apps-
[^7]: https://dev.vk.com/en/messages.send
[^8]: https://www.obrien.page/blog/2023/03_10_google_and_github_actions/
[^9]: https://support.google.com/a/answer/13686736?hl=en
[^10]: https://stackoverflow.com/questions/65216016/how-to-send-post-requests-to-the-vk-a
[^11]: https://apipheny.io/github-api-google-sheets/
[^12]: https://stackoverflow.com/questions/64775506/apps-script-external-stylesheet
[^13]: https://messaggio.com/guides/requirements-for-the-content-and-format-of-vk-message
```

```
[^14]: https://github.com/marketplace/actions/sync-github-issues-to-a-sheet
[^15]: https://www.youtube.com/watch?v=liiKfPJeWGo
```