

# GetCourse: ????? ?????????? ? Google ????????

Скрипт для Google Sheets, который собирает статистику рассылок, количество пользователей в сегментах и статистику заказов из GetCourse.

## ??????????

- **Письма** — выгрузка статистики рассылок (всего, доставлено, просмотры, клики, отписки, ошибки, запрещено)
- **Сегменты пользователей** — подсчёт пользователей в сегменте по ссылке
- **Сегменты заказов** — статистика заказов: общее количество, платных, сумма платных, оплаченных, сумма оплаченных
- **Мульти-аккаунт** — поддержка нескольких аккаунтов GetCourse

## ??????????

- Язык интерфейса пользователя GetCourse должен быть **русским**
- У пользователя должен быть доступ ко всем разделам, из которых собираются данные (рассылки, пользователи, заказы)

## ??????????

1. Открыть Google Таблицу
2. **Расширения** → **Apps Script**
3. Удалить содержимое файла `Code.gs`
4. Вставить код из раздела [Код скрипта](#)
5. Сохранить (Ctrl+S)
6. Вернуться в таблицу, обновить страницу
7. В меню появится вкладка **GetCourse**

## ??????? ????????

1. **GetCourse** → **Управление аккаунтами** → добавить аккаунт (имя, email, пароль, домен вида `https://your-school.getcourse.ru`)
2. **GetCourse** → **Проверить подключение** — убедиться, что авторизация работает
3. При первом запуске Google попросит разрешения — нажать "Разрешить"

????????????

???????????? ???? ?????

1. В ячейки вписать ID рассылок (число) или ссылки на рассылки
2. Выделить эти ячейки
3. **GetCourse** → **Письма: обновить выбранные**
4. В ячейках справа от ID появятся данные (название, всего, доставлено и т.д.), каждая ячейка с примечанием

???????????? ???? ????????? ? ???? ?????

1. В ячейку вставить ссылку на сегмент (или гиперссылку)
2. Выделить ячейку
3. **GetCourse** → **Пользователи: кол-во в сегментах**
4. В ячейке справа — количество, через одну — дата обновления

???????????? ???? ????? ? ???? ?????

1. В ячейку вставить ссылку на сегмент заказов (URL, гиперссылка или формула HYPERLINK с динамическими параметрами)
2. Выделить ячейку
3. **GetCourse** → **Заказы: статистика сегментов**
4. В ячейках справа появятся: кол-во заказов, платных, сумма платных, оплаченных, сумма оплаченных, дата обновления

Поддерживаются ссылки с `segment_id` (сохранённые сегменты) и с `rule_string` (произвольные выборки).

??? ???????

```
/**
 * GC: Email Stats
```

```

*
* Сбор статистики рассылок GetCourse прямо в Google Таблице.
*
* Установка:
* 1. Расширения → Apps Script → вставить этот код
* 2. Вписать ID или ссылки рассылок в ячейки
* 3. Выделить ячейки → меню GC: Email Stats → Обновить выбранные
*/

// =====
// Меню
// =====

function onOpen() {
  var menuName = 'GetCourse';
  var accounts = getAccounts();
  if (accounts.length > 0) {
    var idx = getActiveAccountIndex();
    var acc = accounts[idx];
    menuName = 'GetCourse [' + (acc.name || acc.email) + ']';
  }

  SpreadsheetApp.getUi()
    .createMenu(menuName)
    .addItem('Письма: обновить выбранные', 'updateSelectedRows')
    .addItem('Пользователи: кол-во в сегментах', 'updateSegmentCounts')
    .addItem('Заказы: статистика сегментов', 'updateDealSegmentStats')
    .addSeparator()
    .addItem('Управление аккаунтами', 'showAccountManager')
    .addItem('Проверить подключение', 'testConnection')
    .addToUi();
}

// =====
// Управление аккаунтами
// =====

function getAccounts() {
  var props = PropertiesService.getScriptProperties();

```

```

var json = props.getProperty('gc_accounts');
if (!json) return [];
try { return JSON.parse(json); } catch (e) { return []; }
}

function getActiveAccountIndex() {
    var props = PropertiesService.getScriptProperties();
    var idx = parseInt(props.getProperty('gc_active_account') || '0', 10);
    var accounts = getAccounts();
    if (idx >= accounts.length) idx = 0;
    return idx;
}

function saveAccount(accountData) {
    var accounts = getAccounts();
    var props = PropertiesService.getScriptProperties();

    if (accountData.index !== undefined && accountData.index !== null && accountData.index
    >= 0 && accountData.index < accounts.length) {
        accounts[accountData.index] = {
            name: accountData.name,
            email: accountData.email,
            password: accountData.password,
            domain: accountData.domain
        };
    } else {
        accounts.push({
            name: accountData.name,
            email: accountData.email,
            password: accountData.password,
            domain: accountData.domain
        });
    }

    props.setProperty('gc_accounts', JSON.stringify(accounts));

    if (accounts.length === 1) {
        props.setProperty('gc_active_account', '0');
    }
}

```

```

    return accounts.length - 1;
}

function deleteAccount(index) {
    var accounts = getAccounts();
    var props = PropertiesService.getScriptProperties();

    if (index < 0 || index >= accounts.length) return;
    accounts.splice(index, 1);
    props.setProperty('gc_accounts', JSON.stringify(accounts));

    var activeIdx = getActiveAccountIndex();
    if (index === activeIdx) {
        props.setProperty('gc_active_account', '0');
        CacheService.getScriptCache().remove('gc_session');
    } else if (index < activeIdx) {
        props.setProperty('gc_active_account', String(activeIdx - 1));
    }
}

function setActiveAccount(index) {
    var accounts = getAccounts();
    if (index < 0 || index >= accounts.length) return;
    PropertiesService.getScriptProperties().setProperty('gc_active_account', String(index));
    CacheService.getScriptCache().remove('gc_session');
}

function getAccountsForDialog() {
    var accounts = getAccounts();
    var activeIdx = getActiveAccountIndex();
    return { accounts: accounts, activeIndex: activeIdx };
}

function getConfig() {
    var accounts = getAccounts();
    if (accounts.length === 0) {
        throw new Error('NO_ACCOUNTS: Нет сохранённых аккаунтов. Добавьте аккаунт через меню
GetCourse → Управление аккаунтами.');
```

```

    }
    var idx = getActiveAccountIndex();
    var acc = accounts[idx];
    return {
        email: acc.email,
        password: acc.password,
        domain: acc.domain
    };
}

// =====
// Диалог управления аккаунтами
// =====

function showAccountManager() {
    var html = HtmlService.createHtmlOutput(getAccountManagerHtml_())
        .setWidth(520)
        .setHeight(480)
        .setTitle('Управление аккаунтами');
    SpreadsheetApp.getUi().showModalDialog(html, 'Управление аккаунтами GetCourse');
}

function getAccountManagerHtml_() {
    return '\
<!DOCTYPE html>\
<html>\
<head>\
<style>\
    * { box-sizing: border-box; font-family: "Google Sans", Roboto, Arial, sans-serif; }\
    body { margin: 0; padding: 16px; font-size: 14px; color: #202124; }\
    h3 { margin: 0 0 12px; font-size: 16px; }\
    .account-list { margin-bottom: 16px; }\
    .account-item {\
        display: flex; align-items: center; padding: 10px 12px;\
        border: 1px solid #dadce0; border-radius: 8px; margin-bottom: 8px;\
        cursor: pointer; transition: background 0.15s;\
    }\
    .account-item:hover { background: #f1f3f4; }\
    .account-item.active { border-color: #1a73e8; background: #e8f0fe; }\

```

```

.account-item .info { flex: 1; margin-left: 10px; }\
.account-item .name { font-weight: 500; }\
.account-item .email { font-size: 12px; color: #5f6368; }\
.account-item .domain { font-size: 11px; color: #80868b; }\
.account-item .actions { display: flex; gap: 4px; }\
.btn { padding: 6px 16px; border-radius: 4px; border: 1px solid #dadce0;\
  background: #fff; cursor: pointer; font-size: 13px; }\
.btn:hover { background: #f1f3f4; }\
.btn-primary { background: #1a73e8; color: #fff; border: none; }\
.btn-primary:hover { background: #1557b0; }\
.btn-danger { color: #d93025; border-color: #d93025; }\
.btn-danger:hover { background: #fce8e6; }\
.btn-sm { padding: 4px 10px; font-size: 12px; }\
.form-group { margin-bottom: 12px; }\
.form-group label { display: block; margin-bottom: 4px; font-size: 12px; color: #5f6368;\
}\
.form-group input { width: 100%; padding: 8px 10px; border: 1px solid #dadce0;\
  border-radius: 4px; font-size: 14px; }\
.form-group input:focus { outline: none; border-color: #1a73e8; }\
.form-actions { display: flex; gap: 8px; justify-content: flex-end; margin-top: 16px; }\
.radio { width: 16px; height: 16px; accent-color: #1a73e8; cursor: pointer; }\
.empty { text-align: center; padding: 24px; color: #80868b; }\
#form-section { display: none; }\
</style>\
</head>\
<body>\
\
<div id="list-section">\
  <h3>Аккаунты</h3>\
  <div id="account-list" class="account-list"><div class="empty">Загрузка...</div></div>\
  <button class="btn btn-primary" onclick="showForm(-1)">+ Добавить аккаунт</button>\
</div>\
\
<div id="form-section">\
  <h3 id="form-title">Добавить аккаунт</h3>\
  <input type="hidden" id="edit-index" value="-1">\
  <div class="form-group">\
    <label>Название (для удобства)</label>\
    <input type="text" id="f-name" placeholder="Например: Фитнес Мама">\

```

```

</div>\
<div class="form-group">\
  <label>Email</label>\
  <input type="email" id="f-email" placeholder="user@example.com">\
</div>\
<div class="form-group">\
  <label>Пароль</label>\
  <input type="password" id="f-password" placeholder="Пароль от GetCourse">\
</div>\
<div class="form-group">\
  <label>Домен школы</label>\
  <input type="url" id="f-domain" placeholder="https://myschool.getcourse.ru">\
</div>\
<div class="form-actions">\
  <button class="btn" onclick="showList()">Отмена</button>\
  <button class="btn btn-primary" onclick="saveForm()">Сохранить</button>\
</div>\
</div>\
\
<script>\
var currentData = { accounts: [], activeIndex: 0 };
\
function load() {\
  google.script.run.withSuccessHandler(function(data) {\
    currentData = data;\
    render();\
  }).getAccountsForDialog();\
}\
\
function render() {\
  var list = document.getElementById("account-list");\
  if (currentData.accounts.length === 0) {\
    list.innerHTML = "<div class=\\\"empty\\\">Нет аккаунтов. Добавьте первый аккаунт.</div>";\
    return;\
  }\
  var html = "";\
  for (var i = 0; i < currentData.accounts.length; i++) {\
    var a = currentData.accounts[i];\

```



```

    var isActive = (i === currentData.activeIndex);\
    html += "<div class=\\\"account-item\" + (isActive ? \" active\" : \"\") + \"\\\">";\
    html += "<input type=\\\"radio\\\" class=\\\"radio\\\" name=\\\"active\\\" \" + (isActive ?
\"checked\" : \"\") + \" onclick=\\\"activate(\" + i + \" )\\\">";\
    html += "<div class=\\\"info\\\">";\
    html += "<div class=\\\"name\\\">\" + esc(a.name || \"Без названия\") + (isActive ? \" ✓\" :
\"\") + "</div>";\
    html += "<div class=\\\"email\\\">\" + esc(a.email) + "</div>";\
    html += "<div class=\\\"domain\\\">\" + esc(a.domain) + "</div>";\
    html += "</div>";\
    html += "<div class=\\\"actions\\\">";\
    html += "<button class=\\\"btn btn-sm\\\" onclick=\\\"event.stopPropagation();showForm(\"
+ i + \" )\\\">Изм.</button>";\
    html += "<button class=\\\"btn btn-sm btn-danger\\\"
onclick=\\\"event.stopPropagation();del(\" + i + \" )\\\">X</button>";\
    html += "</div></div>";\
  }\
  list.innerHTML = html;\
}\
\
function esc(s) { var d = document.createElement("div"); d.textContent = s; return
d.innerHTML; }\
\
function activate(idx) {\
  google.script.run.withSuccessHandler(function() {\
    currentData.activeIndex = idx;\
    render();\
  }).setActiveAccount(idx);\
}\
\
function del(idx) {\
  if (!confirm("Удалить аккаунт \\\"\" + (currentData.accounts[idx].name ||
currentData.accounts[idx].email) + "\\\"?")) return;\
  google.script.run.withSuccessHandler(function() { load(); }).deleteAccount(idx);\
}\
\
function showForm(idx) {\
  document.getElementById("list-section").style.display = "none";\
  document.getElementById("form-section").style.display = "block";\
}

```

```

document.getElementById("edit-index").value = idx;\
if (idx >= 0 && idx < currentData.accounts.length) {\
    var a = currentData.accounts[idx];\
    document.getElementById("form-title").textContent = "Редактировать аккаунт";\
    document.getElementById("f-name").value = a.name || "";\
    document.getElementById("f-email").value = a.email || "";\
    document.getElementById("f-password").value = a.password || "";\
    document.getElementById("f-domain").value = a.domain || "";\
} else {\
    document.getElementById("form-title").textContent = "Добавить аккаунт";\
    document.getElementById("f-name").value = "";\
    document.getElementById("f-email").value = "";\
    document.getElementById("f-password").value = "";\
    document.getElementById("f-domain").value = "";\
}\
}\
\
function showList() {\
    document.getElementById("form-section").style.display = "none";\
    document.getElementById("list-section").style.display = "block";\
}\
\
function saveForm() {\
    var idx = parseInt(document.getElementById("edit-index").value, 10);\
    var data = {\
        index: idx >= 0 ? idx : null,\
        name: document.getElementById("f-name").value.trim(),\
        email: document.getElementById("f-email").value.trim(),\
        password: document.getElementById("f-password").value,\
        domain: document.getElementById("f-domain").value.trim().replace(/\\\/+$/, "")\
    };\
    if (!data.email || !data.password || !data.domain) {\
        alert("Заполните Email, Пароль и Домен.");\
        return;\
    }\
    google.script.run.withSuccessHandler(function() {\
        showList();\
        load();\
    }).saveAccount(data);\
}

```

```

}\
\
load();\
</script>\
</body>\
</html>';
}

// =====
// Авторизация
// =====

/**
 * Login в GetCourse по email/пароллю через AJAX API.
 * GetCourse использует React-форму и эндпоинт /user/public/user/json
 * Возвращает строку cookies для авторизованной сессии.
 */
function login() {
    var config = getConfig();
    var ua = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/120.0.0.0 Safari/537.36';

    // Шаг 1: GET любой страницы для получения начального cookie сессии
    var getResp = UrlFetchApp.fetch(config.domain + '/cms/system/login', {
        method: 'get',
        headers: { 'User-Agent': ua },
        muteHttpExceptions: true,
        followRedirects: true
    });
    var initialCookies = extractAllCookies_(getResp);

    // Шаг 2: AJAX-логин через /user/public/user/json
    var ajaxUrl = config.domain + '/user/public/user/json';
    var postResp = UrlFetchApp.fetch(ajaxUrl, {
        method: 'post',
        headers: {
            'User-Agent': ua,
            'Content-Type': 'application/x-www-form-urlencoded',
            'X-Requested-With': 'XMLHttpRequest',

```

```

    'Referer': config.domain + '/cms/system/login',
    'Cookie': initialCookies
  },
  payload: 'action=login&email=' + encodeURIComponent(config.email) +
    '&password=' + encodeURIComponent(config.password),
  muteHttpExceptions: true,
  followRedirects: false
});

var postCode = postResp.getResponseCode();
var postBody = postResp.getContentText();
var postCookies = extractAllCookies_(postResp);
var finalCookies = mergeCookies_(initialCookies, postCookies);

// Парсим JSON-ответ
var json;
try {
  json = JSON.parse(postBody);
} catch (e) {
  throw new Error('LOGIN_FAILED: Неожиданный ответ сервера (HTTP ' + postCode + '): ' +
    postBody.substring(0, 200));
}

// Проверяем результат
if (json.error) {
  throw new Error('LOGIN_FAILED: ' + (json.error.message || json.error.text ||
    JSON.stringify(json.error)));
}

// Шаг 3: Если в ответе есть redirectUrl – переходим по нему (устанавливает финальные
cookies)
if (json.data && json.data.redirectUrl) {
  var redirectUrl = json.data.redirectUrl;
  if (redirectUrl.indexOf('http') !== 0) {
    redirectUrl = config.domain + redirectUrl;
  }
  var redirResp = UrlFetchApp.fetch(redirectUrl, {
    method: 'get',
    headers: { 'User-Agent': ua, 'Cookie': finalCookies },
  });
}

```

```

        muteHttpExceptions: true,
        followRedirects: true
    });
    var redirectCookies = extractAllCookies_(redirectResp);
    finalCookies = mergeCookies_(finalCookies, redirectCookies);
}

// Шаг 4: Проверяем что авторизация работает
var testResp = UrlFetchApp.fetch(config.domain +
'/notifications/control/mailings/active', {
    method: 'get',
    headers: { 'User-Agent': ua, 'Cookie': finalCookies },
    muteHttpExceptions: true,
    followRedirects: true
});
var testCookies = extractAllCookies_(testResp);
finalCookies = mergeCookies_(finalCookies, testCookies);

if (isLoginPage_(testResp.getContentText())) {
    throw new Error('LOGIN_FAILED: Логин прошёл, но сессия не активна. Ответ JSON: ' +
postBody.substring(0, 300));
}

// Сохраняем cookies в кеш на 2 часа
CacheService.getScriptCache().put('gc_session', finalCookies, 7200);
return finalCookies;
}

/**
 * Извлекает BCE cookies из заголовков Set-Cookie ответа.
 * Возвращает строку "name1=val1; name2=val2"
 */
function extractAllCookies_(response) {
    var allHeaders = response.getAllHeaders();
    var setCookieHeaders = allHeaders['Set-Cookie'];
    if (!setCookieHeaders) return '';

    if (typeof setCookieHeaders === 'string') {
        setCookieHeaders = [setCookieHeaders];
    }
}

```

```

}

var cookies = {};
for (var i = 0; i < setCookieHeaders.length; i++) {
    var pair = setCookieHeaders[i].split(';')[0]; // берём только name=value
    var eqIdx = pair.indexOf('=');
    if (eqIdx > 0) {
        var name = pair.substring(0, eqIdx).trim();
        var value = pair.substring(eqIdx + 1).trim();
        cookies[name] = value;
    }
}

var parts = [];
for (var key in cookies) {
    parts.push(key + '=' + cookies[key]);
}
return parts.join('; ');
}

/**
 * Объединяет две строки cookies. Новые перезаписывают старые.
 */
function mergeCookies_(existing, fresh) {
    if (!fresh) return existing;
    if (!existing) return fresh;

    var cookies = {};

    // Парсим существующие
    var parts = existing.split(';');
    for (var i = 0; i < parts.length; i++) {
        var eqIdx = parts[i].indexOf('=');
        if (eqIdx > 0) {
            cookies[parts[i].substring(0, eqIdx).trim()] = parts[i].substring(eqIdx + 1).trim();
        }
    }

    // Перезаписываем новыми

```

```

    parts = fresh.split(';');
    for (var j = 0; j < parts.length; j++) {
        var eqIdx2 = parts[j].indexOf('=');
        if (eqIdx2 > 0) {
            cookies[parts[j].substring(0, eqIdx2).trim()] = parts[j].substring(eqIdx2 +
1).trim();
        }
    }

    var result = [];
    for (var key in cookies) {
        result.push(key + '=' + cookies[key]);
    }
    return result.join('; ');
}

/**
 * Возвращает актуальный cookie сессии.
 * Сначала проверяет кеш, если нет — логинится заново.
 */
function getSessionCookie() {
    var cached = CacheService.getScriptCache().get('gc_session');
    if (cached) return cached;
    return login();
}

// =====
// HTTP-запрос к GetCourse
// =====

function fetchMailingPage(mailingId) {
    var config = getConfig();
    var url = config.domain + '/notifications/control/mailings/update/id/' + mailingId +
'/part/main';
    var sessionCookie = getSessionCookie();

    var html = fetchWithSession_(url, sessionCookie);

    // Проверяем, не попали ли на страницу логина

```

```

if (isLoginPage_(html)) {
    // Сессия истекла – перелогиниваемся
    CacheService.getScriptCache().remove('gc_session');
    sessionCookie = login();
    html = fetchWithSession_(url, sessionCookie);

    if (isLoginPage_(html)) {
        throw new Error('LOGIN_FAILED: Не удалось авторизоваться');
    }
}

return html;
}

function fetchWithSession_(url, sessionCookie) {
    var response = UrlFetchApp.fetch(url, {
        method: 'get',
        headers: {
            'Cookie': sessionCookie,
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/120.0.0.0 Safari/537.36',
            'Accept':
'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8',
            'Accept-Language': 'ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7'
        },
        muteHttpExceptions: true,
        followRedirects: true
    });

    var code = response.getResponseCode();
    if (code !== 200) {
        throw new Error('HTTP_ERROR_' + code);
    }

    return response.getContentText();
}

function isLoginPage_(html) {
    return html.indexOf('action="/cms/system/login"') !== -1 ||

```



```

        html.indexOf('id="loginForm"') !== -1 ||
        html.indexOf('LoginForm[email]') !== -1 ||
        html.indexOf('LoginForm%5Bemail%5D') !== -1 ||
        html.indexOf('Pagina de logare') !== -1 ||
        html.indexOf('page-cms_system-login') !== -1 ||
        (html.indexOf('login') !== -1 && html.indexOf('Статистика рассылки') === -1 &&
html.indexOf('gc-user-logged') === -1);
    }

// =====
// Парсинг HTML
// =====

function parseMailingStats(html) {
    var stats = {
        title: '',
        total: 0,
        delivered: 0,
        views: 0,
        clicks: 0,
        unsubscribes: 0,
        errors: 0,
        restricted: 0
    };

    // Название рассылки из <title>
    var titleMatch = html.match(/<title>([^<]+)</title>/);
    if (titleMatch) {
        stats.title = titleMatch[1].trim();
    }

    // --- Панель 1: "Статистика рассылки" (values-table) ---

    // Всего сообщений
    var totalMatch = html.match(/Всего
сообщений</td>\s*<td[^>]*>\s*<a[^>]*>\s*([\d\s]+)\s*</a>/);
    if (totalMatch) {
        stats.total = parseInt(totalMatch[1].replace(/\s/g, ''), 10) || 0;
    }

```

```

// доставлено
var deliveredMatch = html.match(/<td
class="subkey">\s*доставлено\s*</td>\s*<td[^>]*>\s*<a[^>]*>\s*([\d\s]+)\s*</a>/);
if (deliveredMatch) {
    stats.delivered = parseInt(deliveredMatch[1].replace(/\s/g, ''), 10) || 0;
}

// ошибка
var errorsMatch = html.match(/<td
class="subkey">\s*ошибка\s*</td>\s*<td[^>]*>\s*<a[^>]*>\s*([\d\s]+)\s*</a>/);
if (errorsMatch) {
    stats.errors = parseInt(errorsMatch[1].replace(/\s/g, ''), 10) || 0;
}

// запрещено
var restrictedMatch = html.match(/<td
class="subkey">\s*запрещено\s*</td>\s*<td[^>]*>\s*<a[^>]*>\s*([\d\s]+)\s*</a>/);
if (restrictedMatch) {
    stats.restricted = parseInt(restrictedMatch[1].replace(/\s/g, ''), 10) || 0;
}

// --- Панель 2: "Статистика пользователей" ---

// просмотров (открытия)
var viewsMatch = html.match(/([\d\s]+)\s*просмотр[а-я]*\s*\(/);
if (viewsMatch) {
    stats.views = parseInt(viewsMatch[1].replace(/\s/g, ''), 10) || 0;
}

// кликов
var clicksMatch = html.match(/([\d\s]+)\s*клик[а-я]*\s*\(/);
if (clicksMatch) {
    stats.clicks = parseInt(clicksMatch[1].replace(/\s/g, ''), 10) || 0;
}

// отписки
var unsubMatch = html.match(/([\d\s]+)\s*отписка[а-яё]*\s*\(/);
if (unsubMatch) {

```

```

        stats.unsubscribes = parseInt(unsubMatch[1].replace(/\s/g, ''), 10) || 0;
    }

    return stats;
}

// =====
// Извлечение ID из ячейки
// =====

function extractMailingId_(cellValue) {
    if (!cellValue) return null;

    var str = String(cellValue).split(/\s*\|\s*/)[0].trim();

    // Если чистое число
    if (/^\d+$/.test(str)) return str;

    // URL с /id/ЧИСЛО
    var urlMatch = str.match(/\/id\/(\d+)/);
    if (urlMatch) return urlMatch[1];

    // Первое число в строке
    var numMatch = str.match(/(\d{5,})/);
    if (numMatch) return numMatch[1];

    return null;
}

// =====
// Обновление статистики
// =====

function updateSelectedRows() {
    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var sheet = ss.getActiveSheet();
    var selection = sheet.getActiveRange();

    if (!selection) {

```

```
    SpreadsheetApp.getUi().alert('Выделите ячейки с ID рассылок.');
```

```
    return;
```

```
  }
```

  

```
  var updated = 0;
```

```
  var numRows = selection.getNumRows();
```

```
  var numCols = selection.getNumColumns();
```

```
  var startRow = selection.getRow();
```

```
  var startCol = selection.getColumn();
```

  

```
  for (var r = 0; r < numRows; r++) {
```

```
    for (var c = 0; c < numCols; c++) {
```

```
      var cell = sheet.getRange(startRow + r, startCol + c);
```

```
      var cellValue = cell.getValue();
```

```
      if (!cellValue) continue;
```

  

```
      var mailingId = extractMailingId_(cellValue);
```

```
      if (!mailingId) continue;
```

  

```
      ss.toast('Обработка ID ' + mailingId + '...', 'GetCourse', 3);
```

  

```
      var row = startRow + r;
```

```
      var col = startCol + c;
```

  

```
      try {
```

```
        var html = fetchMailingPage(mailingId);
```

```
        var stats = parseMailingStats(html);
```

```
        var now = Utilities.formatDate(new Date(), Session.getScriptTimeZone(),
```

```
'dd.MM.yyyy HH:mm');
```

  

```
        var fields = [
```

```
          [stats.title, 'Название'],
```

```
          [stats.total, 'Всего'],
```

```
          [stats.delivered, 'Доставлено'],
```

```
          [stats.views, 'Просмотры'],
```

```
          [stats.clicks, 'Клики'],
```

```
          [stats.unsubscribes, 'Отписки'],
```

```
          [stats.errors, 'Ошибки'],
```

```
          [stats.restricted, 'Запрещено'],
```

```

        [now, 'Обновлено']
    ];

    for (var i = 0; i < fields.length; i++) {
        sheet.getRange(row, col + 1 + i).setValue(fields[i][0]).setNote(fields[i][1]);
    }
    SpreadsheetApp.flush();
    updated++;
} catch (e) {
    if (e.message.indexOf('LOGIN_FAILED') !== -1) {
        SpreadsheetApp.getUi().alert('Ошибка авторизации: ' + e.message);
        return;
    }
    sheet.getRange(row, col + 1).setValue('Ошибка: ' + e.message).setNote('Ошибка');
    SpreadsheetApp.flush();
}

Utilities.sleep(500);
}
}

ss.toast('Обновлено: ' + updated, 'GC: Email Stats', 5);
}

// =====
// Сегменты: количество пользователей
// =====

function extractSegmentUrl_(cell) {
    // 1. Rich-text ссылка – работает для HYPERLINK формул (включая с &/TEXT),
    //    и для ссылок вставленных через Ctrl+K
    var richText = cell.getRichTextValue();
    if (richText) {
        var url = richText.getLinkUrl();
        if (url) return url;

        var runs = richText.getRuns();
        for (var i = 0; i < runs.length; i++) {
            var runUrl = runs[i].getLinkUrl();

```

```

        if (runUrl) return runUrl;
    }
}

// 2. HYPERLINK-формула с простой строкой: =HYPERLINK("url"; "text")
var formula = cell.getFormula();
if (formula) {
    var m = formula.match(/HYPERLINK\s*\(\s*"([^"]+)"/i);
    if (m) return m[1];
}

// 3. Обычный текст URL
var val = String(cell.getValue()).trim();
if (val.match(/^https?:\/\//)) return val;

return null;
}

function fetchSegmentPage(url) {
    var sessionCookie = getSessionCookie();
    var html = fetchWithSession_(url, sessionCookie);

    if (isLoginPage_(html)) {
        CacheService.getScriptCache().remove('gc_session');
        sessionCookie = login();
        html = fetchWithSession_(url, sessionCookie);

        if (isLoginPage_(html)) {
            throw new Error('LOGIN_FAILED: Не удалось авторизоваться');
        }
    }

    return html;
}

function parseSegmentCount(html) {
    // <div class="summary">Показаны <b>1-30</b> из <b>1 941</b> записи.</div>
    var m = html.match(/<div class="summary">Показаны\s*<b>[^<]+</b>\s*из\s*<b>([\d\s]+)</b>/);

```

```
    if (m) return parseInt(m[1].replace(/\s/g, ''), 10);
    return null;
}

function updateSegmentCounts() {
    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var sheet = ss.getActiveSheet();
    var selection = sheet.getActiveRange();

    if (!selection) {
        SpreadsheetApp.getUi().alert('Выделите ячейки со ссылками на сегменты.');
```

return;

```
    }

    var updated = 0;
    var numRows = selection.getNumRows();
    var numCols = selection.getNumColumns();
    var startRow = selection.getRow();
    var startCol = selection.getColumn();

    for (var r = 0; r < numRows; r++) {
        for (var c = 0; c < numCols; c++) {
            var cell = sheet.getRange(startRow + r, startCol + c);
            if (!cell.getValue()) continue;

            var url = extractSegmentUrl_(cell);
            if (!url) continue;

            ss.toast('Обработка сегмента...', 'GC: Сегменты', 3);

            try {
                var html = fetchSegmentPage(url);
                var count = parseSegmentCount(html);

                if (count === null) {
                    ss.toast('Не удалось найти количество пользователей', 'GC: Сегменты', 3);
                    continue;
                }
            }
        }
    }
}
```

```

        var now = Utilities.formatDate(new Date(), Session.getScriptTimeZone(),
'dd.MM.yyyy HH:mm');
        var col = startCol + c;

        sheet.getRange(startRow + r, col + 1).setValue(count);
        sheet.getRange(startRow + r, col + 2).setValue(now);
        SpreadsheetApp.flush();
        updated++;
    } catch (e) {
        if (e.message.indexOf('LOGIN_FAILED') !== -1) {
            SpreadsheetApp.getUi().alert('Ошибка авторизации: ' + e.message);
            return;
        }
        sheet.getRange(startRow + r, startCol + c + 1).setValue('Ошибка: ' + e.message);
        SpreadsheetApp.flush();
    }

    Utilities.sleep(500);
}
}

ss.toast('Обновлено: ' + updated, 'GC: Сегменты', 5);
}

// =====
// Сегменты заказов (deals): статистика
// =====

/**
 * Извлекает rule_string из URL сегмента заказов.
 * Поддерживает форматы: DealContext[rule_string]=... и DealContext%5Brule_string%5D=...
 */
function extractDealRule_(url) {
    var decoded = decodeURIComponent(url);
    var m = decoded.match(/DealContext\[rule_string\]=([^\&]*)/);
    if (m && m[1]) {
        return decodeURIComponent(m[1]);
    }
    return null;
}

```



```

}

/**
 * Извлекает JSON-объект из строки начиная с позиции открывающей скобки {.
 * Считает вложенные {} для нахождения правильной закрывающей скобки.
 */
function extractJsonObject_(str, startPos) {
    if (str.charAt(startPos) !== '{') return null;
    var depth = 0;
    var inString = false;
    var escape = false;
    for (var i = startPos; i < str.length; i++) {
        var ch = str.charAt(i);
        if (escape) { escape = false; continue; }
        if (ch === '\\') { escape = true; continue; }
        if (ch === '"') { inString = !inString; continue; }
        if (inString) continue;
        if (ch === '{') depth++;
        if (ch === '}') { depth--; if (depth === 0) return str.substring(startPos, i + 1); }
    }
    return null;
}

/**
 * Для сохранённых сегментов (segment_id без rule_string):
 * загружает страницу и извлекает правило из инициализации rulePlugin({"rule": {...}}).
 */
function fetchRuleFromPage_(url, sessionCookie) {
    var html = fetchWithSession_(url, sessionCookie);

    if (isLoginPage_(html)) {
        CacheService.getScriptCache().remove('gc_session');
        sessionCookie = login();
        html = fetchWithSession_(url, sessionCookie);
        if (isLoginPage_(html)) {
            throw new Error('LOGIN_FAILED: Не удалось авторизоваться');
        }
    }
}

```

```

// Ищем rulePlugin({ в HTML и извлекаем полный JSON-аргумент
var marker = 'rulePlugin(';
var idx = html.indexOf(marker + '{');
if (idx === -1) return null;

var jsonStart = idx + marker.length;
var jsonStr = extractJsonObject_(html, jsonStart);
if (!jsonStr) return null;

try {
    var config = JSON.parse(jsonStr);
    if (config.rule) {
        return JSON.stringify(config.rule);
    }
} catch (e) {}

return null;
}

/**
 * Извлекает домен из URL сегмента (https://domain.com).
 */
function extractDomain_(url) {
    var m = url.match(/^(https?:\/\/\/[^\/]+)/);
    return m ? m[1] : null;
}

/**
 * Сериализует JS-объект в формат jQuery $.param() (nested keys).
 * {rule: {type: "x", params: {mode: "and"}}} →
 * "rule[type]=x&rule[params][mode]=and"
 */
function jqueryParam_(obj, prefix) {
    var parts = [];
    for (var key in obj) {
        if (!obj.hasOwnProperty(key)) continue;
        var fullKey = prefix ? prefix + '[' + key + ']' : key;
        var val = obj[key];
        if (val !== null && typeof val === 'object') {

```

```

        parts.push(jqueryParam_(val, fullKey));
    } else {
        parts.push(encodeURIComponent(fullKey) + '=' + encodeURIComponent(val === null ? ''
: val));
    }
}
return parts.filter(function(s) { return s; }).join('&');
}

/**
 * Загружает статистику заказов через AJAX-эндпоинт /pl/sales/stat/short-chart-data.
 * Отправляет rule как nested form params (формат jQuery $.param).
 * Возвращает HTML-фрагмент со статистикой.
 */
function fetchDealStatsAjax_(domain, rule, sessionCookie) {
    var statUrl = domain + '/pl/sales/stat/short-chart-data';

    var ruleObj = null;
    if (rule) {
        try { ruleObj = JSON.parse(rule); } catch (e) { ruleObj = null; }
    }

    var payload = jqueryParam_({ rule: ruleObj });

    var response = UrlFetchApp.fetch(statUrl, {
        method: 'post',
        headers: {
            'Cookie': sessionCookie,
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/120.0.0.0 Safari/537.36',
            'Content-Type': 'application/x-www-form-urlencoded',
            'X-Requested-With': 'XMLHttpRequest',
            'Referer': domain + '/pl/sales/deal/index'
        },
        payload: payload,
        muteHttpExceptions: true,
        followRedirects: true
    });

```

```

var code = response.getResponseCode();
if (code !== 200) {
    throw new Error('STAT_HTTP_ERROR_' + code);
}

var body = response.getContentText();
var json;
try {
    json = JSON.parse(body);
} catch (e) {
    throw new Error('STAT_PARSE_ERROR: ' + body.substring(0, 200));
}

if (!json.success || !json.data || !json.data.html) {
    throw new Error('STAT_ERROR: ' + body.substring(0, 200));
}

return json.data.html;
}

/**
 * Парсит общее кол-во заказов со страницы.
 * Источники: summary div или rule-test-block (data-count).
 */
function parseDealPageTotal_(html) {
    // 1. rule-test-block: data-count="487"
    var dataCount = html.match(/data-count="(\d+)"/);
    if (dataCount) {
        return parseInt(dataCount[1], 10);
    }

    // 2. summary: из <b>89301</b> запись
    var summary = html.match(/<div
class="summary">Показаны\s*<b>[^<]+<\b>\s*из\s*<b>([\d\s]+)<\b>/);
    if (summary) {
        return parseInt(summary[1].replace(/\s/g, ''), 10);
    }

    return null;
}

```

```

}

/**
 * Парсит HTML-фрагмент статистики заказов из stat-table (AJAX).
 * Структура:
 *   <b>476</b> заявок
 *   <b>11</b> платных заказов <i>55€</i>
 *   <b>3</b> оплачено 27.27% <b>14€</b>
 */
function parseStatHtml_(statHtml) {
    var stats = {
        paidCount: null,
        paidSum: null,
        completedCount: null,
        completedSum: null
    };

    var paidRow = statHtml.match(/data-is-
priceless="0"[\s\S]*?<b>([\d\s]+)</b>[\s\S]*?платных заказов[\s\S]*?<i>([\d\s.]+)\/);
    if (paidRow) {
        stats.paidCount = parseInt(paidRow[1].replace(/\s/g, ''), 10);
        stats.paidSum = parseFloat(paidRow[2].replace(/\s,/g, ''));
    }

    var completedRow = statHtml.match(/data-
status="payed"[\s\S]*?<b>([\d\s]+)</b>[\s\S]*?оплачено[\s\S]*?<b>([\d\s.]+)\/);
    if (completedRow) {
        stats.completedCount = parseInt(completedRow[1].replace(/\s/g, ''), 10);
        stats.completedSum = parseFloat(completedRow[2].replace(/\s,/g, ''));
    }

    return stats;
}

function updateDealSegmentStats() {
    var ss = SpreadsheetApp.getActiveSpreadsheet();
    var sheet = ss.getActiveSheet();
    var selection = sheet.getActiveRange();

```

```

if (!selection) {
    SpreadsheetApp.getUi().alert('Выделите ячейки со ссылками на сегменты заказов.');
```

return;

```

}

var updated = 0;
var numRows = selection.getNumRows();
var numCols = selection.getNumColumns();
var startRow = selection.getRow();
var startCol = selection.getColumn();

for (var r = 0; r < numRows; r++) {
    for (var c = 0; c < numCols; c++) {
        var cell = sheet.getRange(startRow + r, startCol + c);
        if (!cell.getValue()) continue;

        var url = extractSegmentUrl_(cell);
        if (!url) continue;

        var domain = extractDomain_(url);
        if (!domain) {
            domain = getConfig().domain.replace(/\$/+/ , '');
        }

        var row = startRow + r;
        var col = startCol + c;

        try {
            ss.toast('Загрузка страницы сегмента...', 'GC: Заказы', 60);
            var sessionCookie = getSessionCookie();

            // Шаг 1: загружаем страницу – берём общее кол-во и правило
            var pageHtml = fetchWithSession_(url, sessionCookie);
            if (isLoginPage_(pageHtml)) {
                CacheService.getScriptCache().remove('gc_session');
                sessionCookie = login();
                pageHtml = fetchWithSession_(url, sessionCookie);
                if (isLoginPage_(pageHtml)) {
                    throw new Error('LOGIN_FAILED: Не удалось авторизоваться');
```

```

    }
}

var totalOrders = parseDealPageTotal_(pageHtml);
if (totalOrders === null) {
    sheet.getRange(row, col + 1).setValue('Н/Д – нет данных на
странице').setNote('Ошибка');
    SpreadsheetApp.flush();
    continue;
}

// Шаг 2: получаем правило и вызываем stat endpoint для разбивки
var rule = extractDealRule_(url);
if (!rule) {
    // Для сохранённых сегментов – извлекаем правило из rulePlugin на странице
    var marker = 'rulePlugin(';
    var ruleIdx = pageHtml.indexOf(marker + '{');
    if (ruleIdx !== -1) {
        var jsonStr = extractJsonObject_(pageHtml, ruleIdx + marker.length);
        if (jsonStr) {
            try {
                var config = JSON.parse(jsonStr);
                if (config.rule) rule = JSON.stringify(config.rule);
            } catch (e) {}
        }
    }
}

var stats = { paidCount: null, paidSum: null, completedCount: null, completedSum:
null };
if (rule) {
    ss.toast('Загрузка статистики...', 'GC: Заказы', 60);
    try {
        var statHtml = fetchDealStatsAjax_(domain, rule, sessionCookie);
        stats = parseStatHtml_(statHtml);
    } catch (e) {
        // Если stat endpoint не сработал – оставляем null
    }
}
}

```

```

        var now = Utilities.formatDate(new Date(), Session.getScriptTimeZone(),
'dd.MM.yyyy HH:mm');

        var fields = [
            [totalOrders,      'Кол-во заказов'],
            [stats.paidCount,   'Платных заказов'],
            [stats.paidSum,     'Сумма платных'],
            [stats.completedCount, 'Оплаченных заказов'],
            [stats.completedSum, 'Сумма оплаченных'],
            [now,               'Обновлено']
        ];

        for (var i = 0; i < fields.length; i++) {
            var val = fields[i][0];
            sheet.getRange(row, col + 1 + i).setValue(val !== null ? val :
'Н/Д').setNote(fields[i][1]);
        }
        SpreadsheetApp.flush();
        updated++;
    } catch (e) {
        if (e.message.indexOf('LOGIN_FAILED') !== -1) {
            SpreadsheetApp.getUi().alert('Ошибка авторизации: ' + e.message);
            return;
        }
        sheet.getRange(row, col + 1).setValue('Ошибка: ' + e.message).setNote('Ошибка');
        SpreadsheetApp.flush();
    }

    Utilities.sleep(500);
}

}

ss.toast('Обновлено: ' + updated, 'GC: Заказы', 5);
}

function debugDealSegment() {
    // Тест с сохранённым сегментом (89301 заказов)
    var testUrl =

```



```
'https://fitnessmama.school/pl/sales/deal/index?DealContext%5Bsegment_id%5D=3010534&DealContext%5Brule_string%5D=&formParams%5Bclarity_uid%5D=fHNB_TNrF_imWfnGQA0YK225r030UbZK';
```

```
Logger.log('=== DEBUG: page total + stat breakdown ===');
```

```
try {
```

```
    var domain = extractDomain_(testUrl);
```

```
    var sessionCookie = getSessionCookie();
```

```
    // Шаг 1: загружаем страницу
```

```
    var pageHtml = fetchWithSession_(testUrl, sessionCookie);
```

```
    var totalOrders = parseDealPageTotal_(pageHtml);
```

```
    Logger.log('Общее кол-во (со страницы): ' + totalOrders);
```

```
    // Шаг 2: извлекаем правило и вызываем stat
```

```
    var rule = extractDealRule_(testUrl);
```

```
    if (!rule) {
```

```
        var marker = 'rulePlugin(';
```

```
        var ruleIdx = pageHtml.indexOf(marker + '{');
```

```
        if (ruleIdx !== -1) {
```

```
            var jsonStr = extractJsonObject_(pageHtml, ruleIdx + marker.length);
```

```
            if (jsonStr) {
```

```
                var config = JSON.parse(jsonStr);
```

```
                if (config.rule) rule = JSON.stringify(config.rule);
```

```
            }
```

```
        }
```

```
    }
```

```
    Logger.log('Rule: ' + (rule ? rule.substring(0, 100) + '...' : 'нет'));
```

```
    if (rule) {
```

```
        var statHtml = fetchDealStatsAjax_(domain, rule, sessionCookie);
```

```
        var stats = parseStatHtml_(statHtml);
```

```
        Logger.log('Платных: ' + stats.paidCount + ', сумма: ' + stats.paidSum);
```

```
        Logger.log('Оплаченных: ' + stats.completedCount + ', сумма: ' +
```

```
stats.completedSum);
```

```
    }
```

```
} catch (e) {
```

```
    Logger.log('ERROR: ' + e.message);
```

```
}
```

```

}

// =====
// Тест подключения
// =====

function testConnection() {
    var ui = SpreadsheetApp.getUi();
    var config = getConfig();

    try {
        CacheService.getScriptCache().remove('gc_session');
        var sessionCookie = login();
        ui.alert('Подключение успешно!',
            'Email: ' + config.email + '\n' +
            'Домен: ' + config.domain + '\n' +
            'Cookie: ' + sessionCookie.substring(0, 40) + '...\n\n' +
            'Авторизация работает. Можно обновлять статистику.',
            ui.ButtonSet.OK);
    } catch (e) {
        ui.alert('Ошибка подключения', e.message, ui.ButtonSet.OK);
    }
}

/**
 * Отладка сегмента: показывает что возвращает сервер для URL сегмента.
 * Замените testUrl на нужную ссылку, запустите из Apps Script.
 */
function debugSegment() {
    var testUrl =
'https://fitnessmama.school/pl/user/user/index?uc%5Bsegment_id%5D=0&uc%5Brule_string%5D=%7B%22type%22%3A%22user_createdat%22%2C%22inverted%22%3A0%2C%22params%22%3A%7B%22value%22%3A%7B%22from%22%3Anull%2C%22to%22%3Anull%2C%22toNDays%22%3A%2230%22%2C%22fromNDays%22%3Anull%2C%22dateType%22%3A%22last_n_days%22%2C%22withTime%22%3Afalse%7D%2C%22valueMode%22%3Anull%7D%2C%22maxSize%22%3A%22%22%7D&formParams%5Bclarity_uid%5D=JoSTNr9IVThIbcbL_0LoVFNrnDiOXnbX';

    try {
        var html = fetchSegmentPage(testUrl);
    }
}

```

```

var title = html.match(/<title>([<]+)<\/title>/);
var isLogin = isLoginPage_(html);
var hasFilterCount = html.indexOf('filter-count') !== -1;
var hasDataCount = html.indexOf('data-count') !== -1;
var count = parseSegmentCount(html);

Logger.log('=== DEBUG segment ===');
Logger.log('URL: ' + testUrl.substring(0, 80) + '...');
Logger.log('Title: ' + (title ? title[1] : 'не найден'));
Logger.log('isLoginPage: ' + isLogin);
Logger.log('hasFilterCount: ' + hasFilterCount);
Logger.log('hasDataCount: ' + hasDataCount);
Logger.log('parsedCount: ' + count);
Logger.log('HTML length: ' + html.length);
Logger.log('First 2000 chars: ' + html.substring(0, 2000));
} catch (e) {
    Logger.log('ERROR: ' + e.message);
}
}

/**
 * Отладка: показывает что возвращает сервер для конкретного mailing ID.
 * Запустите вручную из Apps Script, изменив ID ниже.
 */
function debugMailing() {
    var testId = 4508232; // Замените на нужный ID
    CacheService.getScriptCache().remove('gc_session');

    try {
        var html = fetchMailingPage(testId);
        var title = html.match(/<title>([<]+)<\/title>/);
        var hasStats = html.indexOf('Статистика рассылки') !== -1;
        var hasUserStats = html.indexOf('просмотр') !== -1;
        var isLogin = isLoginPage_(html);
        var bodyClass = html.match(/class="(["]*page-["]*)"/) || ['', 'не найден'];

        Logger.log('=== DEBUG mailing ID: ' + testId + ' ===');
        Logger.log('Title: ' + (title ? title[1] : 'не найден'));
        Logger.log('Body class: ' + bodyClass[1]);
    }
}

```

```
    Logger.log('isLoginPage: ' + isLogin);
    Logger.log('hasStats: ' + hasStats);
    Logger.log('hasUserStats: ' + hasUserStats);
    Logger.log('HTML length: ' + html.length);
    Logger.log('First 1000 chars: ' + html.substring(0, 1000));
  } catch (e) {
    Logger.log('ERROR: ' + e.message);
  }
}
```

---

Revision #7

Created 2026-04-07 15:42:20 UTC by David

Updated 2026-04-10 14:29:48 UTC by David