

Скрипт отслеживания перехода в мессенджеры

Скрипт фиксирует реальные переходы в мессенджеры.

Это очень сильный сигнал для алгоритма, потому что переход в мессенджер — почти всегда намерение связаться.

Инструкция

Просто вставьте номер вашего счетчика ниже и получите готовый скрипт.

Не забудь настроить цели, чтобы отслеживать события через JavaScript-событие

messenger_slider_success

Срабатывает при успешном прохождении слайдера (когда пользователь провел ползунок вправо до конца) для любого мессенджера

messenger_slider_success_telegram

Срабатывает при успешном прохождении слайдера для перехода в Telegram

messenger_slider_success_whatsapp

Срабатывает при успешном прохождении слайдера для перехода в WhatsApp

messenger_slider_success_max

Срабатывает при успешном прохождении слайдера для перехода в Max

Скрипт отслеживания перехода в мессенджеры

<!-- ADSкий скрипт — мессенджеры (Telegram / WhatsApp / Max) --> <script>

/**

* =====

* Скрипт защиты от ботов для мессенджеров

* Разработка: Истомин Антон



```

* TG-блог: "ADСкий Яндекс Директ"
* =====
*/
(function () {
    var COUNTER_ID = 11111111;
    var OVERLAY_ID = "pre-redirect-overlay";
    var lastTriggerTs = 0;

    function sendGoal(goal) {
        if (typeof ym === "undefined") return;
        try { ym(COUNTER_ID, "reachGoal", goal); } catch (e) {}
    }

    function getMessengerName(url) {
        if (!url) return null;
        var s = String(url).trim().toLowerCase();

        if (s.indexOf("tg://") === 0) return "telegram";
        if (s.indexOf("whatsapp://") === 0) return "whatsapp";
        if (s.indexOf("https://t.me") === 0 || s.indexOf("http://t.me") === 0) return "telegram";
        if (s.indexOf("https://telegram.me") === 0 || s.indexOf("http://telegram.me") === 0)
            return "telegram";
        if (s.indexOf("https://wa.me") === 0 || s.indexOf("http://wa.me") === 0) return
            "whatsapp";
        if (s.indexOf("https://api.whatsapp.com") === 0 ||
            s.indexOf("http://api.whatsapp.com") === 0) return "whatsapp";
        if (s.indexOf("https://max.ru") === 0 || s.indexOf("http://max.ru") === 0) return "max";

        try {
            var u = new URL(url, window.location.origin);
            var full = (u.hostname + u.pathname).toLowerCase();

```



```
    if (full.indexOf("t.me") !== -1 || full.indexOf("telegram.me") !== -1) return "telegram";  
    if (full.indexOf("wa.me") !== -1 || full.indexOf("api.whatsapp.com") !== -1) return  
"whatsapp";  
    if (full.indexOf("max.ru") !== -1) return "max";  
    } catch (e) {}  
  
    return null;  
}
```

```
function hardStop(e) {  
    if (!e) return;  
    e.preventDefault();  
    e.stopPropagation();  
    if (typeof e.stopImmediatePropagation === "function")  
e.stopImmediatePropagation();  
    e.cancelBubble = true;  
}
```

```
function patchLink(a) {  
    if (!a || a.nodeType !== 1) return null;  
    if (a.getAttribute("data-msg-patched") === "1") return a;
```

```
    var href = a.getAttribute("href");  
    var name = getMessengerName(href);  
    if (!name) return null;
```

```
    var clone = a.cloneNode(true);  
    clone.setAttribute("data-msg-patched", "1");  
    clone.setAttribute("data-msg-name", name);  
    clone.setAttribute("data-msg-href", href);  
    clone.setAttribute("href", "#");
```



```
clone.removeAttribute("target");
clone.removeAttribute("onclick");
clone.onclick = null;

if (a.parentNode) a.parentNode.replaceChild(clone, a);
return clone;
}

function patchAll(root) {
    var scope = root && root.querySelectorAll ? root : document;
    if (scope.matches && scope.matches("a[href]")) patchLink(scope);

    var links = scope.querySelectorAll ? scope.querySelectorAll("a[href]") : [];
    for (var i = 0; i < links.length; i++) patchLink(links[i]);
}

function closeOverlay() {
    var el = document.getElementById(OVERLAY_ID);
    if (el && el.parentNode) el.parentNode.removeChild(el);
}

function showSliderAndRedirect(url, messengerName) {
    if (document.getElementById(OVERLAY_ID)) return;

    var title =
        messengerName === "telegram" ? "Открываем Telegram..." :
        messengerName === "whatsapp" ? "Открываем WhatsApp..." :
        messengerName === "max" ? "Открываем Max..." :
        "Открываем мессенджер...";
```



```

var overlay = document.createElement("div");

overlay.id = OVERLAY_ID;

overlay.style.cssText =
"position:fixed;inset:0;background:rgba(0,0,0,.6);z-index:2147483647;display:flex;align-items:center;justify-content:center;cursor:pointer;";

var box = document.createElement("div");

box.style.cssText = "background:#fff;border-radius:16px;padding:28px
24px;max-width:90%;width:340px;text-align:center;box-shadow:0 8px 32px
rgba(0,0,0,.25);font-family:sans-serif;position:relative;cursor:default;z-index:2147483647
;";

box.innerHTML =

'<div style="font-size:20px;font-weight:700;margin-bottom:12px;">' + title + "</div>" +

'<div
style="font-size:14px;color:#333;margin-bottom:16px;line-height:1.45;">Проведите
ползунок вправо, чтобы перейти</div>' +

'<div id="msg-slider"
style="width:100%;height:50px;background:#e0e0e0;border-radius:25px;position:relative
;margin-bottom:12px;overflow:hidden;touch-action:none;">' +

'<div id="msg-track"
style="position:absolute;top:0;left:0;height:100%;width:100%;"></div>' +

'<div id="msg-handle"
style="position:absolute;top:5px;left:5px;width:40px;height:40px;background:#4285f4;bo
rder-radius:50%;cursor:grab;touch-action:none;display:flex;align-items:center;justify-cont
ent:center;color:#fff;font-size:20px;user-select:none;">→</div>' +

'<div id="msg-slider-text"
style="position:absolute;top:50%;left:50%;transform:translate(-50%,-50%);color:#777;poi
nter-events:none;font-size:14px;">Проведите вправо</div>' +

"</div>";

overlay.appendChild(box);

document.body.appendChild(overlay);

var slider = box.querySelector("#msg-slider");
var handle = box.querySelector("#msg-handle");

```



```
var track = box.querySelector("#msg-track");

var dragging = false;
var startX = 0;
var startLeft = 5;
var maxLeft = 0;

requestAnimationFrame(function () {
    maxLeft = slider.offsetWidth - handle.offsetWidth - 10;
});

function getX(e) {
    if (e.touches && e.touches[0]) return e.touches[0].clientX;
    if (e.changedTouches && e.changedTouches[0]) return e.changedTouches[0].clientX;
    return e.clientX;
}

function onStart(e) {
    dragging = true;
    startX = getX(e);
    startLeft = parseInt(handle.style.left, 10) || 5;
    handle.style.cursor = "grabbing";
    handle.style.background = "#3367d6";
    hardStop(e);
}

function onMove(e) {
    if (!dragging) return;
    var x = getX(e);
    var next = Math.max(5, Math.min(startLeft + (x - startX), maxLeft));
```



```
handle.style.left = next + "px";
var progress = maxLeft > 0 ? (next - 5) / maxLeft : 0;
track.style.background = "linear-gradient(to right,#a0c3ff " + (progress * 100) +
"%,#e0e0e0 " + (progress * 100) + "%)";
hardStop(e);
}
```

```
function reset() {
  handle.style.transition = "left .25s ease";
  handle.style.left = "5px";
  track.style.background = "#e0e0e0";
  setTimeout(function () { handle.style.transition = ""; }, 250);
}
```

```
function cleanup() {
  document.removeEventListener("mousemove", onMove);
  document.removeEventListener("touchmove", onMove);
  document.removeEventListener("mouseup", onEnd);
  document.removeEventListener("touchend", onEnd);
  handle.removeEventListener("mousedown", onStart);
  handle.removeEventListener("touchstart", onStart);
  window.removeEventListener("pageshow", onPageShow);
}
```

```
function onPageShow() {
  closeOverlay();
  window.removeEventListener("pageshow", onPageShow);
}
```

```
function onEnd(e) {
```



```
if (!dragging) return;
dragging = false;
handle.style.cursor = "grab";
handle.style.background = "#4285f4";

var pos = parseInt(handle.style.left, 10) || 5;
var success = pos >= (maxLeft - 2);

if (success) {
    sendGoal("messenger_slider_success");
    sendGoal("messenger_slider_success_" + messengerName);
    sendGoal("messenger_redirect_" + messengerName);

    cleanup();
    closeOverlay();

    try { window.location.href = url; } catch (err) { window.open(url, "_self"); }
} else {
    reset();
}
hardStop(e);
}

overlay.addEventListener("click", function (e) {
    if (e.target === overlay) {
        cleanup();
        closeOverlay();
    }
    hardStop(e);
}, true);
```



```
handle.addEventListener("mousedown", onStart);
handle.addEventListener("touchstart", onStart, { passive: false });
document.addEventListener("mousemove", onMove, { passive: false });
document.addEventListener("touchmove", onMove, { passive: false });
document.addEventListener("mouseup", onEnd, { passive: false });
document.addEventListener("touchend", onEnd, { passive: false });
window.addEventListener("pageshow", onPageShow);
}
```

```
function findPatchedLink(target) {
  if (!target || !target.closest) return null;
  return target.closest("a[data-msg-patched='1'][data-msg-href]");
}
```

```
function globalBlocker(e) {
  var link = findPatchedLink(e.target);
  if (!link) return;

  hardStop(e);

  var type = e.type;
  if (type !== "click" && type !== "touchend" && type !== "pointerup") return;

  var now = Date.now();
  if (now - lastTriggerTs < 400) return;
  lastTriggerTs = now;

  var url = link.getAttribute("data-msg-href");
  var name = link.getAttribute("data-msg-name") || getMessengerName(url);
```



```
if (!url || !name) return;

showSliderAndRedirect(url, name);
}

patchAll(document);

var mo = new MutationObserver(function (mutations) {
  for (var i = 0; i < mutations.length; i++) {
    var m = mutations[i];
    for (var j = 0; j < m.addedNodes.length; j++) {
      var node = m.addedNodes[j];
      if (node && node.nodeType === 1) patchAll(node);
    }
  }
});

mo.observe(document.documentElement, { childList: true, subtree: true });

["pointerdown", "mousedown", "touchstart", "click", "auxclick", "touchend",
"pointerup"].forEach(function (evt) {
  window.addEventListener(evt, globalBlocker, { capture: true, passive: false });
});
})();
</script>
```

