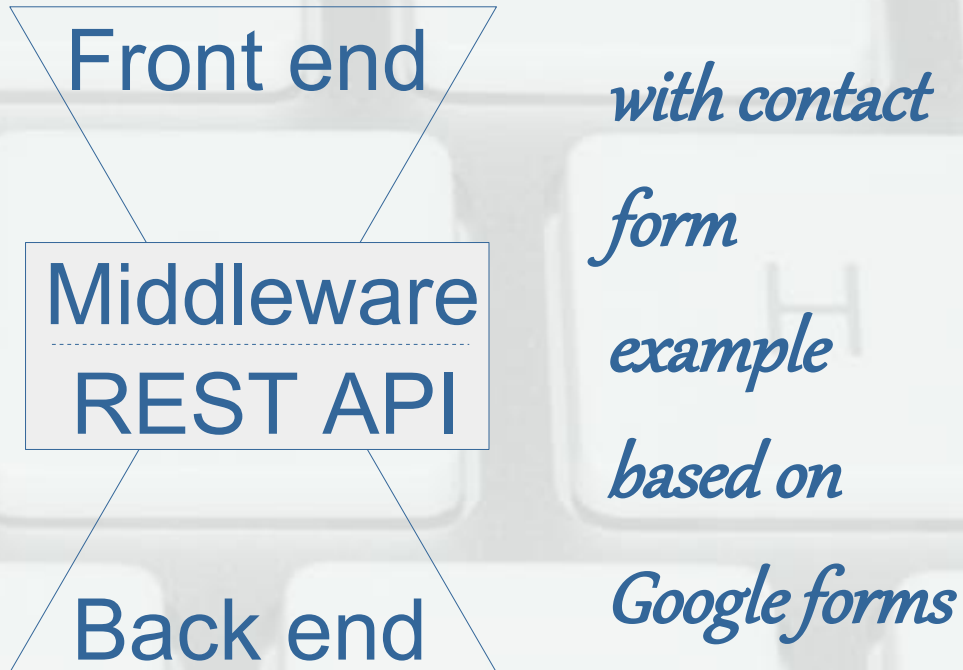


# W2BUSINESS QA ACADEMY

Open explaining lesson:



# LECTORS

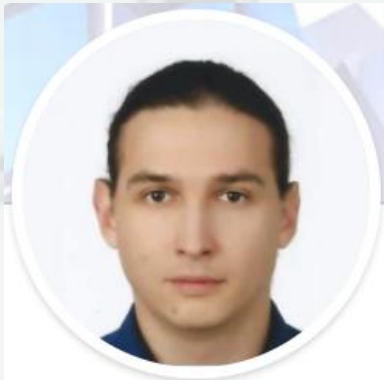
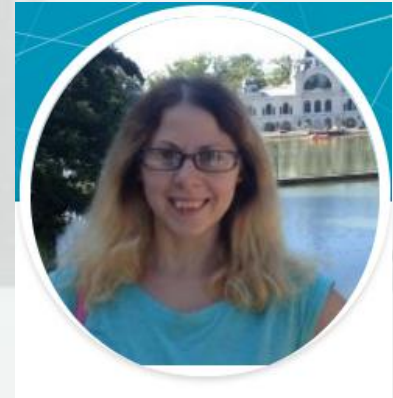
## Svitlana Samko

<https://www.linkedin.com/in/svitlana-samko/>

Quality Assurance Technical Lead, Senior Developer in Test, over 10 years web development practice, over 10 delivered software projects for middle and large business. Professional Scrum Master, Requirements Engineer, Management Master, Digital Product Manager Master, Full-stack developer BSc in Computer Science

*My most beneficial skill: I like to learn business from the inside.*

*Only so one can be sure that we build right product in the right way at any stage of development process.*



## Andrii Stepura

<https://www.linkedin.com/in/andriistepura/>

Quality Assurance Technical Lead, Senior QA Automation Engineer, Product Owner, Professional Scrum Master, Full-stack developer BSc in Computer Science. Over 14 years web development practice, over 300 delivered web projects.

*My most beneficial skill: Imagination to think like a stakeholder.*

*Every piece of software starts from an idea. The first written code lines are just a half of the delivery of that idea.*

# GOAL DEFINITION

Create website contact form with implementation at three layers for familiarize and clarify web-development terms:

- Front end
- Back end
- Middle-ware
- REST API

# OUR WAY TO GOAL

Less place stereotypes, more for self abstract definition  
with web-development practice activity



Graphics from [bluecoders.io](http://bluecoders.io) & [theodawson.net](http://theodawson.net)

# REQUIRED TOOLS

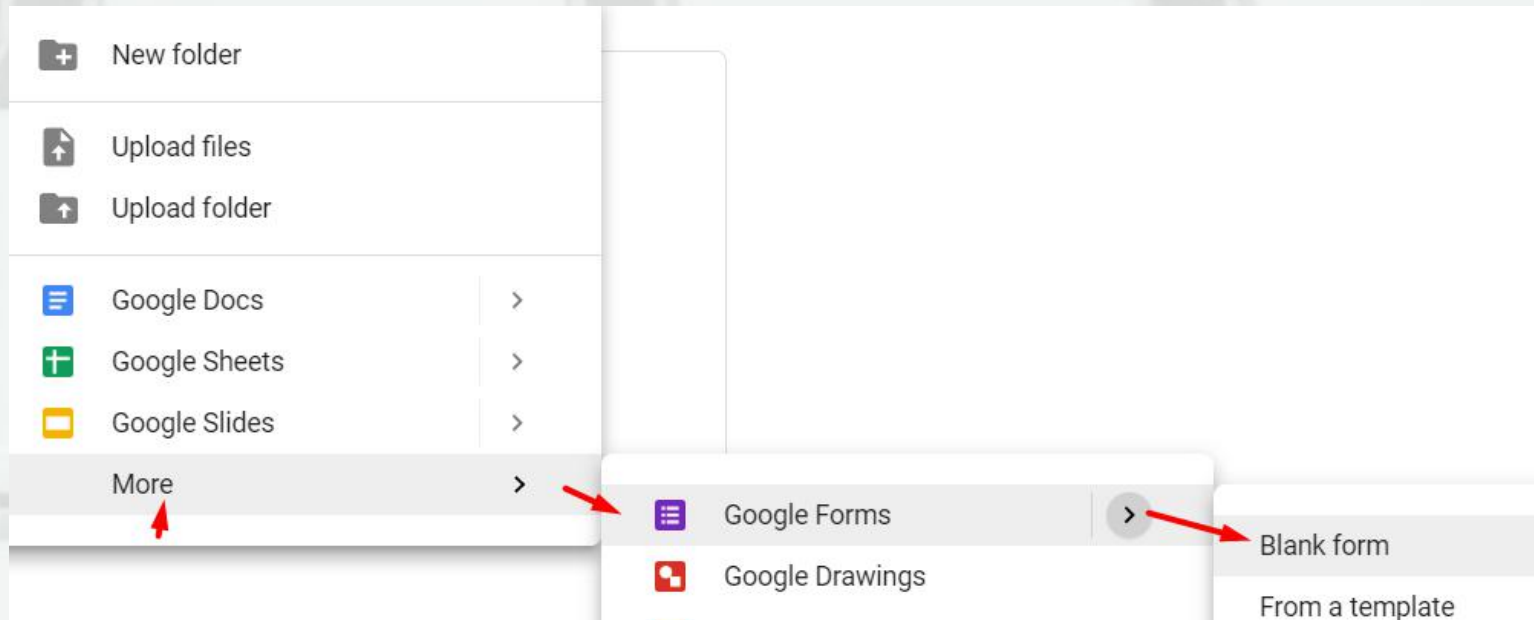
For creation final solution and repeat all by yourself you will need at least next tools:

- **Google account** (<https://accounts.google.com/> )
- **Web browser** (for demo used Chrome - <https://www.google.com/chrome/> )
- **Code editor** (as ex. Notepad++ - <https://notepad-plus-plus.org/download/> )
- **REST API** (REST call tool Postman - <https://www.getpostman.com/downloads/> )
- **PHP server** (hosting or local server, as ex. XAMPP - <https://www.apachefriends.org/> )



# CREATE GOOGLE FORM

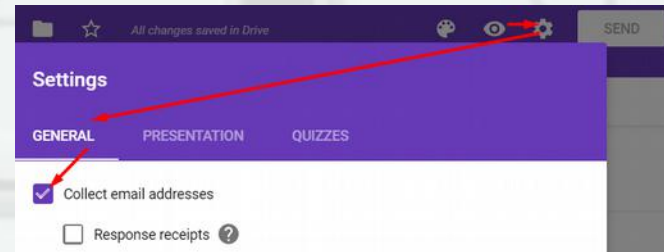
Login at <https://drive.google.com/> with your Google Account and create form:



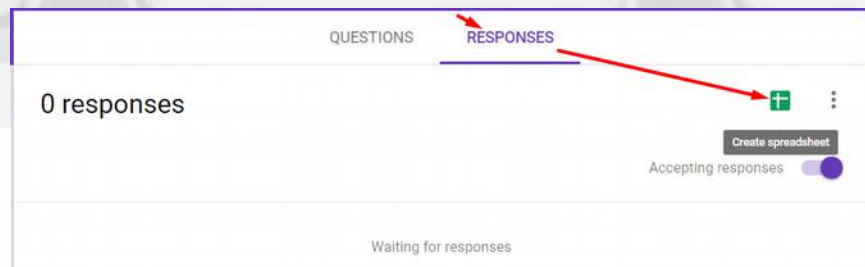
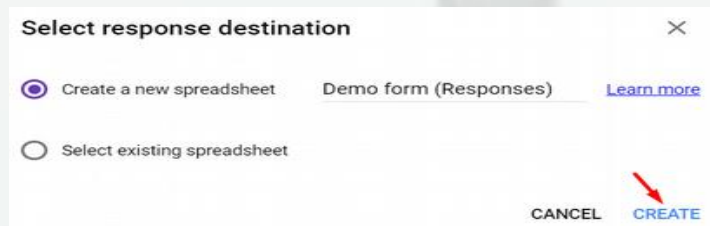
# CREATE GOOGLE FORM

Add fields need for receive data from our website visitors, as ex.:

- Name (short answer text)
- E-mail (use form settings)
- Message (short answer text)

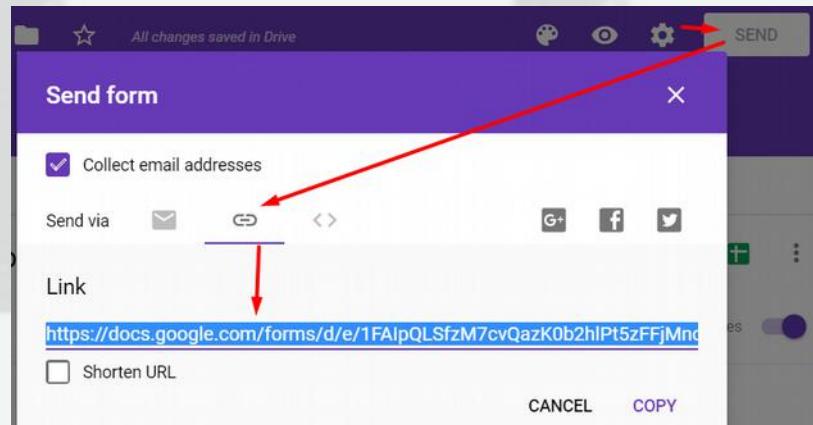


Set store answers in Google spreadsheet, its table like MS Excel, we will use it as Database for storing:



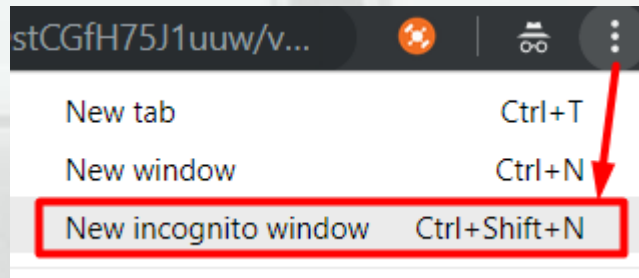
# EXPLORE GOOGLE FORM

Get this form link:



[https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/viewform?usp=sf\\_link](https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/viewform?usp=sf_link)

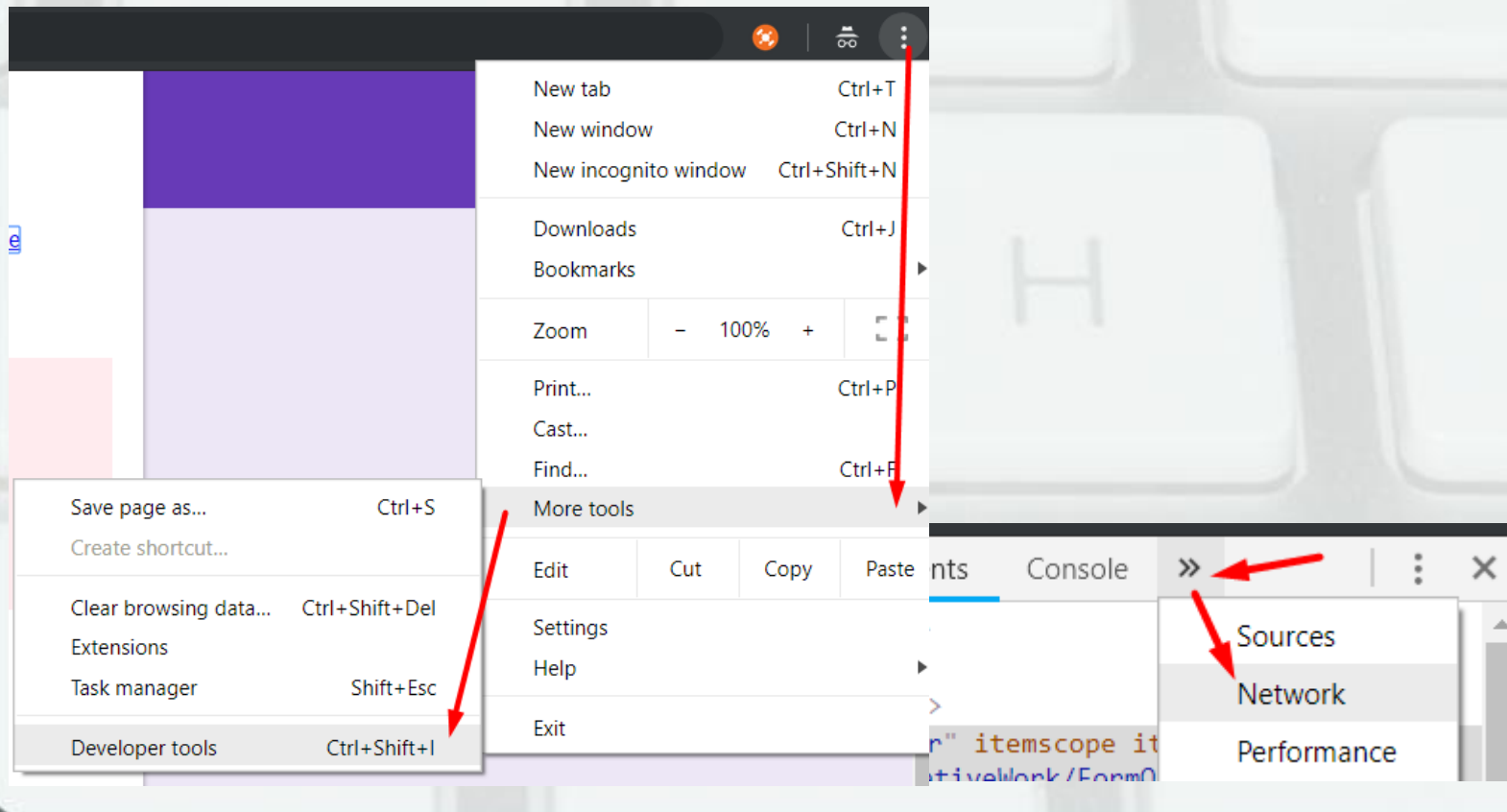
and open as other user (use incognito mode):





# EXPLORE GOOGLE FORM

Open developer tools console (Ctrl+Shift+I)  
and switch to Network:



# EXPLORE GOOGLE FORM

Fill form and click on Submit button:

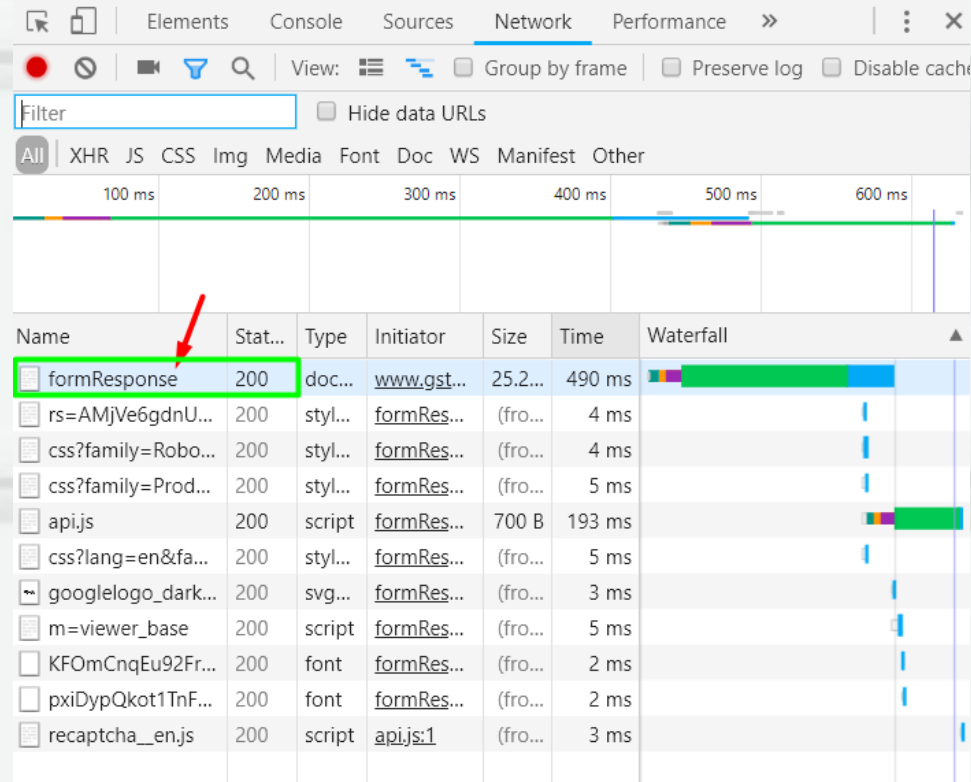
The screenshot shows a Google Form titled "Demo form" with the following fields:

- Email address \***: test@w2business.pl
- Name**: Test form name
- Your message**: Test form message

A blue **SUBMIT** button is at the bottom left. A red arrow points from the SUBMIT button to the Network tab in the Chrome DevTools sidebar on the right. The Network tab shows a table with columns: Filter, All, XHR, JS, CSS, Img, Media, Font, Doc, WS, Manifest, C. Below the table, it says "Recording network activity..." and "Perform a request or hit **Ctrl + R** to record the reload."

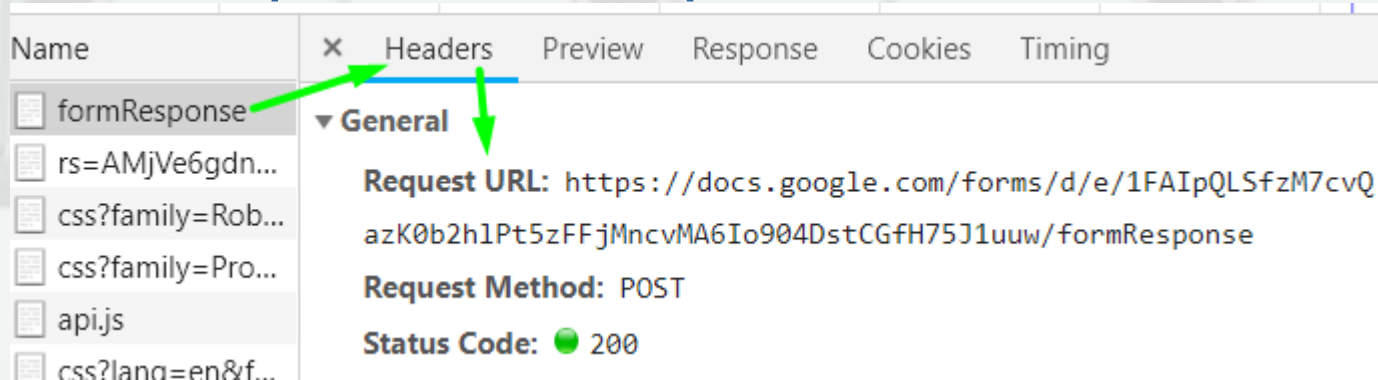
# EXPLORE GOOGLE FORM

In network communication record with this post to Google form API is appeared:



# EXPLORE POST REST REQUEST

Let's explore this request, headers:



Request URL:

<https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2h1Pt5zFFjMncvMA6Io904DstCGfH75J1uuw/formResponse>

Request Method: **POST**

Status Code: **200**

Where request URL – is Google API endpoint for our form

Method is HTTP call type, more about REST -

[https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol#Request\\_methods](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods)

Status is result from endpoint, more about status codes -

[https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

# EXPLORE POST REST REQUEST

With POST HTTP method we send not just headers (instead GET), we also send body:

▼ Form Data    view source    view URL encoded

**emailAddress:** test@w2business.pl  
**entry.1224778685:** Test form name  
**entry.512804397:** Test form message

**fvv:** 1

**draftResponse:** [null,null,"-9115861482995980081"]

**pageHistory:** 0

**fbzx:** -9115861482995980081

emailAddress: test@w2business.pl  
entry.1224778685: Test form name  
entry.512804397: Test form message

Same data saved in our spreadsheet:

Email address	Name	Your message
test@w2business.pl	Test form name	Test form message

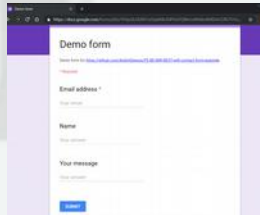


# SOLUTION ARCHITECTURE

Let's analyze our current solution layers:

Front  
end

HTML5  
GOOGLE  
FORM



HTML form received by browser with GET request to endpoint -

<https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/viewform>

Request Response

Back  
end

GOOGLE  
FORM  
REST API

Google API



REST API endpoint receive POST request, validate and save to DB

<https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/formResponse>

DB

spreadsheet



B	C	D
Email address	Name	Your message
test@w2business.pl	Test form name	Test form message

# HTTP RESTS WITH POSTMAN

Use Postman tool for send POST HTTP request instead browser client:

The screenshot shows the Postman interface for a POST request. The request is named "Post to Google Form". The URL is `https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6Io904DstCGfH75J1uuw/formResponse`. The "Body" tab is selected, and the "form-data" radio button is chosen. The body contains three form fields:

KEY	VALUE	DESCRIPTION
emailAddress	test@w2business.pl from Postman	
entry.1224778685	Test form name from Postman	
entry.512804397	Test form message from Postman	

A green arrow points to the "Send" button.

Form-data:

emailAddress:test@w2business.pl from Postman

entry.1224778685:Test form name from Postman

entry.512804397:Test form message from Postman

# HELLO ERROR “400 Bad Request”


Our request data not satisfied for REST API endpoint validation rules, so we receive response code 400 instead 200

Status: 400 Bad Request Time: 327 ms Size: 101 KB

Check DB – any changes, just 1 record try again with proper form-data emailAddress: emailAddress:test\_from\_postman@w2business.pl

Status: 200 OK Time: 522 ms Size: 88.03 KB

and form data saved in DB (spreadsheet):

Email address	Name	Your message	
test@w2business.pl	Test form name	Test form message	
test_from_postman@w2b	Test form name from Post	Test form message from Postman	

# SOLUTION ARCHITECTURE

Let's analyze our current solution layers:

Front  
end

REST  
CLIENT



Back  
end

GOOGLE  
FORM  
REST API

Google API



REST API endpoint  
receive POST request,  
validate and save to DB

<https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/formResponse>

DB

spreadsheet



B	C	D
Email address	Name	Your message
test@w2business.pl	Test form name	Test form message

# EXPLORE GOOGLE FORM HTML

Return to Google form and inspect HTML code with Elements tab:

The screenshot shows a Google Form titled "Demo form" with the following content:

Demo form for <https://github.com/AndriiStepura/FE-BE-MW-REST-with-contact-form-example>

\* Required

Email address \*

Your email

Name

Your answer

The browser's developer tools are open, showing the HTML source code. The 'Elements' tab is selected. The HTML structure is as follows:

```
<meta itemprop="image" content="https://ssl.gstatic.com/docs/forms/social-social-forms-big-2.png">
<meta itemprop="imageUrl" content="https://ssl.gstatic.com/docs/forms/social/social-forms-big-2.png">
<div class="freebirdFormviewerViewFormContentWrapper">
  <div class="freebirdFormviewerViewFormBanner freebirdHeaderMast"></div>
  <div class="freebirdFormviewerViewCenteredContent">
    <form action="https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2h1Pt5zFFjMncvMA6Io904DstCGfH75J1u.../formResponse" target="_self" method="POST" id="mG61Hd">
      ...
    </form>
  </div>
</div>
```

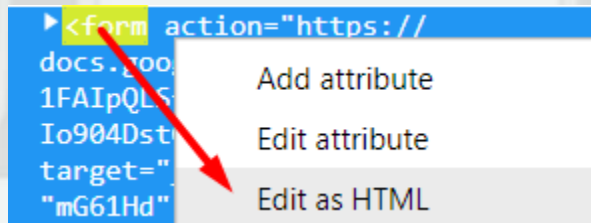
Green arrows indicate the mapping between the form fields and the HTML elements:

- Arrow 1 points to the top of the form container.
- Arrow 2 points to the 'Email address' field.
- Arrow 3 points to the 'Name' field.



## CREATE OUR HTML FORM

Copy all form HTML code to code editor:



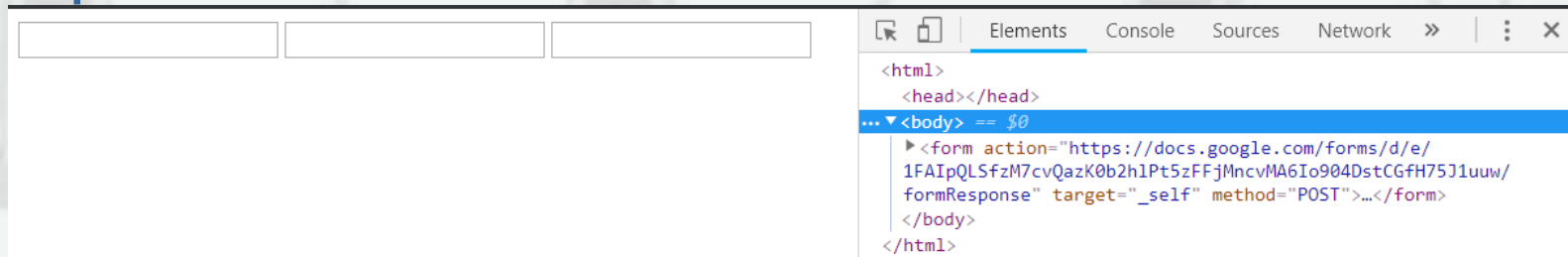
```
1 <form action=
  "https://docs.google.com/form
  Response" target="_self" me
  exportFormCard"><div class
  ></div><div class="freebird
```

After remove classes, additional div and attributes we receive short form HTML code:

```
<form
action="https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQaz
K0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/formResponse"
target="_self" method="POST">
    <input type="email" name="emailAddress">
    <input type="text" name="entry.1224778685">
    <input type="text" name="entry.512804397">
</form>
```

# CREATE OUR HTML FORM

Open new created HTML form in browser:



it's almost worked we need just add a button for submit this form, so let's add:

```
<form
action="https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b
2h1Pt5zFFjMncvMA6Io904DstCGfH75J1uuw/formResponse"
target="_self" method="POST">
    <input type="email" name="emailAddress">
    <input type="text" name="entry.1224778685">
    <input type="text" name="entry.512804397">
    <input type="submit" name="submit" value="Submit">
</form>
```

# CREATE OUR HTML FORM

Let's try submit correct data to OUR HTML form and check data saved in DB (spreadsheet):

The screenshot shows a web browser with a 'Demo form' on the left and the Chrome DevTools 'Network' tab on the right. The form has a message 'Your response has been recorded.' and a link 'Submit another response'. The 'Network' tab shows a 'form:response' entry. The 'Form Data' section of the response is highlighted with a green box, showing the following data:

entry	Test form name from Our HTML	Test form message from Our HTML
entry:1224778685	Test form name from Our HTML	Test form message from Our HTML
entry:512804397	Test form name from Our HTML	Test form message from Our HTML

Email address	Name	Your message
test@w2business.pl	Test form name	Test form message
test_from_postman@w2b	Test form name from Post	Test form message from Postman
test_from_our_html@w2b	Test form name from Our	Test form message from Our HTML

It's work (with redirect to Google forms, but work). Data from our HTML form, local hosted page or hosted online at our website now send data to Google forms

# SOLUTION ARCHITECTURE

Let's analyze our current solution layers:

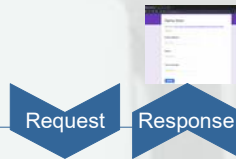
Front  
end

OUR  
WEBSITE



Our HTML form POST data and redirected to Google form page

GOOGLE  
FORM



Back  
end

GOOGLE  
FORM  
REST API

Google API



REST API endpoint receive POST request, validate and save to DB

<https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGfH75J1uuw/formResponse>

DB

spreadsheet



B	C	D
Email address	Name	Your message
test@w2business.pl	Test form name	Test form message

# OUR Front end

Let's make form more pretty with styled by CSS and HTML5 tags:

```
<link type="text/css" rel="stylesheet" href="style.css" />
```

```
<form action="https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6Io904DstCGfH75J1uww/formResponse" target="_self" method="POST">
```

```
<input type="email" name="emailAddress" placeholder="Enter your E-mail">
```

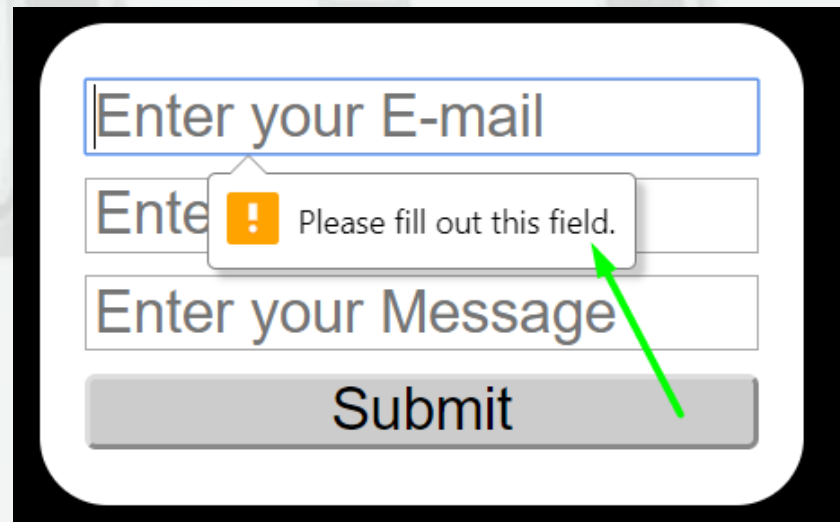
```
<input type="text" name="entry.1224778685" placeholder="Enter your Name">
```

```
<input type="text" name="entry.512804397" placeholder="Enter your Message">
```

```
<input type="submit" name="submit" value="Submit">
</form>
```

style.css file:

```
body {background: #000;}
form {background: #fff; display:block;
width:300px; -webkit-border-radius: 30px;
border-radius:30px; padding:20px; margin-left:
auto; margin-right: auto;}
input {width:100%; font-size:26px; padding: 0
2px; margin: 5px 0;}
input[type=submit] {background:#ccc;
cursor:pointer; -webkit-border-radius: 5px;
border-radius: 5px;}
```





# OUR Front end

Looks better but form still redirect us to Google form page. But we know about HTTP method POST and can implement this call (same as in Postman) at our Front end side with JavaScript.

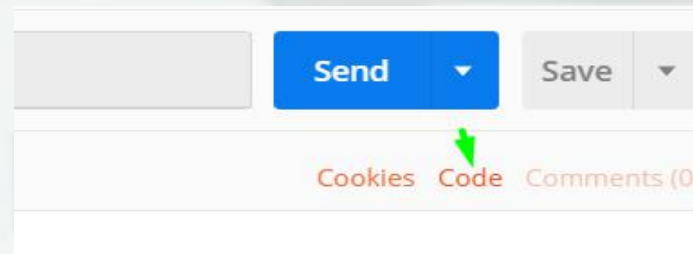
1. Create file form.js and add it to HTML page:

```
<script type="text/javascript" src="form.js"></script>
```

2. Update Submit button with call post function:

```
<input type="button" name="submit" value="Submit" onclick="sendForm()">
```

Get code generated by Postman and use as function in our form.js



# OUR Front end

## Update form.js file code:

```
function sendForm(){
    var data = new FormData();
    data.append("emailAddress", "test_from_FE_JS1@w2business.pl");
    data.append("entry.1224778685", "Test form name from JS");
    data.append("entry.512804397", "Test form message from JS");
    var xhr = new XMLHttpRequest();
    xhr.withCredentials = true;
    xhr.addEventListener("readystatechange", function () {
        if (this.readyState === 4) {console.log(this.responseText);});
    xhr.open("POST",
"https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFj
MncvMA6lo904DstCGfH75J1uuw/formResponse");
    xhr.send(data);
}
```

# OUR Front end

Click on Submit button:



data from JS file saved to DB:

B	C	D	E
Email address	Name	Your message	
test@w2business.pl	Test form name	Test form message	
test_from_postman@w2b	Test form name from Post	Test form message from Postman	
test_from_our_html@w2b	Test form name from Our	Test form message from Our HTML	
test_from_FE_JS1@w2b	Test form name from JS	Test form message from JS	

# OUR Front end

## Update form.js file with validation:

```
document.getElementById('formErrors').innerHTML= "";
var errors = [];
var emailInForm = document.getElementsByName("emailAddress")[0].value;
var nameInForm = document.getElementsByName("entry.1224778685")[0].value;
var messageInForm = document.getElementsByName("entry.512804397")[0].value;

if(emailInForm == ""){errors.push("Fill field Email");}
else{
  if(!validateEmail(emailInForm)){errors.push("Please enter valid Email");}
}

if(nameInForm == ""){errors.push("Fill field Name");}

if(messageInForm == ""){errors.push("Fill field Message");}
```

# OUR Front end

Update form.js with email validation function:

```
function validateEmail(email) {  
    var re = /^[^<>()\[\]\\\.,;:\s@"]+(\.[^<>()\[\]\\\.,;:\s@"]+)*|(".+")@(\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\]|([a-zA-Z-0-9]+\.)+[a-zA-Z]{2,}))$/;  
    return re.test(String(email).toLowerCase());  
}
```

Update style.css for errors:

```
#formErrors div{  
    background: #d68585;  
    margin: 5px 0;  
    text-align: center;  
    border-radius: 10px;  
    color: #fff;  
}
```



# OUR Front end

Update form.js to show errors or send data:

```
if (errors.length > 0){
  errors.forEach(function(element) {
    console.log("Show error - " + element);
    let errorBlock = document.createElement("div");
    errorBlock.appendChild(document.createTextNode(element));
    document.getElementById('formErrors').appendChild(errorBlock);
  });
}
else{//form validation end
  var data = new FormData();
  data.append("emailAddress", emailInForm);
  data.append("entry.1224778685", nameInForm);
  data.append("entry.512804397", messageInForm);

  ... etc var xhr = new XMLHttpRequest(); ...
```

# OUR Front end

Let's test negative and positive data sets:

Please enter valid Email

Fill field Name

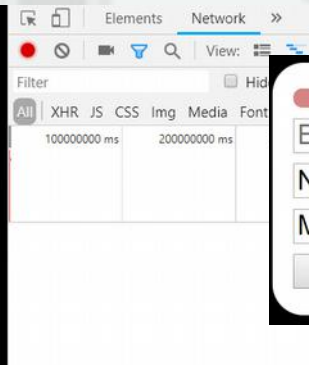
Fill field Message

not email

Enter your Name

Enter your Message

Submit



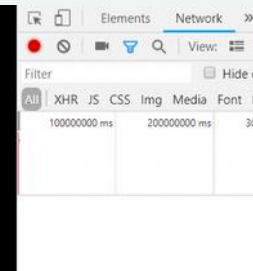
Fill field Email

Enter your E-mail

Name in HTML form

Message from HTML form

Submit

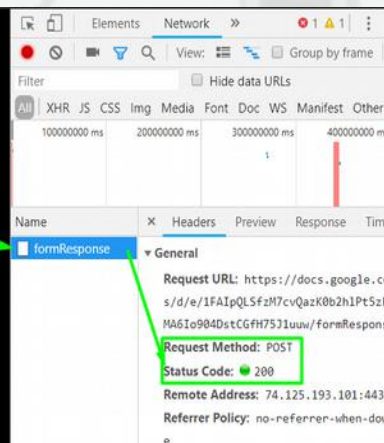


test\_from\_our\_FE\_JS@w

Name in HTML form

Message from HTML form

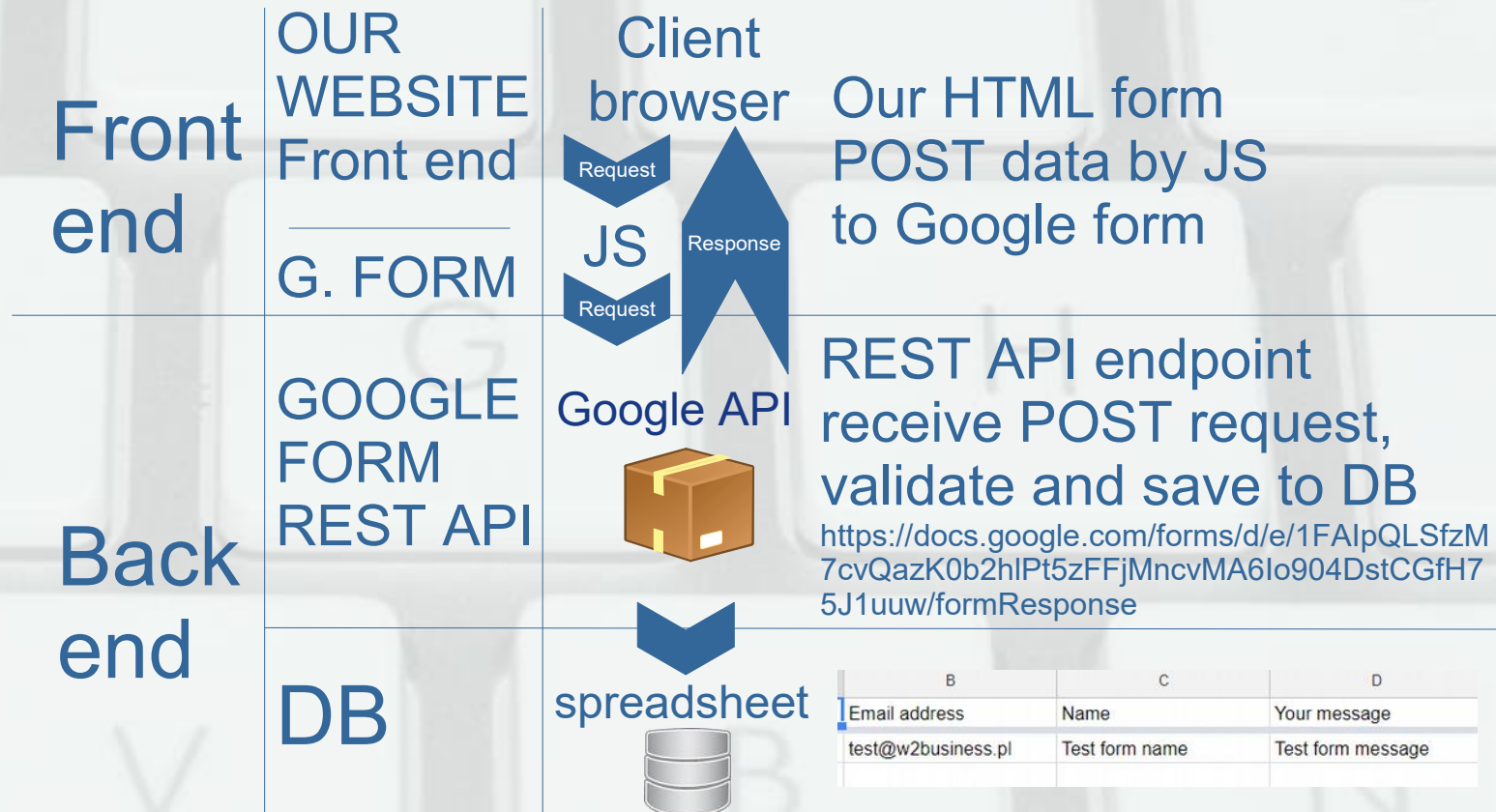
Submit



Email address	Name	Your message
test@w2business.pl	Test form name	Test form message
test_from_postman@w2b	Test form name from Post	Test form message from Postman
test_from_our_html@w2b	Test form name from Our	Test form message from Our HTML
test_from_FE_JS1@w2b	Test form name from JS	Test form message from JS
test_from_our_FE_JS@w	Name in HTML form	Message from HTML form by JS DB

# SOLUTION ARCHITECTURE

Let's analyze our current solution layers:



# CORS?

✖ Access to XMLHttpRequest at '<https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2h1Pt5zFFjMncvMA6Io904DstCGfH75J1uuw/formResponse>' from origin 'null' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

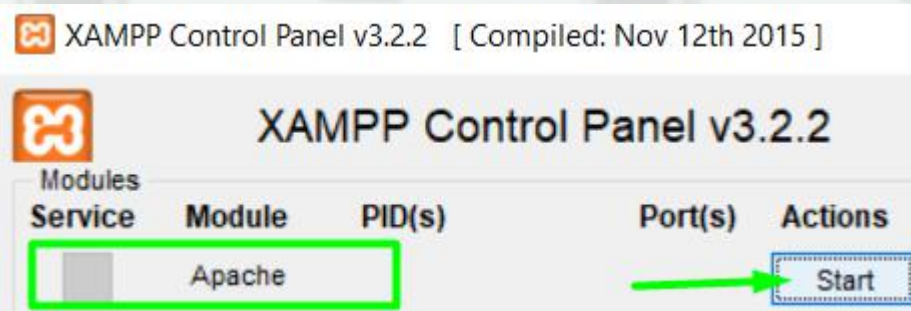
[form.js:45](#)

⚠ ▶ Cross-Origin Read Blocking (CORB) [form.js:50](#)  
blocked cross-origin response <https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2h1Pt5zFFjMncvMA6Io904DstCGfH75J1uuw/formResponse> with MIME type text/html. See <https://www.chromestatus.com/feature/5629709824032768> for more details.

<https://stackoverflow.com/questions/20035101/why-does-my-javascript-get-a-no-access-control-allow-origin-header-is-present>

# OUR Back end

Use web hosting or install XAMPP and start Apache server



According to HOW-TO Guides -

<http://localhost/dashboard/docs/configure-vhosts.html> add virtual hosted domain. Add in file `\apache\conf\extra\httpd-vhosts.conf` :

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/"
    ServerName localhost
</VirtualHost>
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/demo/htdocs"
    ServerName demo.localhost
</VirtualHost>
```

... restart Apache and open demo.localhost in browser



# OUR Back end

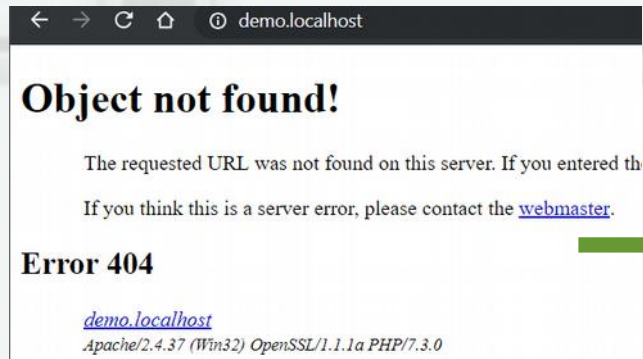
At your website create file index.php

```
<?php  
echo "PHP here, version: ".phpversion();  
?>
```

```
<br>  
HTML here
```

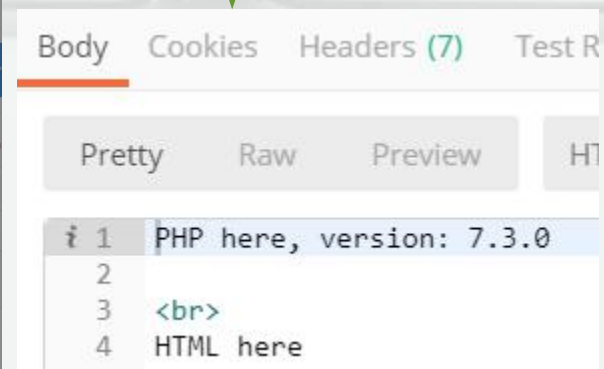
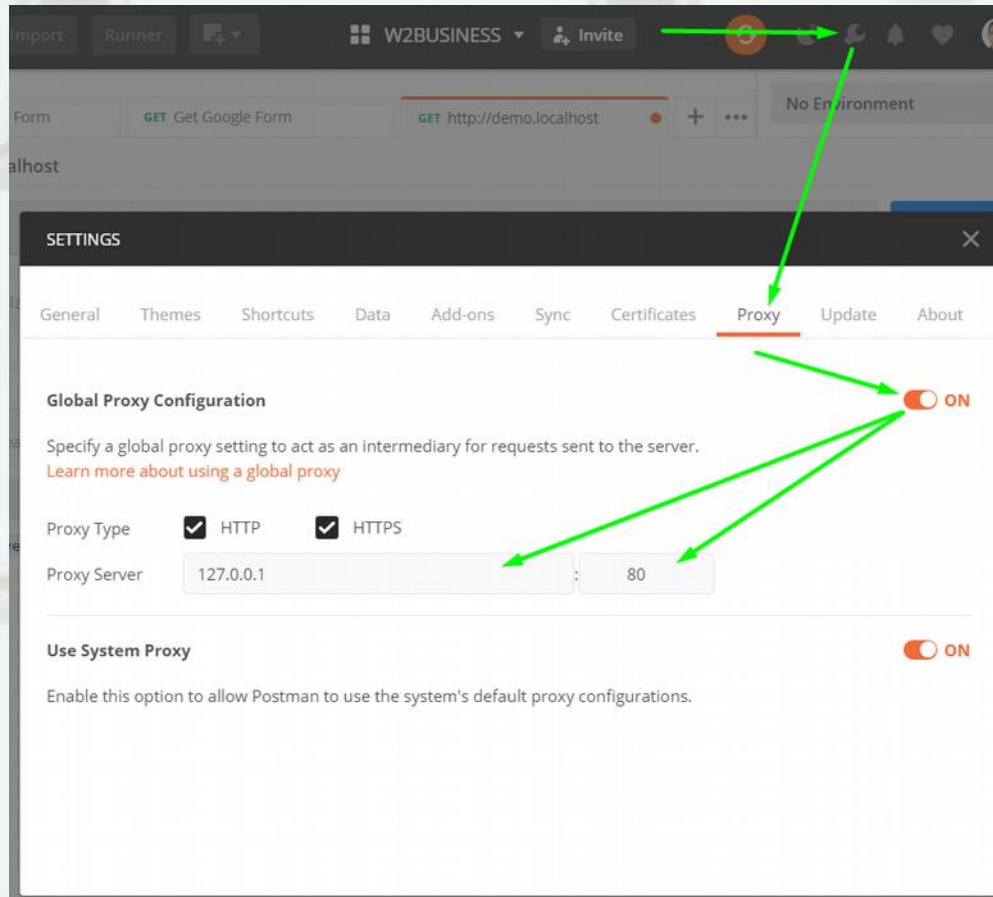
Note for XAMPP – create folders \demo\htdocs in \xampp\htdocs and put this file inside.

Refresh website in browser:



# OUR Back end

Change Postman settings and call GET demo.localhost

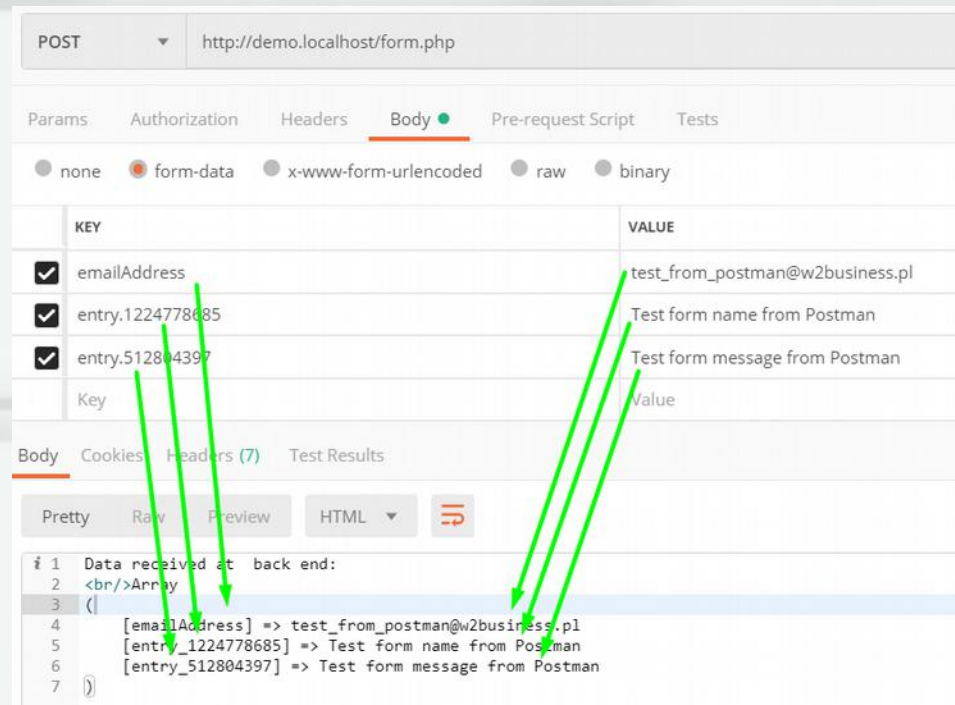


# OUR Back end

At your website create file form.php

```
<?php
echo "Data received at back end:."<br/>";
print_r($_POST);
?>
```

Call this form from browser  
(GET request) and from  
Postman as POST and  
change Google form  
endpoint url to  
demo.localhost/form.php :



# OUR Back end

## Lets update form.php

```
<?php
if($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['postEmail']) &&
isset($_POST['postName']) &&
isset($_POST['postMessage'])){

    $email = $_POST['postEmail'];
    $name = $_POST['postName'];
    $message = $_POST['postMessage'];

    if ($name == "" || $email == "" || $message == ""){
        http_response_code(400);
        echo "Fill all required fields!";
    }
    else{ // All form data looks good, so:
        http_response_code(200);
        echo "Data valid";
    }
}
else{
    http_response_code(406);
    echo "Provide all required data: postEmail,
postName, postMessage";
}
?>
```

## Call updated BE form.php:

POST http://demo.localhost/form.php

Params Authorization Headers **Body** Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE
<input checked="" type="checkbox"/> emailAdre...	test_from_postman@w2business.pl
<input checked="" type="checkbox"/> entry.1224...	Test form name from Postman
<input checked="" type="checkbox"/> entry.5128...	Test form message from Postman
Key	Value

Body Cookies Headers (7) Test Results Status: **406 Not Acceptable**

Pretty Raw Preview HTML

i 1 Provide all required data: postEmail, postName, postMessage

## Change postman POST requests keys and call again:

☒ postEmail

☒ postName

☒ postMessage

Key Value

Body Cookies Headers (6) Test Results Status: **400 Bad Request**

Pretty Raw Preview HTML

i 1 Fill all required fields!

# OUR Back end

## Lets update form.php

```
<?php
if($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['postEmail']) &&
isset($_POST['postName']) &&
isset($_POST['postMessage'])){

    $email = $_POST['postEmail'];
    $name = $_POST['postName'];
    $message = $_POST['postMessage'];

    if ($name == "" || $email == "" || $message == ""){
        http_response_code(400);
        echo "Fill all required fields!";
    }
    else{ // All form data looks good, so:
        http_response_code(200);
        echo "Data valid";
    }
}
else{
    http_response_code(406);
    echo "Provide all required data: postEmail,
postName, postMessage";
}
?>
```

Call updated BE form.php with proper data:

POST http://demo.localhost/form.php

KEY	VALUE
<input checked="" type="checkbox"/> postEmail	1
<input checked="" type="checkbox"/> postName	2
<input checked="" type="checkbox"/> postMessa...	3
Key	Value

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview HTML

i 1 Data valid



# OUR Back end

## Lets increase BE validation in form.php

```
<?php
if($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['postEmail']) &&
isset($_POST['postName']) &&
isset($_POST['postMessage'])){

    $email = strtolower($_POST['postEmail']);
    $name = $_POST['postName'];
    $message = $_POST['postMessage'];

    if ($name == "" || $email == "" || $message == ""){
        http_response_code(400);
        echo "Fill all required fields!";
    }
    else{ // All form data looks good, so:
        http_response_code(200);
        echo "Data valid";
    }
}
else{
    http_response_code(406);
    echo "Provide all required data: postEmail,
postName, postMessage";
}
?>
```

Add errors for invalid cases:

```
elseif (strlen($name) < 3) {
    http_response_code(400);
    echo "Name must be not less than 2 characters";
}
elseif (strlen($name) > 25) {
    http_response_code(400);
    echo "Name must be less than 26 characters";
}
elseif(!preg_match("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.?[a-z0-9-]{2,20})$", $email)){
    http_response_code(400);
    echo "Enter the correct e-mail";
}
elseif (strlen($message) < 5) {
    http_response_code(400);
    echo "Message must be not less than 5 characters";
}
elseif (strlen($message) > 300) {
    http_response_code(400);
    echo "Message must be less than 300 characters";
}
```

# OUR Back end

If data proper according to our back end validation let's send this to Google form

Replace code: With request which send data to Google API:

```
// All form data looks  
good, so:  
http_response_code  
(200);  
echo "Data valid";
```

```
// All form data looks good, so lets start curl to Google form:  
$url =  
'https://docs.google.com/forms/d/e/1FAIpQLSfzM7cvQazK0b2hIPt5zFFjMncvMA6lo904DstCGf  
H75J1uuw/formResponse';  
// Fix for cyrillic variables with urlencode:  
$myvars = 'emailAddress=' . urlencode($email) . '&entry.1224778685=' .  
urlencode($name) . '&entry.512804397=' . urlencode($message);  
$ch = curl_init( $url );  
curl_setopt( $ch, CURLOPT_POST, 1);  
curl_setopt( $ch, CURLOPT_POSTFIELDS, $myvars);  
curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, 1);  
curl_setopt( $ch, CURLOPT_HEADER, 1);  
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1);  
// grab URL and pass it to the browser  
$response = curl_exec( $ch );  
$responseinfo = curl_getinfo($ch);  
// close cURL resource, and free up system resources  
curl_close($ch);  
if($responseinfo["http_code"] == 200){  
    http_response_code(200);  
    echo "Form received.";  
}  
else{  
    http_response_code(421);  
    echo "Form not received, please try again or contact with us by phone";  
}
```

# OUR Back end

Let's test it. Positive data set send from Postman to OUR back end must be forwarded to Google Form API and saved in our DB (spreadsheet):

The screenshot displays two applications side-by-side. On the left is the Postman interface, and on the right is a Google Sheet.

**Postman Interface:**

- Request:** Method: POST, URL: `http://demo.localhost/form.php`.
- Body (JSON):**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> postEmail	test_from_postman_to_our_BE@w2business.pl	
<input checked="" type="checkbox"/> postName	Name in Postman to Our BE	
<input checked="" type="checkbox"/> postMessage	Message from Postman to Our BE	
Key	Value	Description
- Response:** Status: 200 OK, Time: 425 ms, Size: 267 B. The body shows: `Form received.`

**Google Sheet Interface:**

- Columns:** B (Email address), C (Name), D (Your message).
- Data Rows:**

B	C	D
test@w2business.pl	Test form name	Test form message
test_from_postman_to_our_BE@w2business.pl	Test form name from Postman	Test form message from Postman
test_from_our_html@w2business.pl	Test form name from Our HTML	Test form message from Our HTML
test_from_FE_JS1@w2business.pl	Test form name from JS	Test form message from JS
test_from_our_FE_JS@w2business.pl	Name in HTML form	Message from HTML form by JS DB
test_from_postman_to_our_BE@w2business.pl	Name in Postman to Our BE	Message from Postman to Our BE

Green arrows indicate the flow of data from the Postman request body to the corresponding rows in the Google Sheet.

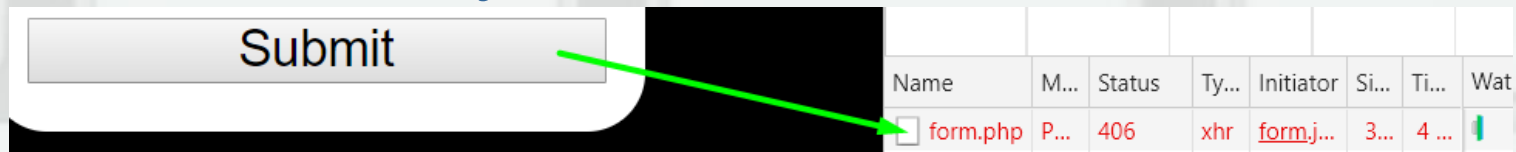
## Final solution (integration FE-MW)

1. Copy code from form.html to index.php (also You can rename this file to index.html)
2. Place files form.js and style.css in website root directory
3. In form.js replace line 49 with this call to our back end: `xhr.open("POST", "form.php");`

Back end become a middleware now.

# Final solution (update FE)

Open our FE (<http://demo.localhost/>) in browser and try submit form



FE send to BE incorrect POST form keys values, so need update html form code:

```
<input type="email" name="postEmail" placeholder="Enter your E-mail">  
<input type="text" name="postName" placeholder="Enter your Name">  
<input type="text" name="postMessage" placeholder="Enter your Message">
```

FE form.js code also need update. Replace all emailAddress for postEmail  
entry.1224778685 for postName  
entry.512804397 for postMessage



# Final solution (test)

Let's try submit correct data to OUR website form and check data saved in DB:

The screenshot shows a web browser at `demo.localhost` with a form containing the email `test-FE-MW-BE@w2busi`, the text `Final Solution Name`, and the text `Final Solution message`. A green arrow points from the `Submit` button to the network tab. The network tab shows a list of requests, with the last request highlighted. A green box highlights the response data in the right pane, which is a JSON object with the following structure:

B	C	D
Email address	Name	Your message
test_from_our_FE_JS@w	Name in HTML form	Message from HTML form
test_from_postman_to_ou	Name in Postman to Our	Message from Postman to
test-fe-mw-be@w2busine	Final Solution Name	Final Solution message

Now it works without redirect to Google forms!

# Final solution (update FE)





## Let's update our FE in form.js

Disable submit while wait for response (for test use sleep(5); at form.php), show errors from middleware hide form if success and :

```
xhr.addEventListener("readystatechange", function () {  
  if (this.readyState !== 4) {  
    console.log("while wait for response result, block form");  
    document.getElementsByName("submit")[0].disabled = true;  
  }  
  if (this.readyState === 4) {  
    console.log(this.responseText);  
    if (this.responseText == "Form received.") {  
      document.getElementsByTagName("form")[0].innerHTML = this.responseText;  
    }  
    else {  
      document.getElementsByName("submit")[0].disabled = false;  
      let errorBlock = document.createElement("div");  
      errorBlock.appendChild(document.createTextNode(this.responseText));  
      document.getElementById('formErrors').appendChild(errorBlock);  
    }  
  }  
});
```

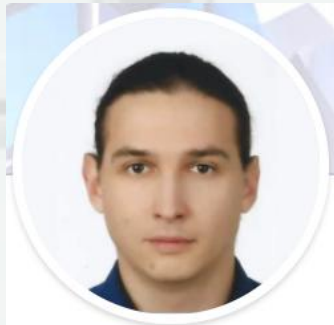
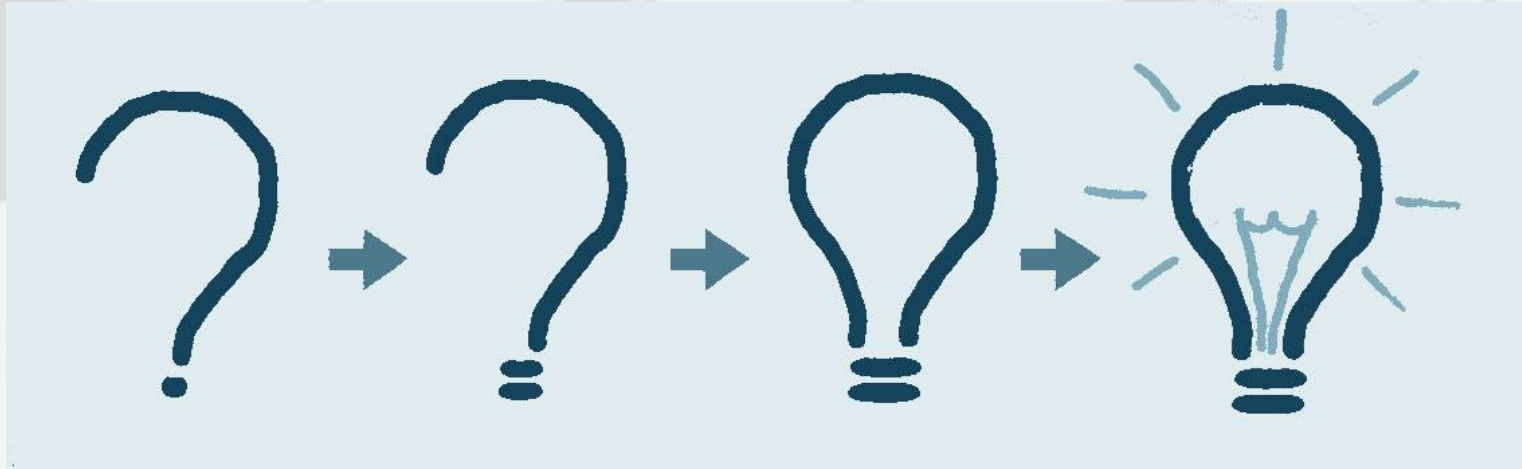
# FINAL SOLUTION ARCHITECTURE

Let's analyze our final solution layers:

Front end	OUR WEBSITE HTML+JS	JS update DOM (HTML) 	Client browser show our HTML form, validate and send data to middleware								
Middle ware	Our server side BE	PHP 	Server side validation, communication with BE and return results to FE								
Back end	GOOGLE FORM REST API	Google API 	REST API endpoint receive POST request, validate and save to DB								
	DB	spreadsheet 	Stores data <table><tr><th>B</th><th>C</th><th>D</th></tr><tr><td>Email address</td><td>Name</td><td>Your message</td></tr><tr><td>test@w2business.pl</td><td>Test form name</td><td>Test form message</td></tr></table>	B	C	D	Email address	Name	Your message	test@w2business.pl	Test form name
B	C	D									
Email address	Name	Your message									
test@w2business.pl	Test form name	Test form message									

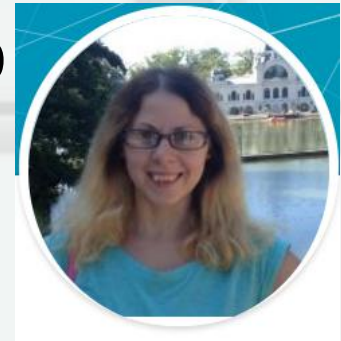
# THANK YOU...

## Do you have any questions?



**Svitlana Samko**  
<https://www.linkedin.com/in/svitlana-samko/>

**Andrii Stepura**  
<https://www.linkedin.com/in/andriistepura/>



*Graphics from focus-education.co.uk*