/\*\*  C | F | G • Apps Script  — v5.1 fail-safe

 \*  Подписанные запросы проходят в обычном режиме.

 \*  Если подпись не сошлась, заявка всё равно принимается (sig\_status='weak').

 \*  Структура таблицы не меняется.

 \*/

/\* === ENV (Script Properties) === \*/

var PROPS        = PropertiesService.getScriptProperties().getProperties();

var SS\_ID        = PROPS.SPREADSHEET\_ID || '';

var EMAIL\_TO     = PROPS.EMAIL\_TO || '';

var TG\_TOKEN     = PROPS.BOT\_TOKEN || '';

var TG\_CHAT      = Number(PROPS.CHAT\_ID || 0) || 0;

var CFG\_KEY      = PROPS.CFG\_KEY || '';

var CFG\_KEY\_V2   = PROPS.CFG\_KEY\_V2 || '';

var CFG\_SECRET   = PROPS.CFG\_SECRET || '';

/\* === HTTP === \*/

function doGet(e) {

  if (e && e.parameter && e.parameter.test === 'config') {

    if (String(e.parameter.token || '') === String(CFG\_SECRET || '')) {

      return json({ ok: true });

    }

    return json({ ok: false, error: 'forbidden' });

  }

  return json({

    ok: true,

    mode: 'status',

    env: { has\_CFG\_KEY: !!CFG\_KEY, has\_CFG\_KEY\_V2: !!CFG\_KEY\_V2 }

  });

}

function doPost(e) {

  try {

    if (!CFG\_KEY && !CFG\_KEY\_V2) return json({ ok:false, error:'env\_not\_configured' });

    // 1) вход

    var parsed = parsePayload(e) || {};

    var inputSig = parsed && parsed.\_sig;

    // 2) clone без \_sig

    var clone = JSON.parse(JSON.stringify(parsed || {}));

    if (clone.\_sig != null) delete clone.\_sig;

    // 3) канон

    var canon = stablePayload(clone);

    var canonStr = JSON.stringify(canon);

    // 4) проверка подписи (канон и «сырой», оба ключа)

    var accepted = false;

    var keys = [CFG\_KEY, CFG\_KEY\_V2].filter(Boolean);

    for (var i = 0; i < keys.length && !accepted; i++) {

      var k = keys[i];

      var hexCanon = hmacHex(canonStr, k);

      var hexRaw   = hmacHex(JSON.stringify(clone), k);

      if (inputSig && (inputSig === hexCanon || inputSig === hexRaw)) accepted = true;

    }

    // 5) если подпись не ок — мягкий фолбэк (принимаем как есть)

    var sig\_status = accepted ? 'ok' : 'weak';

    // 6) нормализованные данные → таблица и уведомления

    var when = new Date();

    var meta = {

      when: when,

      page\_url: String(canon.page\_url || ''),

      referrer: String(canon.referrer || ''),

      utm: extractUTM(canon),

      sig\_status: sig\_status

    };

    var sheetRes = tryWrap(function(){ return writeRow(canon, meta); });

    tryWrap(function(){ return notifyEmail(canon, meta); });

    tryWrap(function(){ return notifyTelegram(canon, meta); });

    return json({

      ok: true,

      sheet: sheetRes && sheetRes.sheet,

      range: sheetRes && sheetRes.rangeA1,

      id: sheetRes && sheetRes.rowId

    });

  } catch (err) {

    return json({ ok:false, error:'server\_crash', message:String(err && err.message || err) });

  }

}

/\* === Storage / Notifications === \*/

function writeRow(data, meta) {

  if (!SS\_ID) return { error:'no\_spreadsheet' };

  var ss = SpreadsheetApp.openById(SS\_ID);

  var sheet = ss.getSheets()[0];

  var row = [

    meta.when,

    data.action,

    data.locale,

    data.name,

    data.company,

    data.inn,

    data.email,

    data.phone,

    data.desc,

    data.industry,

    data.revenue,

    data.geo,

    data.urgency,

    data.page\_url,

    data.referrer,

    data.utm\_source || '',

    data.utm\_medium || '',

    data.utm\_campaign || '',

    data.utm\_term || '',

    data.utm\_content || ''

  ];

  var last = sheet.getLastRow();

  var range = sheet.getRange(last + 1, 1, 1, row.length);

  range.setValues([row]);

  // Лёгкий лог в журнал (без секретов)

  try { Logger.log('lead #%s sig=%s url=%s', (last+1), meta.sig\_status, data.page\_url||''); } catch(\_){}

  return { sheet: sheet.getName(), rangeA1: range.getA1Notation(), rowId: last + 1 };

}

function notifyEmail(data, meta) {

  if (!EMAIL\_TO) return;

  var when = Utilities.formatDate(meta.when, 'Etc/UTC', 'yyyy-MM-dd HH:mm:ss') + 'Z';

  var body = [

    'Время: ' + when,

    'Имя: ' + (data.name || ''),

    'Компания: ' + (data.company || ''),

    'Телефон: ' + (data.phone || ''),

    'E-mail: ' + (data.email || ''),

    'Комментарий: ' + (data.desc || ''),

    'URL: ' + (meta.page\_url || ''),

    'Referrer: ' + (meta.referrer || ''),

    'UTM: ' + JSON.stringify(meta.utm || {}),

    'Подпись: ' + meta.sig\_status

  ].join('\n');

  MailApp.sendEmail({ to: EMAIL\_TO, subject: 'CFG: новая заявка — ' + (data.action || 'lead'), htmlBody: body.replace(/\n/g,'<br>') });

}

function notifyTelegram(data, meta) {

  if (!TG\_TOKEN || !TG\_CHAT) return;

  var when = Utilities.formatDate(meta.when, 'Etc/UTC', 'yyyy-MM-dd HH:mm:ss') + 'Z';

  var msg =

    'CFG • заявка' +

    '\n— ' + (data.action || 'lead') +

    '\nИмя: ' + (data.name || '') +

    '\nКомпания: ' + (data.company || '') +

    '\nТелефон: ' + (data.phone || '') +

    '\nEmail: ' + (data.email || '') +

    '\nКомментарий: ' + (data.desc || '') +

    '\nURL: ' + (meta.page\_url || '') +

    '\nПодпись: ' + meta.sig\_status +

    '\nВремя: ' + when;

  var url = 'https://api.telegram.org/bot' + TG\_TOKEN + '/sendMessage';

  UrlFetchApp.fetch(url, {

    method: 'post',

    contentType: 'application/json',

    payload: JSON.stringify({ chat\_id: TG\_CHAT, text: msg }),

    muteHttpExceptions: true

  });

}

/\* === Utils === \*/

function json(obj){ var o=ContentService.createTextOutput(JSON.stringify(obj)); o.setMimeType(ContentService.MimeType.JSON); return o; }

function tryWrap(fn){ try{ return fn(); } catch(e){ return { ok:false, error:String(e && e.message || e) }; } }

function hmacHex(s, key){ var b=Utilities.computeHmacSha256Signature(s,key); return b.map(function(x){ x=(x+256)%256; var h=x.toString(16); return h.length===1?'0'+h:h; }).join(''); }

function extractUTM(data){

  var out={}; ['utm\_source','utm\_medium','utm\_campaign','utm\_term','utm\_content'].forEach(function(k){

    if (data && data[k] != null) out[k]=String(data[k]);

    else if (data && data.utm && data.utm[k] != null) out[k]=String(data.utm[k]);

  });

  return out;

}

function sanitizeData(d){

  d=d||{};

  var S=function(v){ return String(v==null?'':v).replace(/[\u00A0\u202F\u2007\u2060]/g,' ').replace(/\s+/g,' ').trim(); };

  var P=function(v){ v=String(v==null?'':v); v=v.replace(/[^+\d]/g,''); if (v && v[0] !== '+' && /^8\d{10}$/.test(v)) v='+7'+v.slice(1); return v; };

  var o={

    action:S(d.action || 'estimate\_24h'),

    locale:S(d.locale || 'ru'),

    name:S(d.name || d.fullname),

    company:S(d.company || d.org),

    inn:S(d.inn || d.tax\_id),

    email:S(d.email),

    phone:P(d.phone),

    desc:S(d.message || d.desc || d.comment || d.task),

    industry:S(d.industry),

    revenue:S(d.revenue),

    geo:S(d.geo),

    urgency:S(d.urgency),

    page\_url:S(d.page\_url),

    referrer:S(d.referrer)

  };

  ['utm\_source','utm\_medium','utm\_campaign','utm\_term','utm\_content'].forEach(function(k){

    if (d && d[k] != null) o[k]=S(d[k]);

    else if (d && d.utm && d.utm[k] != null) o[k]=S(d.utm[k]);

  });

  return o;

}

function stablePayload(d){

  var o=sanitizeData(d);

  var keys=Object.keys(o).sort();

  var ordered={}; for (var i=0;i<keys.length;i++) ordered[keys[i]]=o[keys[i]];

  return ordered;

}

function parsePayload(e){

  var out={};

  if (e && e.postData && typeof e.postData.contents==='string'){

    var ct=String(e.postData.type||''); var body=e.postData.contents;

    if (ct.indexOf('application/json')!==-1){

      try{ out=JSON.parse(body); }catch(\_){}

    } else {

      try{ body.split('&').forEach(function(kv){ if(!kv) return; var p=kv.split('='); var k=decodeURIComponent(p[0]||''); var v=decodeURIComponent((p[1]||'').replace(/\+/g,' ')); if(k) out[k]=v; }); }catch(\_){}

    }

  }

  var params=(e && e.parameter) || {};

  for (var k in params){ if (params.hasOwnProperty(k)) out[k]=params[k]; }

  return out;

}