

EmailJS

とりあえず結論からいうと `preventDefault();` がform送信時の処理内にかかれていないのではと予想してます。

(自分も当初上手く送信されなかったのでデバッグした所、上記が原因で送信が出来てなかっただけでした。)

EmailJSの設定情報、コード自体にも誤りがないのにメール送信が上手くいっていませんでしたのであれば結構濃厚だと思います。

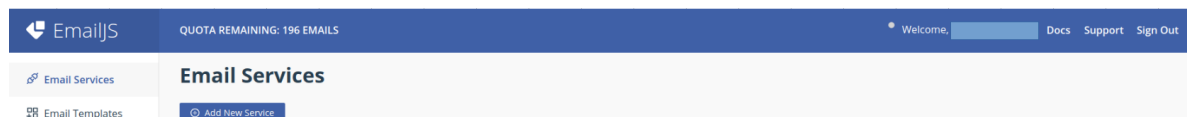
現状のコードのどの部分に`preventDefault();` を書けばいいか調べて試してみても良いと思いますし、

一応自分の環境で上手く行った手順を以降につらつらと書いてますので

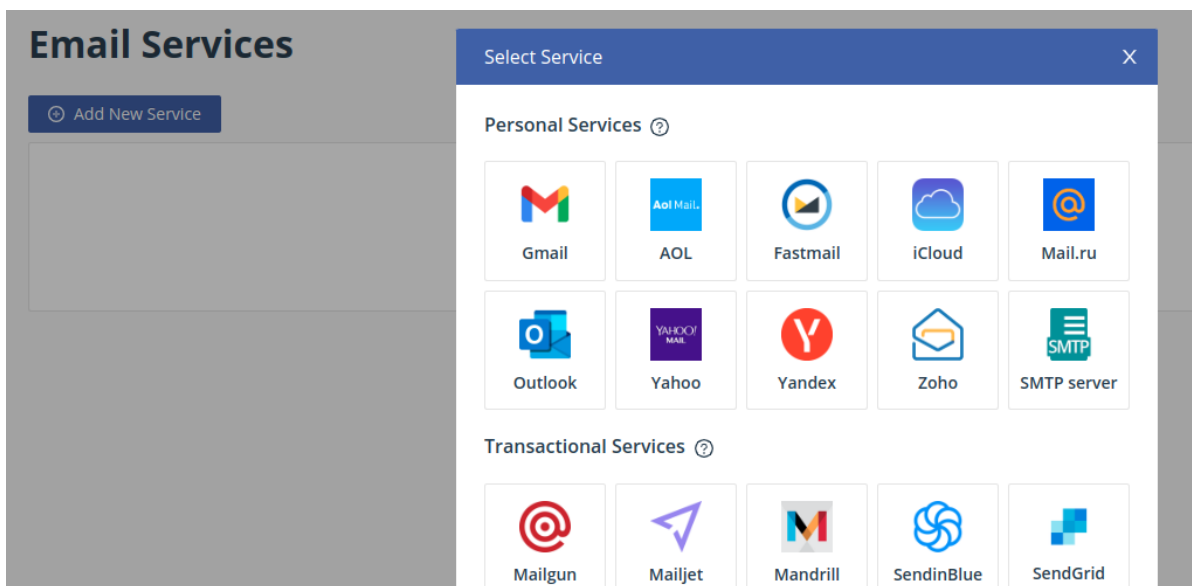
参照しながら最初からやり直してみてもいいかもです！

EmailJSのアカウント登録からテストmail送信まで

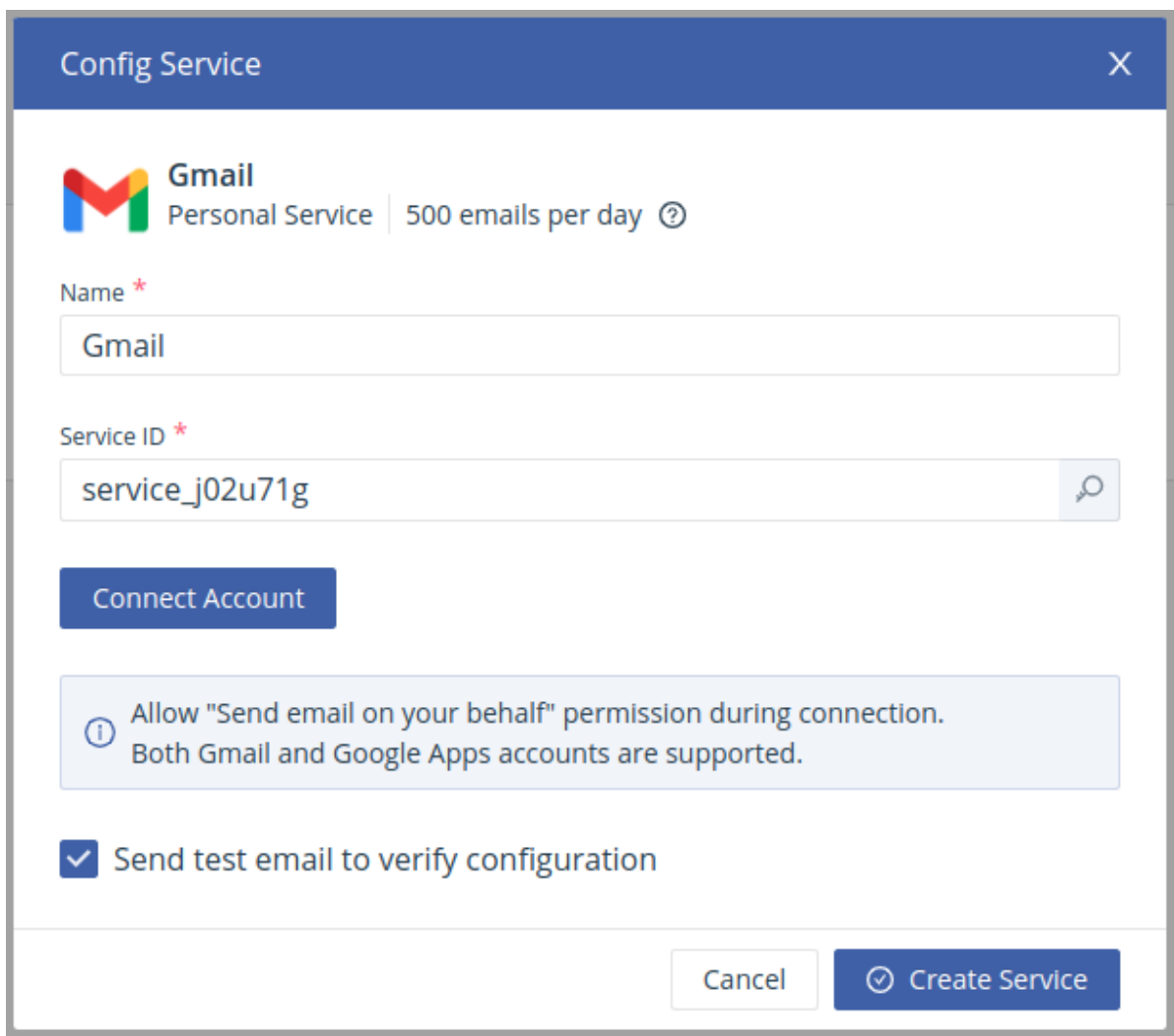
- 「Add New Service」をクリック



- 「Gmail」をクリック



- 「Connect Account」をクリック



- 使用するGoogleアカウントを選択




- 「ユーザー本人に代わってメールを送信 です。」も必ずチェック！！その後「続行」をクリック



- 「Create Service」をクリック

Config Service


 **Gmail**
Personal Service | 500 emails per day ?

Name *

Service ID *

Gmail Connect

Connected as Disconnect

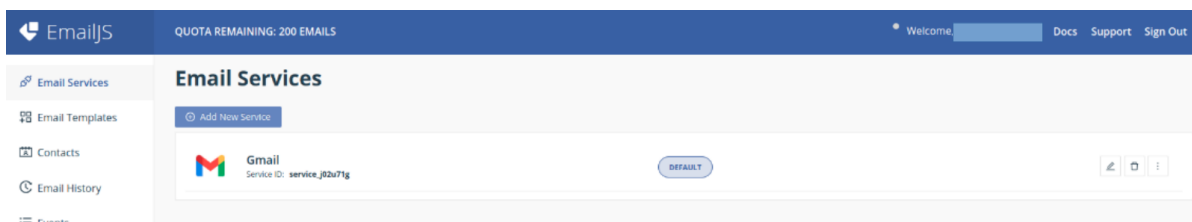
 Allow "Send email on your behalf" permission during connection.
Both Gmail and Google Apps accounts are supported.

☒ Send test email to verify configuration

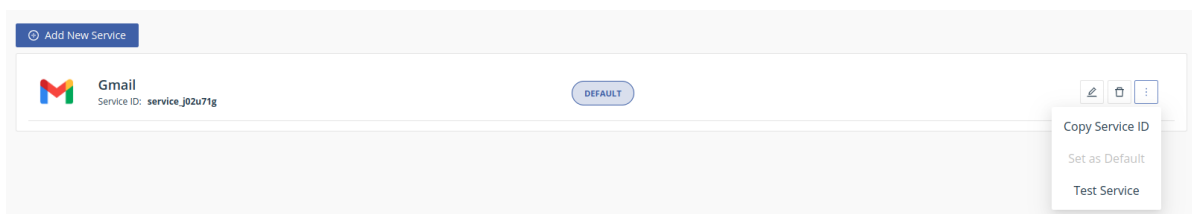
Cancel

Create Service

- 以下のようにGmailがServiceとして追加されるので三点リーダー「・・・(縦)」をクリック



- 「Test Service」をクリック



- 「Send the Test Email」 をクリックしてテストメールを送信

Test Email Service

×

ⓘ Test email will be sent to

Send the test email via the "Gmail" email service to administrator's email address.

Response

⚠ Send the Test Email

Cancel

- 200 OK がResponseに返ってきてればOK

Test Email Service

×

ⓘ Test email will be sent to

Send the test email via the "Gmail" email service to administrator's email address.

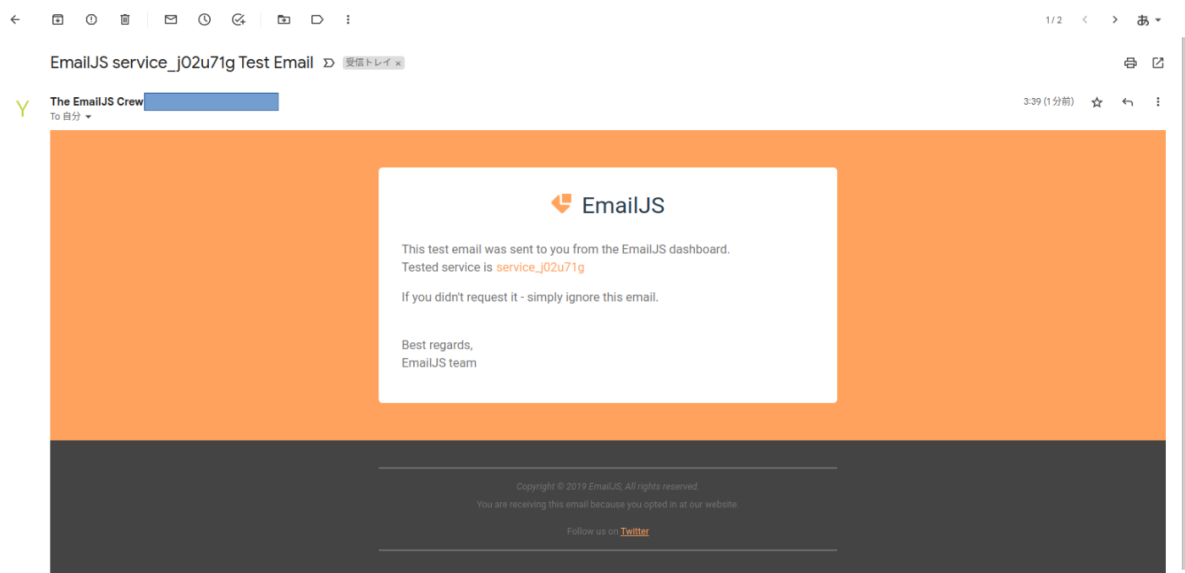
Response

200 OK

⚠ Send the Test Email

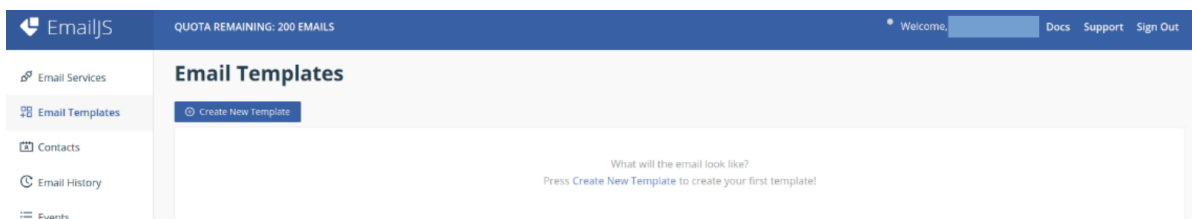
Cancel

- 念の為本当に届いているかを確認



Templateを設定

- サイドメニューの「Email Templates」をクリック



- 「Create New Tempate」をクリックすると以下の画面となる
- 以下のように content 部分のみ任意の値に書き換えてから「Save」をクリック

※他はデフォルト設定のまま(変えたかったら全部うまくいった後にご自分で調整しながら変えてください。)



(上記キャプチャ content 部分のコピペ用)

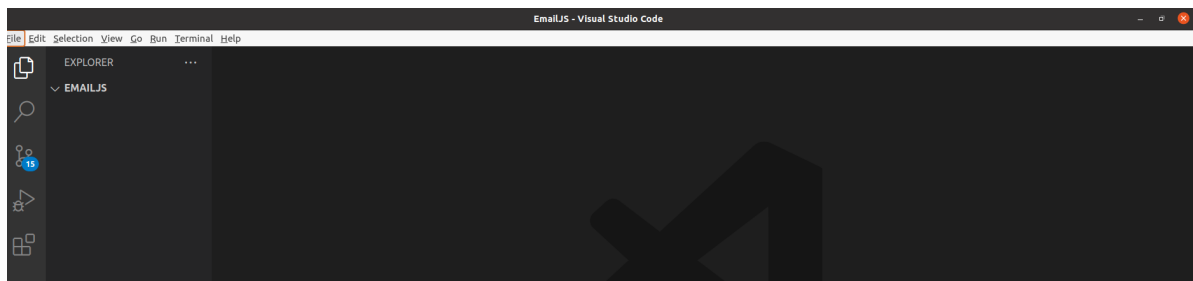
```
1   こんにちは 、
2
3   {{from_name}} : ({{email}}) から新しいメッセージが届きました
4
5   {{message}}
6
7   幸運をお祈りしています!!
8   EmailJS team
```

JavaScriptから送信してみる

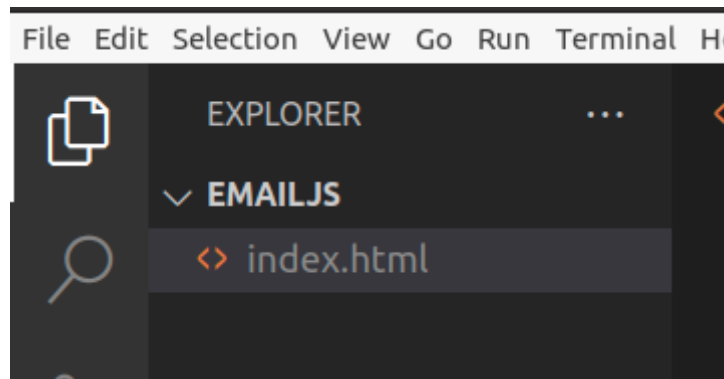
- Desktop配下に「EmailJS」フォルダを作成してVSCodeで開く(本当はどこでも良いし名前も何でも良いです笑)



以下のようにVSCodeで開くとEXPLORERの表示上は大文字になっているが気にしない



- 直下にindex.htmlを作成



index.htmlのコードは以下

```
1 <!DOCTYPE html>
2 <html lang="ja">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-
8     scale=1.0">
9   <link
10     href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bo
11       otstrap.min.css" rel="stylesheet"
12     integrity="sha384-
13     EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSspd3yD65VohhpuuCOmLAS
14     jC" crossorigin="anonymous">
15   <title>Document</title>
16 </head>
17
18 <div class="container mt-5">
19   <h1>Test Mail</h1>
20   <form class="my-5" id="form" novalidate>
21     <div class="mb-3">
22       <label for="email" class="form-label">Email
23       address</label>
```

```
18     <input type="email" class="form-control" id="email"
placeholder="name@example.com">
19   </div>
20   <div class="mb-3">
21     <label for="name" class="form-label">Your Name</label>
22     <input class="form-control" id="name" rows="3"></input>
23   </div>
24   <div class="mb-3">
25     <label for="message" class="form-label">Example
textarea</label>
26     <textarea class="form-control" id="message" rows="3">
</textarea>
27   </div>
28   <button class="btn btn-success" id="sendMailButton">Email
sending test !!</button>
29 </form>
30 </div>
31
32 <body>
33   <!-- EmailJSを使えるようにCDNを取得 -->
34   <script
src="https://cdn.jsdelivr.net/npm/@emailjs/browser@3/dist/mai
l.min.js"></script>
35
36   <script>
37     // EmailJS使用前の初期化处理
38     (function () {
39       emailjs.init("YOUR_PUBLIC_KEY");
40     })();
41
42     // formの値をTemplateに渡すパラメータにセット
43     function templateVariablesSet(argEmail, argName,
argMessage) {
44       let obj = {
45         email: argEmail,
46         from_name: argName,
47         message: argMessage,
48       }
49       return obj;
50     };
51
52     // triggerとなるHTMLの要素を取得
53     const sendMailButton =
document.getElementById('sendMailButton');
54
```

```

55 // triggerとなるボタンを押下した際にメール送信処理を呼び出すように
EventListernerに登録しておく
56 sendMailButton.addEventListener('click', event => {
57     event.preventDefault(); // もしかしたらデフォルトの動作を止め
るこの処理が抜けてるのが一番の原因かも??
58     let argEmail = document.getElementById('email').value;
59     let argName = document.getElementById('name').value;
60     let argMessage =
document.getElementById('message').value;
61
62     templateVariables = templateVariablesSet(argEmail,
argName, argMessage);
63
64     emailjs.send('YOUR_SERVICE_ID', 'YOUR_TEMPLATE_ID',
templateVariables)
65     .then(function (response) {
66         alert('SUCCESS!');
67         console.log('SUCCESS!', response.status,
response.text);
68     }, function (error) {
69         alert('FAILED...');
70         console.log('FAILED...', error);
71     });
72 });
73 </script>
74
75 </body>
76
77 </html>

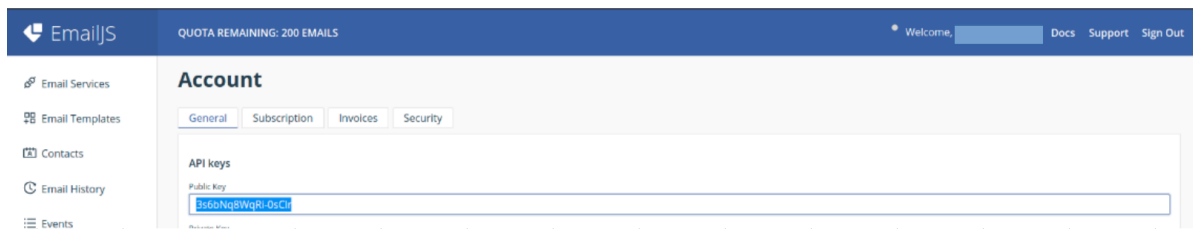
```

- 以下1. 2. 3. の箇所を自分の情報に書き換える

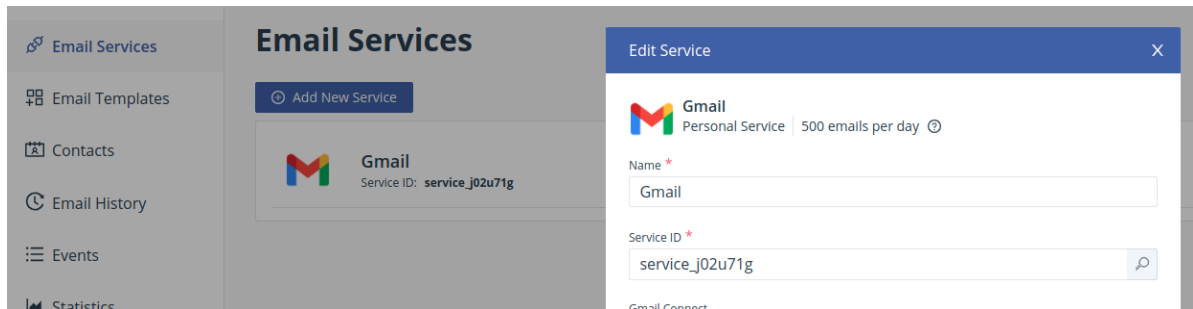
1. YOUR_PUBLIC_KEY (39行目)
2. YOUR_SERVICE_ID (64行目)
3. YOUR_TEMPLATE_ID (64行目)

参考)

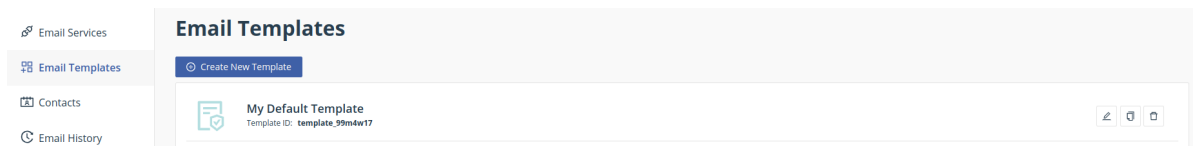
1. ブラウザに移動して右上のメニューのユーザ名をクリックしAccountページを開いたところにPublic key



2. サイドバーの「Email Services」で序盤に作成したService ID



3. サイドバーの「Email Templates」からTemplate ID

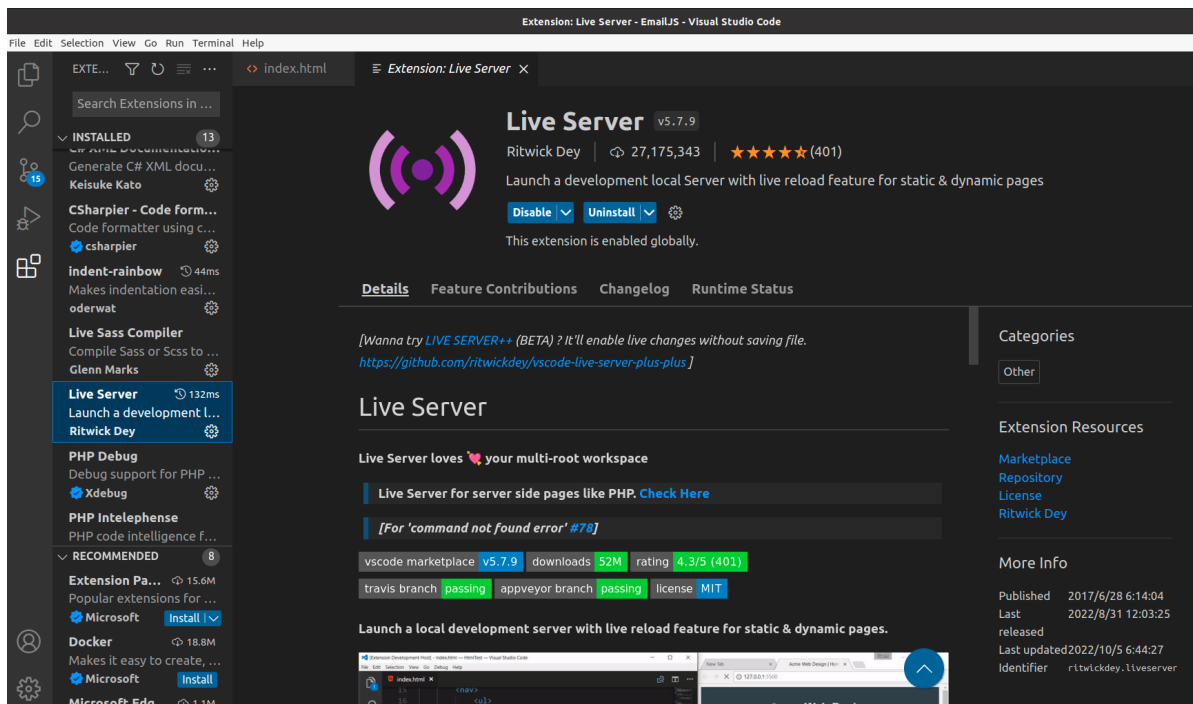


後はブラウザから実行してみる。

おまけ

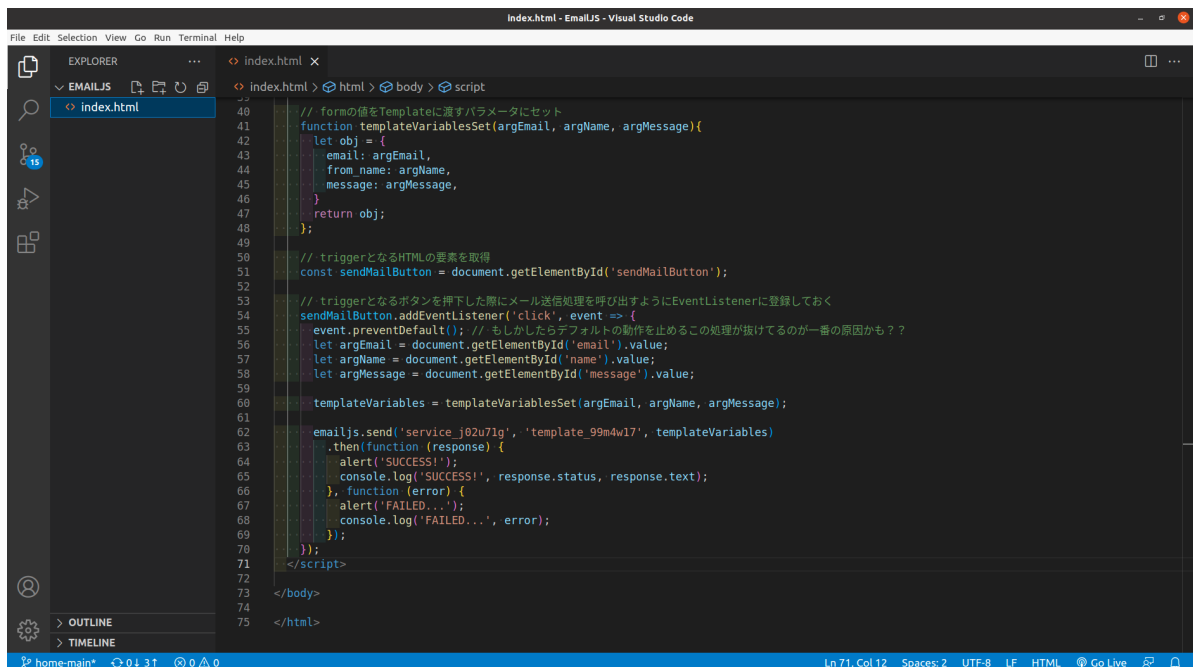
VSCodeで拡張機能の「Live Server」入れてないのであれば入れとくと今後も楽なのでおすすめです。

ちなみにこれ。



以降はLive Serverを使ってブラウザからメール送信を行うまでの一連のキャプチャ

- index.htmlを開いた状態で右下の「Go Live」をクリック



- 勝手にブラウザが起動するので後はフォームに任意の値を入力して「Email sending test !!」をクリック



Test Mail

Email address

Your Name

Example textarea

Email sending test !!

- 終了

参考サイト

[ブラウザのJavaScriptから直接メールを送れるPaaS、EmailJS](#)

[emailjs.send](#)