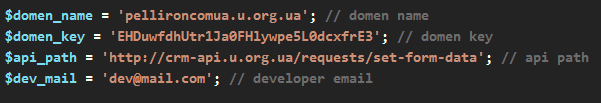
1. Заходим на страницу с формой и проверяем ее работоспособность
   1. заполняем форму тестовыми данными (в теме или имени указать контрольное слово **TEST**, email использовать действующий: свой тестовый или рабочий, чтоб не отправлять случайные запросы на чужие emails)
   2. если форма работает не корректно, записываем в отчет о проделанной работе что конкретно и в какой форме не работает. Дальнейшие действия не имеют смысла, пока форма не будет работать как полагается
2. Вносим новый домен в систему (если еще не зарегистрирован) <http://crm.u.org.ua/sofonaRequests/requests-domains>
   1. для добавления жмем «Create Requests Domains»
   2. вводим название домена (**без http://**)
   3. жмем «Save»
   4. копируем и сохраняем ключ в Excel таблицу со списком форм (или смотреть ключи на этой же странице)
3. Заходим на сайт с формами через ФТП
   1. копируем в корень сайта папку «**smetrics**» (брать свежую версию из репозитория CRM **common/modules/sofonaRequests/smetrics**)
   2. открываем файл «**smetrics/metrics.php**» и заполняем конфиг в начале файла



* 1. необходимо найти индекс файл шаблона или место, где подключается первый JS скрипт на странице и перед ним подключить (по возможности скрипт должен быть выше остальных JS скриптов, но не отдельно от них):

**<script src="/smetrics/metrics.js"></script>**

***Привемр***:



Если так не получается, то на усмотрение разработчика. Можно с кем-то посоветоваться ☺

1. Переходим на страницу добавления форм:

<http://crm.u.org.ua/sofonaRequests/requests-forms>

* 1. жмем «Create Requests Forms»
  2. выбираем Domain
  3. вводим url с формой (без домена: ***/contact****,* ***/news****,* ***/*** *- корень сайта,* ***\**** *- все страницы)*
  4. далее необходимо выбрать селектор, по которому будет определяться форма на странице. Для этого открываем инспектор кода (F12, Ctrl+U или «правой кнопкой по форме -> Исследовать элемент», …), находим форму и выбираем корректный селектор для этой формы.



- Оптимальным вариантом, если доступно, использовать ID формы.

Важно!!! Ид формы может генерироваться и в конце дописываться каждый раз новые цифры или буквы. Если есть в конце ID какие-то цифры или непонятные буквы, необходимо несколько раз перезагрузить страницу и убедиться, что ID формы не меняется, иначе выбрать другой вариант.

- Если ID формы не доступен, необходимо использовать селекторы, при этом желательно привязываться к ближайшему от формы родительскому элементу с неизменным ID или тегом (‘**#div-custom-id > p form’**, ‘**footer .test-form’**)

Список рекомендуемых селекторов:

**#id** – элемент с данным id

**div** – элементы с таким тегом

**.class** – элементы с таким классом

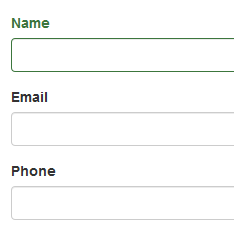
**div p** – элементы p, являющиеся потомками div

**div > p** – только непосредственные потомки

***«Псевдоклассы» (***:visited***) и «селекторы на атрибут» (***[name="value"]***) использовать не желательно, большая часть может не работать. Применять только в случае крайней необходимости и проверять работоспособность обязательно!***

-при необходимости, также можно указать номер формы. Например, если нам нужна первая форма в футоре, то в «**Selector**» мы прописываем «**footer**» – тег футера, и в «**Form** **Number**» номер формы ‘**1**’. То есть первая форма в футоре.

* 1. в инспекторе кода, если есть поля “**Name**”, “**Email**” или “**Phone**”, необходимо найти их атрибут “**name=**” и заполнить соответствующие поля в форме добавления. Иначе оставить пустыми



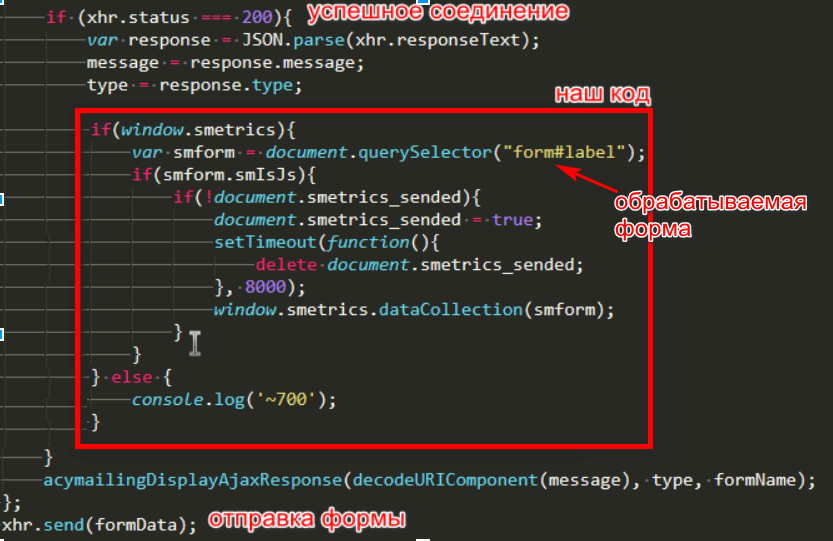
* 1. все остальные имена инпутов формы, необходимо перечислить в «Fields Names», в формате:

**inputname1:Город;inputname2:Пол; jform[contact\_subject]:Тема;jform[contact\_message]:Текст;**

* 1. Save
  2. Если появилась кнопка ошибки отправки конфига на сайт с формой, необходимо нажать повторно кнопку отправки конфига через несколько минут. Если повторно не удается отправить, сообщить ответственному разработчику или ПМу

1. Проверяем работоспособность
   1. переходим на страницу с добавляемой формой
   2. обновляем страницу **с очисткой кеша** (Ctrl + F5)
   3. заполняем форму тестовыми данными (в теме или имени указать контрольное слово **TEST**, email использовать действующий: свой тестовый или рабочий, чтоб не отправлять случайные запросы на чужие emails)
   4. отправить форму
   5. убедиться, что работоспособность формы не была нарушена
   6. проверить пришли ли данные в CRM <http://crm.u.org.ua/sofonaRequests/requests-data>
   7. записать результат работы по добавлению формы в отчет
2. Если форма не заработала:
   1. проверяем с помощью чего отправляется форма. Если с помощью JS, то на странице настройках формы в ЦРМ, необходимо установить галочку “isJs”. На сайте найти обработчик формы, выбрать место отправки формы в JSе и непосредственно перед отправкой формы, добавить следующий код:

|  |
| --- |
| if(window.smetrics){  var smform = this;  if(smform.smIsJs){  if(!document.smetrics\_sended){  document.smetrics\_sended = true;  setTimeout(function(){  delete document.smetrics\_sended;  }, 8000);  window.smetrics.dataCollection(smform);  }  }  } else {  console.log('~700');  } |

**Пример 1: (простой вариант)**

**Пример 2: асинхронная отправка (если данные не успевают отправляться)**

По умолчанию, JS не ожидает завершение выполнения предыдущих функций и если разместить наш код, непосредственно перед вызовом триггера **submit()**, то наш код не успеет отработать до перезагрузки страницы. В таком случае необходимо использовать второй, ожидающий завершения отправки данных в ЦРМ.

В данном случае, в JS коде, форма отправляется с помощью “**head\_form.submit();”**. Тогда необходимо данный вызов триггера сабмита обернуть нашим кодом:

|  |
| --- |
| if(window.smetrics){  var smform = this;  if(smform.smIsJs){  if(!document.smetrics\_sended){  document.smetrics\_sended = true;  setTimeout(function(){  delete document.smetrics\_sended;  }, 8000);  window.smetrics.dataCollection(smform, function(metrics){  **head\_form.submit();**  });  }  }  else{  **head\_form.submit();**  }  } else {  console.log('~700');  **head\_form.submit();**  } |

**Пример 3: отложенная загрузка (самый плохой вариант).**

Например если document еще не ready, а форму уже можно отправить. Или любая другая отложенная отправка…

|  |
| --- |
| // функция для отложенной отправки, если большая задержка перед  function sendMetrics(m, callback)  {  var **maxResend** = **18**;  var **timeResend** = **1000**;  var mSend = function(item){  if(window.smetrics && item.smcache){  if(item.smIsJs){  if(!document.smetrics\_sended){  document.smetrics\_sended = true;  setTimeout(function(){  delete document.smetrics\_sended;  }, 8000);  window.smetrics.dataCollection(smform);  }  }  return true;  }  return false;  };    var nResend = 0;  var mInterval = setInterval(function() {  if( mSend(m) ){  callback();  clearInterval(mInterval);  }  else{  if(nResend > maxResend){  console.log('~700');  clearInterval(mInterval);  callback();  }  nResend++;  }  }, timeResend);  }  // это вставить там, где отправляется форма  $('#cformtop').on("submit", function()  {  //console.log('send: cformtop');  sendMetrics(this, function(){  //------ **default code here**  });  }); |

**Пример 4: если надо получить результат отправки**

Функция getResult() вызывается по завершению отправки и может содержать такие ответы как:

'done' - успешно принято в ЦРМ

'failed' (то-же что и '~995') - ошибка при отправке или получении, данные поставлены в очередь на отправку. Для получения сохраненных заявок, на стороне црм необходимо выполнять [http://crmApiURL/requests/get-form-data](http://crmapiurl/requests/get-form-data)

В ответе также может содержаться другой текст! По этому для проверке в JSe лушче использовать .indexOf()

|  |
| --- |
| if(window.smetrics){  var smform = this;  if(smform.smIsJs){  if(!document.smetrics\_sended){  document.smetrics\_sended = true;  setTimeout(function(){  delete document.smetrics\_sended;  }, 8000);  window.smetrics.dataCollection(smform, getResult);  }  }  } else {  console.log('~700');  }  function getResult(e){  // e.smAnswer возвращает строку и может иметь два статуса: 'done', 'failed', так-же может содержать текст ошибки PHP  if(e.smAnswer.indexOf('done') + 1)){  console.log('Done!');  } else if(e.smAnswer.indexOf('failed') + 1 || e.smAnswer.indexOf('~995') + 1) {  console.log('Saved and will be sent later');  } else {  console.log('Error!');  }  } |

**Пример 5: WP (avia-footer-scripts)**

// подключение в WP

// site.domen/wp-content/themes/enfold/functions.php

if( is\_page('vakansija') || is\_page('zarabotok') || is\_page('obuchenie') ){  
 wp\_enqueue\_script( 'smetrics', '/smetrics/metrics.min.js', array(), $vn, true);  
// wp\_add\_inline\_script( 'smetrics', 'console.log("test3")', 'after' );  
 }

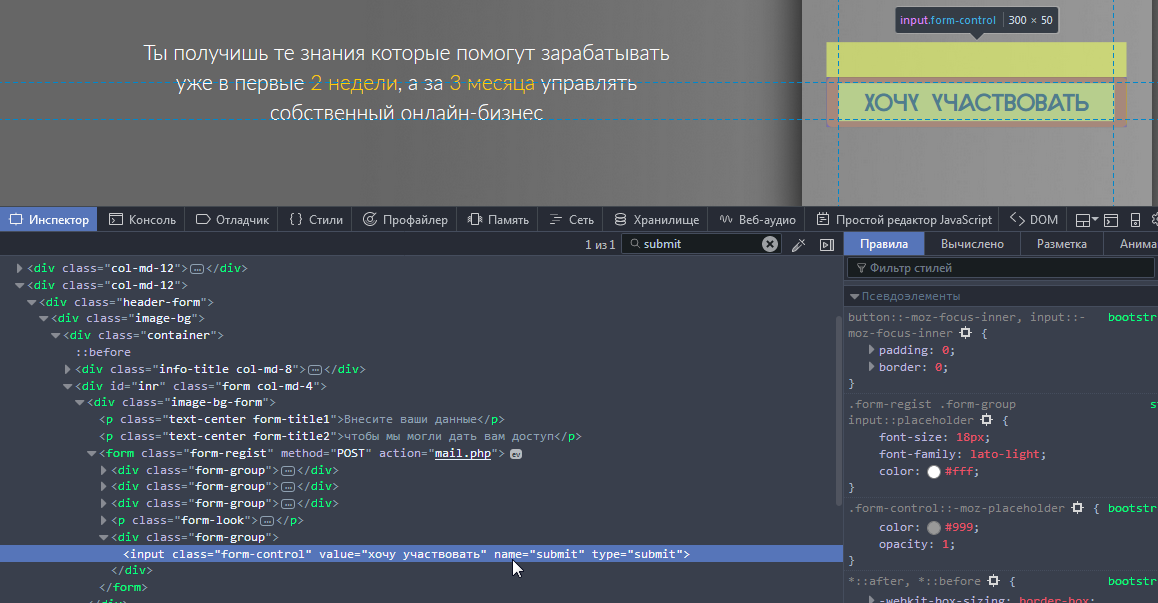
// приемлемо рабочий вариант (достаточно низкая степень коллизий)

jQuery('document').ready(function($){  
 $('form.avia\_ajax\_form input.button').click(function(e){  
 this.form.smSending = true;  
 });  
});  
  
jQuery( document ).ajaxError(function() {  
 var sForm = jQuery('form.avia\_ajax\_form')[0];  
 if(sForm && sForm.smSending){  
 sForm.smSending = false;  
 }  
});  
  
jQuery( document ).ajaxSuccess(function(e, r, s) {  
 if(window.smetrics){  
 var smform = jQuery('form.avia\_ajax\_form')[0];  
 if(smform.smSending){  
 smform.smSending = false;  
 if(!document.smetrics\_sended){  
 document.smetrics\_sended = true;  
 setTimeout(function(){  
 delete document.smetrics\_sended;  
 }, 8000);  
 window.smetrics.dataCollection(smform);  
 }  
 }  
 } else {  
 console.log('~700');  
 }  
});

min:

jQuery("document").ready(function(e){e("form.avia\_ajax\_form input.button").click(function(e){this.form.smSending=!0})}),jQuery(document).ajaxError(function(){var e=jQuery("form.avia\_ajax\_form")[0];e&&e.smSending&&(e.smSending=!1)}),jQuery(document).ajaxSuccess(function(e,n,o){if(window.smetrics){var i=jQuery("form.avia\_ajax\_form")[0];i.smSending&&(i.smSending=!1,document.smetrics\_sended||(document.smetrics\_sended=!0,setTimeout(function(){delete document.smetrics\_sended},8e3),window.smetrics.dataCollection(i)))}else console.log("~700")});

Возможные ошибки:

1. “… .submit is not a function”

Решение: необходимо найти и переименовать кнопку с name=”submit”, например в name=”btnsubmit”