Humanometr. Руководство для разработчиков

Виджет Humanometr может защитить ваши приложения от ботов, спама и других форм автоматизированного злоупотребления. Установка Humanometr выполняется быстро и легко. Это требует добавления некоторого простого HTML и кода на стороне сервера.

Основные принципы

Вы встраиваете виджет Humanometr на свой сайт. Например, на форме входа. Пользователь отвечает на Humanometr и получает код доступа с нашего сервера. Когда пользователь нажимает на форме кнопку «Отправить», код доступа, полученный ранее от Humanometr, отправляется на ваш сервер. Затем ваш сервер проверяет этот код с помощью API сервера Humanometr. Сервис Humanometr отвечает, что код действителен. Ваш сервер теперь знает, что пользователь не является ботом, и позволяет ему войти в систему.

Процесс работы

Основные понятия

- `Пользователь`: Человек или бот который взаимодействует с сайтом посредством браузера.
- `Браузер`: Интернет-браузер.
- `Сервер сайта`: Серверная часть вашего приложения.
- `Сервис капчи`: Сервер на котором работает Humanometr.
- `Хост с виджетом`: Сервер на котором размещен виджет Humanometr.
- `Виджет`: HTML-код капчи Humanometr, который встраивается на страницу вашего приложения.
- `Страница сайта`: Страница вашего приложения, на которой встраивается виджет Humanometr.

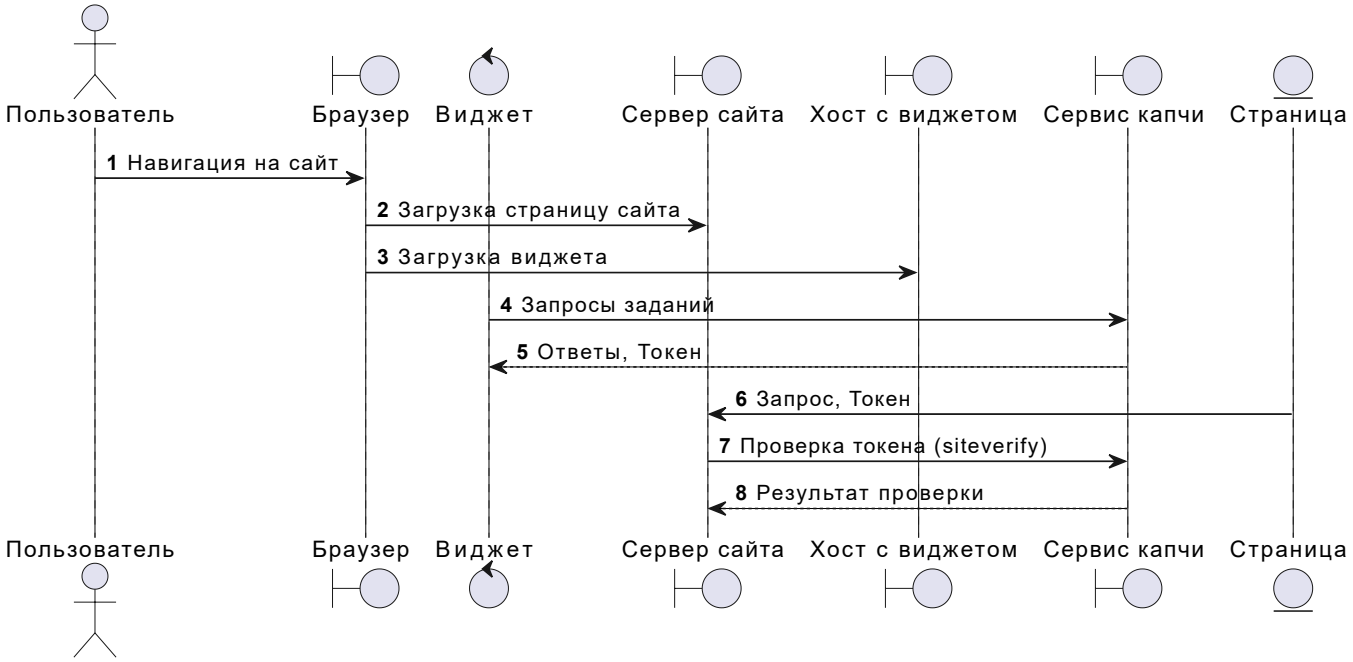
В зависимости от способа установки адреса сервиса капчи и хоста с виджетом могут отличаться.

Локальное развертывание образов из архива с ознакомительной версией

- Сервис капчи: `http://localhost:50011`
- Хост с виджетом: `http://localhost:8080/humanometr.js`

Магазин приложений VK

Адреса развёртывания Вы получаете вместе с информационным письмом после приобретения сервиса Humanometr.



Описание взаимодействия

Пользователь запрашивает страницу вашего прилодения через бразузер. Страница загрудается в браузер и догружает виджет капчи с хоста. Виджет капчи начинает взаимодействие с сервисом капчи, а также с ващей страницей посредством предоставляемого API. После того, как проверка завершена (т.е. пользователь прощел все запланированные испытания), сервис капчи присылает виджету токен. Данный токен долден быть далее передан на серер вашего сайта, посредством, например, POST запроса формы. Далее сервер сайта обращается к сервису капчи для проверки токена и получения сведения об уровне доверия.

Настройка на защищаемой странице

Добавьте виджет Humanometr на свою веб-страницу

Humanometr требует несколько небольших фрагментов кода на стороне клиента для отображения виджета с капчей на HTML-странице. Во-первых, вы должны включить ресурс javascript Humanometr в начале своей HTML-страницы между тегами `<head></head>`. `<script>` должен быть загружен через HTTPS `id="humanometr-script"`

```
<script id="humanometr-script" src="https://<widget-host>/index.js"></script>
```

Теперь, вы должны добавить виджет Humanometr в разметку.

```
<humanometr-widget
        id='<идентификатор_для_доступа_к_виджету>'
        action="<имя_действия>"
        host="<url_сайта>"
        sitekey="<ключ_сайта>"
```

```
  >
  </humanometr-widget>
```

`id` – Идентификатор для доступа к виджету из вашего клиентского кода

`action` - Имя действия

`host` - URL сайта, где размещен скрипт виджета капчи.

`sitekey` – ключ сайта, который вы можете найти на странице своего профиля.

Для работы с виджетом на стороне клиента предоставлено следующее API:

```
// Получаем доступ к элементу виджета
const humanometr = document.getElementById('<id_виджета>');

// Получение токена осуществляется асинхронно с использованием Promise,
// чтобы обеспечить максимальную актуальность токена

humanometr
    .getToken()
    .then(token => {
        // Действия над токеном
    })
    .catch(e => {
        // Обработка исключений
    })

// Запрос на получение новых параметров виджета (срабатывает автоматически
// при загрузке старницы)

humanometr.getCaptcha();

// Чтобы иметь возможность отреагировать на успешное или неуспешное
// прохождение испытаний, следует добавить следующий код:

document.addEventListener('HumanometrEvent', event => {
    if (event.detail.name === 'ChallengePassEvent') {
        // Код, срабатывающий при успешном прохождении
    }

    if (event.detail.name === 'ChallengeFailEvent') {
        // Код, срабатывающий, если испытание не пройдено
    }
})
```

Проверка на стороне сервера

Добавив код на стороне клиента, вы смогли отобразить виджет Humanometr, который определяет, являются ли пользователи реальными людьми или автоматизированными ботами. Когда капча прошла

успешно, сервис Humanometr сохранит на своей стороне данные об этом пользователе. Все что остается – это запросить данные о пользователе с сервиса Humanometr. Для этого требуется на серверной стороне вашего приложения выполнить POST запрос `https://<humanometr-service-host>/siteverify` со следующими параметрами в json-теле запроса:

```
{
  "secret":"<secret>",
  "token": "<token>",
  "remoteip":"<remoteip>",
  "sitekey":"<sitekey>"
}
```

- `secret` – строка секрета, которая создана для конкретного сайта.
- `token` – token, переданный с клиентской стороны, который ранее был получен виджетом от сервиса капчи. Для того чтобы передать его на серверную часть, необходимо вызвать метод getToken() виджета.
- `sitekey` – ключ сайта, для которого осуществляется проверка.
- `remoteip` – ip-адрес пользователя, который вы определяете на серверной части вашего приложения (опциональный параметр).

Пример вызова

```
curl -X 'POST' \
 'http://localhost:5001/siteverify' \ -
H 'accept: text/plain' \
 -H 'Content-Type: application/json' \
 -d '{
 "secret": "<secret>",
 "token": "<token>",
 "sitekey": "<sitekey>"
}',
```

Таким образом в ответе запроса `/siteverify` вы получите данные о том является ли пользователь ботом либо человеком и сможете принимать решение на стороне вашего приложения о том какие возможности будут предоставлены пользователю. Ответ `/siteverify` будет выглядеть сделующим образом:

```
{
    "success":true,
    "challenge_ts":{
        "ticks":0,
        "days":0,
        "hours":0,
        "milliseconds":0,
        "minutes":0,
        "seconds":0,
```

```
        "totalDays":0,
        "totalHours":0,
        "totalMilliseconds":0,
        "totalMinutes":0,
        "totalSeconds":0
    },
    "hostname":"",
    "credit":null,
    "error_Codes":[],
    "score":0,
    "score_Reason":[]
}
```

Пример страницы сайта с виджетом Humanometr

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <script id="humanometr-script" src="http://localhost:8080/humanometr.js">
</script>
    <title>Отзыв на Гуманометр</title>
</head>

<body>
    <h1>Оцените работу Гуманометра</h1>
    <form id="feedbackForm">
        <div class="stars">
            <input type="radio" id="star5" name="rate" value="5"><label
for="star5"></label>
            <input type="radio" id="star4" name="rate" value="4"><label
for="star4"></label>
            <input type="radio" id="star3" name="rate" value="3"><label
for="star3"></label>
            <input type="radio" id="star2" name="rate" value="2"><label
for="star2"></label>
            <input type="radio" id="star1" name="rate" value="1"><label
for="star1"></label>
        </div>
        <div>
            <label for="feedbackText">Мнение (опционально)</label>
            <textarea id="feedbackText" name="feedbackText" rows="4"
                placeholder="Оставьте здесь своё мнение о работе Гуманометра">
</textarea>
        </div>

        <div>
```

```
            <humanometr-widget
                id="humanometr-widget"
                host="http://localhost:50011"
                sitekey="10000000-ffff-ffff-ffff-000000000001">
            </humanometr-widget>
        </div>

        <button type="submit" id="submitBtn" disabled>Отправить отзыв</button>
    </form>
    <div id="submittedFeedback" style="display:none;">
        <h2>Оставленный отзыв</h2>
        <div class="feedback-item">
            <span class="feedback-label">Рейтинг:</span>
            <span class="feedback-value" id="ratingDisplay"></span>
        </div>
        <div class="feedback-item">
            <span class="feedback-label">Мнение:</span>
            <span class="feedback-value" id="opinionDisplay"></span>
        </div>
        <div class="feedback-item">
            <span class="feedback-label">Токен:</span>
            <span class="feedback-value" id="tokenDisplay" style="white-space:
normal;"></span>
        </div>
        <div class="feedback-item">
            <label for="secretInput">Секрет:</label>
            <input type="password" id="secretInput" placeholder="Введите
секрет...">
        </div>
        <button type="button" id="verifyTokenButton" disabled>Проверить
токен</button>
        <p id="tokenHint" style="display: none; color: #888; font-size: 0.9em;">
            Важно: Вызов siteverify должен в общем случае осуществляться с сервера
сайта, так как требует аутентификации
            с использованием секрета.
        </p>
        <div id="responseJson"></div>
        <button type="button" id="getCaptcha">Получить капчу снова</button>
    </div>
    <script>
        let token = null;

        const humanometr = document.getElementById('humanometr-widget');

        // Функция для проверки разблокировки кнопки отправки
        // Кнопка становится активна, если выбран рейтинг и получен валидный токен
        const checkSubmitBtn = () => {
            const submitBtn = document.getElementById('submitBtn');
            if (document.querySelector('input[name="rate"]:checked') && token) {
                submitBtn.disabled = false;
            } else {
                submitBtn.disabled = true;
            }
        }
```

```
        // Функция для обработки обновления токена
        // Отображает или скрывает элементы зависящие от наличия токена
        const handleTokenUpdate = (t) => {
            token = t;
            checkSubmitBtn();
            const verifyBtn = document.getElementById('verifyTokenButton');
            const tokenHint = document.getElementById('tokenHint');
            const getCaptchaBtn = document.getElementById('getCaptcha');
            if (token) {
                verifyBtn.style.display = 'block';
                tokenHint.style.display = 'block';
                getCaptchaBtn.style.display = 'block';
            } else {
                verifyBtn.style.display = 'none';
                tokenHint.style.display = 'none';
                getCaptchaBtn.style.display = 'none';
            }
        }

        // Проверка активности кнопки проверки токена.
        // Предполагается, что секрет должен быть введён пользователем и это поле
обязательное
        const checkVerifyButtonState = () => {
            const verifyBtn = document.getElementById('verifyTokenButton');
            const secret = document.getElementById('secretInput').value;
            verifyBtn.disabled = !secret;
        }

        // Обработчик событий капчи
        document.addEventListener('HumanometrEvent', event => {
            if (event.detail.name === 'ChallengePassEvent') {
                humanometr
                    .getToken()
                    .then(t => {
                        // Если капча успешно пройдена и получен токен
                        handleTokenUpdate(t);
                    })
                    .catch(error => {
                        handleTokenUpdate(null);
                        alert(error);
                    })
            }

            if (event.detail.name === 'ChallengeFailEvent') {
                handleTokenUpdate(null);
            }
        })

        // Слушатель изменения выбора рейтинга
        document.querySelectorAll('input[name="rate"]').forEach(radio => {
            radio.addEventListener('change', checkSubmitBtn);
        });
```

```
        // Обработчик отправки формы
        document.getElementById('feedbackForm').addEventListener('submit',
function (event) {
            event.preventDefault();
            const rate = document.querySelector('input[name="rate"]:checked');
            const feedbackText = document.getElementById('feedbackText').value;

            document.getElementById('ratingDisplay').textContent = rate.value;
            document.getElementById('opinionDisplay').textContent = feedbackText
|| 'Мнение не указано';
            document.getElementById('tokenDisplay').textContent = token;

            document.getElementById('submittedFeedback').style.display = 'block';

            // Сброс формы после отправки
            document.getElementById('feedbackForm').reset();
        });

        // Обработчик кнопки проверки токена
        document.getElementById('verifyTokenButton').addEventListener('click',
function () {
            const secret = document.getElementById('secretInput').value;

            // В сценарии этого примера, время между получением токена и его
проверкой контролируется
            // пользователем. Поэтому, прежде чем послать токен на проверку, мы
перезапрашиваем токен у
            // виджета. Виджет автоматически обновит токен, если текущий уже
невалиден.
            if (token && secret) {
                humanometr
                    .getToken()
                    .then(t => {
                        fetch('http://localhost:50011/siteverify', {
                            method: 'POST',
                            headers: {
                                'Content-Type': 'application/json',
                                'api-version': '2.0'
                            },
                            body: JSON.stringify({
                                token: t,
                                sitekey: "10000000-ffff-ffff-ffff-000000000001",
                                secret
                            })
                        })
                            .then(response => response.json())
                            .then(data => {

document.getElementById('responseJson').textContent = JSON.stringify(data, null,
2);
                            })
                            .catch(error => {

document.getElementById('responseJson').textContent = `Ошибка: ${error}`;
```

```
                            });
                    })
                    .catch(error => {
                        handleTokenUpdate(null);
                        alert(error);
                    });


            }
        });

        // Слушатель на изменение текста секрета
        document.getElementById('secretInput').addEventListener('input',
checkVerifyButtonState);

        // Обработчик кнопки получения новой капчи
        document.getElementById('getCaptcha').addEventListener('click', function
() {
            document.getElementById('feedbackForm').reset();
            document.getElementById('submittedFeedback').style.display = 'none';
            document.getElementById('responseJson').textContent = '';
            handleTokenUpdate(null);
            humanometr.getCaptcha();
        });
    </script>

</html>
```