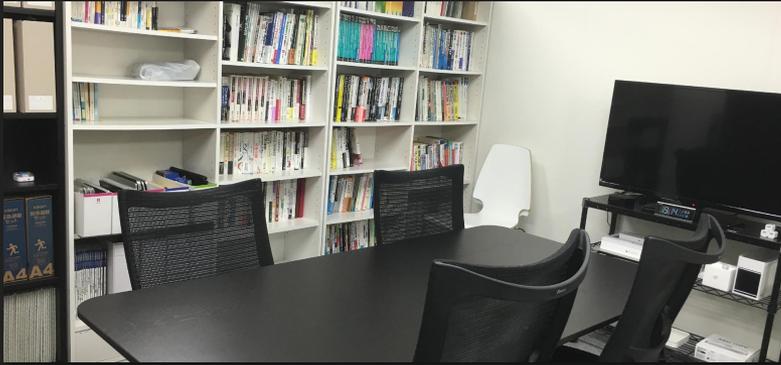


# espar Form

静的サイト・JAMStackサイト用フォームツール

rev.202106

# 株式会社フィードテイラーのご紹介



設立日	2006年7月7日
所在地	大阪市北区東天満 1-12-11
代表者	大石 裕一
資本金	1000万円
従業員数	6人 (アルバイト、外部パートナー含む)
免許/資格	ACN  Consultants Network AWS Technology Partner 情報処理安全確保支援士  
Web	<a href="https://www.feedtailor.jp/">https://www.feedtailor.jp/</a>

## Webサイト高速化・セキュリティ強化事業

espar

WordPressサイトの静的化技術を提供し、サイトの高速性と安全性を高めるサービス

espar Form

静的ページ上で動作する、PHPが不要なJavaScriptベースのメールフォームツール

## エンタープライズiOSアプリ 事業



自社・受託をあわせて実績120個以上。2008年より事業化。B2Bアプリが得意

 エンタープライズiOS研究所

エンタープライズiOSを専門とする国内唯一の特化型メディア

# PHP不要・メールサーバ不要のフォームツール

The screenshot shows the 'お問い合わせ' (Contact Us) page on the feedtailor.jp website. The form includes fields for '会社名' (Company Name), 'お名前' (Name), 'メール' (Email), 'ウェブサイト' (Website), '種類' (Type) with a dropdown menu, and '内容' (Content). A '確認' (Confirm) button is at the bottom. The page also features a navigation menu and a footer with copyright information.

espar FORM

通知



返信



お問い合わせメール 受信トレイ

株式会社フィードテ일러  
To 自分

お問い合わせページから、以下のお問い合わせを頂きました。

■ 入力情報

会社名：株式会社テスト  
お名前：テスト太郎  
メールアドレス：[oiishi@feedtailor.jp](mailto:oiishi@feedtailor.jp)  
ウェブサイト：<https://www.example.com/>  
種別：その他のお問い合わせ  
内容：  
テストの問い合わせ内容

■ 入力元情報

入力日時：2018/09/26 08:50:44  
リクエストIP：122.221.182.90  
ユーザーエージェント：Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_1; rv:52.0) Gecko/20100801 Firefox/52.0  
リファラ：<https://www.feedtailor.jp/form/>

お問い合わせ受付のご案内 受信トレイ

株式会社フィードテ일러  
To 自分

こちらは株式会社フィードテ일러のお問い合わせ窓口です。

この度は、御連絡を頂き誠にありがとうございます。  
以下の内容でお問い合わせを受付させて頂きました。

弊社担当者から追って御連絡致しますので今暫くお待ち下さい。

会社名：株式会社テスト  
お名前：テスト太郎  
メールアドレス：[oiishi@feedtailor.jp](mailto:oiishi@feedtailor.jp)  
ウェブサイト：<https://www.example.com/>  
種別：その他のお問い合わせ  
内容：  
テストの問い合わせ内容

株式会社フィードテ일러  
<https://www.feedtailor.jp/>  
[info@feedtailor.jp](mailto:info@feedtailor.jp)

# 静的化してスピードとセキュリティを担保するのが2020年以降のトレンド

The screenshot shows the Jamstack.org website. The browser address bar displays 'jamstack.org'. The page title is 'Why Jamstack'. The main content area is dark blue with white text. It features a navigation sidebar on the left with links like 'Learn', 'Site Generators', 'Headless CMS', and 'Community'. The main content area has a heading 'Why Jamstack' followed by a paragraph: 'The Jamstack architecture has many benefits, whether you're building a large e-commerce site, SaaS application or personal blog.' Below this are four sections, each with an icon and a title: 'Better Performance' (lightning bolt icon), 'Higher Security' (shield icon), 'Cheaper, Easier Scaling' (stack of coins icon), and 'Better Developer Experience' (code editor icon). Each section contains a short paragraph explaining the benefit. At the bottom of the sidebar, there are social media icons for GitHub, Twitter, and a generic social icon, and a 'Jamstack Conf' button.

## Why Jamstack

The Jamstack architecture has many benefits, whether you're building a large e-commerce site, SaaS application or personal blog.

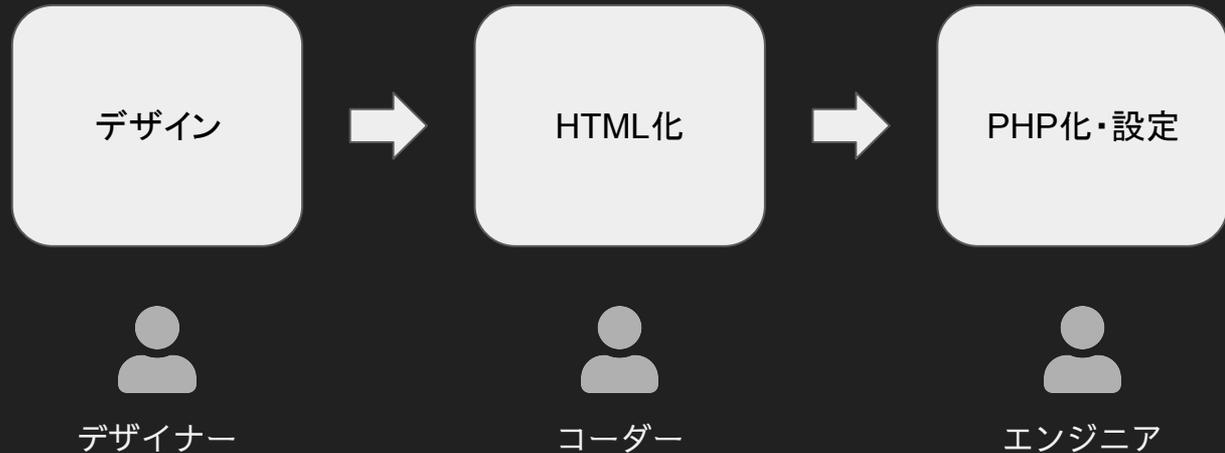
- Better Performance**  
Why wait for pages to build on the fly when you can generate them at deploy time? When it comes to minimizing the time to first byte, nothing beats pre-built files served over a CDN.
- Higher Security**  
With server-side processes abstracted into microservice APIs, surface areas for attacks are reduced. You can also leverage the domain expertise of specialist third-party services.
- Cheaper, Easier Scaling**  
When your deployment amounts to a stack of files that can be served anywhere, scaling is a matter of serving those files in more places. CDNs are perfect for this, and often include scaling in all of their plans.
- Better Developer Experience**  
Loose coupling and separation of controls allow for more targeted development and debugging, and the expanding selection of CMS options for site generators remove the need to maintain a separate stack for content and marketing.

静的化ページ上でjsを使って動的要素を実装する  
『Jamstack』のコンセプトが広がりつつある

# レガシーなフォームと espar form の違い

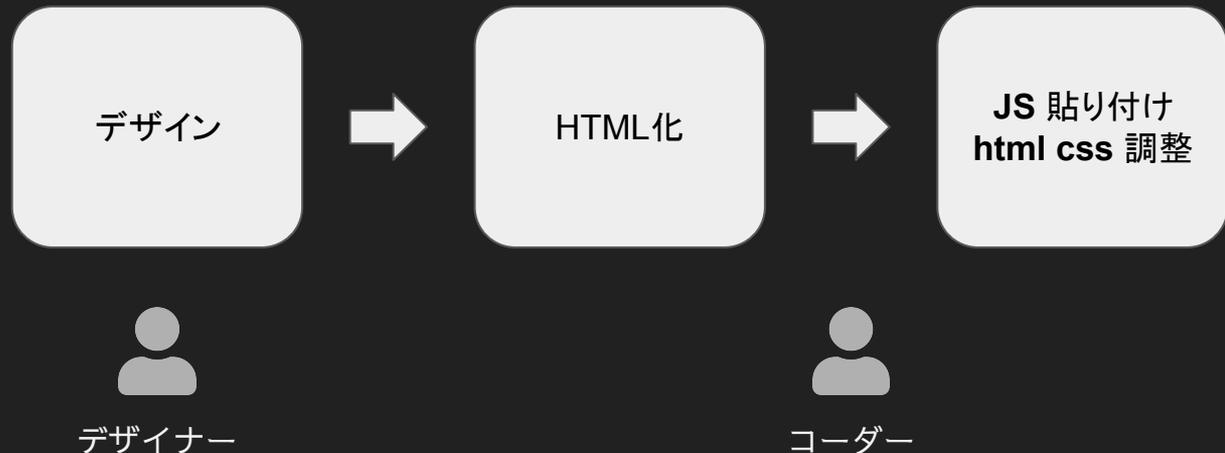
## legacy form

月に1回動く程度の仕組みに時間・お金をかけて、さらにセキュリティリスクを負う



## espar form

html/css/javascriptのみで短期間・低コストに実装しセキュリティリスクも軽減



 Jamstack ready

# espar form の導入手順

## ① フォームを置くページのhtml内に埋込コードを貼り付ける

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.17.0/jquery.validate.min.js"></script>  
<script src="https://bridge.espar.biz/v1/form.js"></script>  
<script>  
  var espar_form = { espf: { api_key: "ZdLwU7s5SIfAPK4x7XteFScaFxm1Sd7nL6N3REP" } };  
</script>
```

# espar form の導入手順

## ① フォームを置くページのhtml内に埋込コードを貼り付ける

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.17.0/jquery.validate.min.js"></script>
<script src="https://bridge.espar.biz/v1/form.js"></script>
<script>
  var espar_form = { espf: { api_key: "ZdLwU7s5SIfAPK4x7XteFScaFxm1Sd7nL6N3REP" } };
</script>
```

## ② 専用クラス等をhtmlに追記してフォームの振る舞いを定義する

```
<form method="POST" class="espf"> ← フォームを識別するID
  <table id="inquiry">
    <tbody>
      <tr>
        <th class="thw160 textalign-left">
          <label for="company">会社名</label>
        </th>
        <td>
          <input type="text" name="company" class="form-control" value="" id="company" size="50">
        </td>
      </tr>
      <tr>
        <th class="thw160 textalign-left">
          <label for="name">お名前</label>&nbsp;<span class="required">必須</span> ← 必須チェック
        </th>
        <td>
          <div><input type="text" name="name" class="form-control espf-required" value="" id="name" size="50"></div>
          <div class="form-input-error espf-e-name-required">名前は必須項目です</div> ← 必須チェックに失敗したら表示されるエラー文言用クラス
        </td>
      </tr>
    </tbody>
  </table>
</form>
```

# espar form の導入手順

## ③ 管理画面でメール内容や送信先の設定

admin.espar.biz

### メールテンプレート編集

共通お問い合わせフォーム (espf, /form)

基本情報 通知メール 返信メール チャット連携 外部 API 連携 詳細設定

返信メールを送信する

差出人 (必須) 株式会社フィードテイル<info@feedtailor.jp>

返信先

宛先 (必須) {{ .mail }}

Cc

Bcc

件名 (必須) お問い合わせ受付のご案内

メール本文

こちらは株式会社フィードテイルのお問い合わせ窓口です。

この度は、御連絡を頂き誠にありがとうございます。  
以下の内容でお問い合わせを受付させて頂きました。

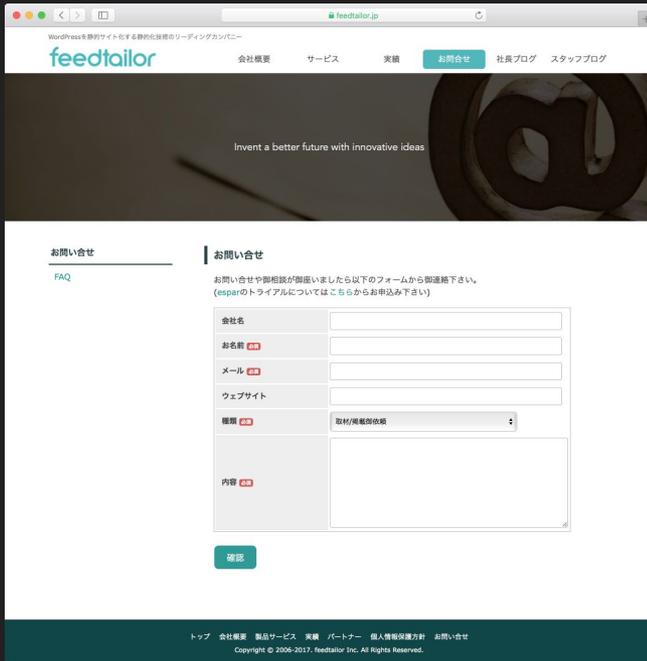
-----

会社名: {{ .company }}  
お名前: {{ .name }}  
メールアドレス: {{ .mail }}  
ウェブサイト: {{ .url }}  
種別: {{ .inquiry\_type }}  
内容:  
{{ .content }}

-----

弊社担当者から追って御連絡致しますので今暫くお待ち下さい。

# 以上の3ステップでhtmlに書いたフォームが動き出す



PHP・DB・CMS・メールサーバ不要

espar FORM

通知



返信



# 確認画面・完了画面は espar form が自動生成

## 入力画面

会社名	<input type="text"/>
お名前 <small>必須</small>	<input type="text"/>
メール <small>必須</small>	<input type="text"/>
ウェブサイト	<input type="text"/>
種類 <small>必須</small>	取材/掲載卸依頼
内容 <small>必須</small>	<input type="text"/>

確認

## 確認画面

会社名	株式会社テスト
お名前 <small>必須</small>	テスト太郎
メール <small>必須</small>	oishi@feedtailor.jp
ウェブサイト	https://www.example.com/
種類 <small>必須</small>	その他のお問い合わせ
内容 <small>必須</small>	テストの問い合わせ内容

戻る    この内容で問い合わせる

## 完了画面

お問い合わせありがとうございました



入力画面のhtmlだけ作る



コーダー

jsで自動生成

espar Form

# 入力条件(バリデータ)を多数用意。エラーは自動表示

## 入力画面

会社名

お名前 必須

メール 必須

ウェブサイト

種類 必須

内容 必須

確認

## エラー画面

会社名

お名前 必須  名前は必須項目です

メール 必須  正しいメールアドレス表記にしてください

ウェブサイト

種類 必須

内容 必須  内容は必須項目です

確認



入力画面の  
htmlだけ作る



コーダー



定義に従って  
自動表示

espar Form

# 豊富な入力条件(バリデータ)

条件識別子	入力値チェック内容
required	必須チェック
email	メールアドレス形式チェック
emailDomain	メールアドレス形式で@の後のドメインが存在するかどうかのチェック
url	URL形式チェック
number	符号、小数点、3桁ごとのカンマ付き半角数字のチェック
digits	半角数字のみかどうかチェック
alpha	半角アルファベットのみかどうかチェック
alphanumeric	半角英数字とアンダースコアのみかどうかチェック
zip	郵便番号形式のチェック
tel	電話番号形式のチェック
minlength	最小文字数チェック
maxlength	最大文字数チェック
min	最小値チェック
max	最大値チェック
equalTo	別の入力項目と一致しているかチェック

## espar form 独自!

お客様がメールアドレス入力を誤って連絡不能になってしまう事態を回避

# CSSクラスとエラー文言を追加するだけで条件チェックが発動

この input 要素は必須です  
という意味のクラスを追加する

```
<td>  
  <div><input type="text" name="name" class="form-control espf-required" value="" id="name" size="50"></div>  
  <div class="form-input-error espf-e-name-required">名前は必須項目です</div>  
</td>
```

エラー文言は最初から書いておく (CSSで display:none; を指定)  
必須条件を満たさない場合のみ表示される

espar form を動作させるのに必要なのはCSSクラス名のみ  
それ以外のhtmlは自由 → 他のフォームサービスに比べて自由度が高い

# 独自の入力条件(バリデーション)をjsで追加可能

バリデータテスト

独自の入力条件をjsで容易に実装が可能

```
validators: {  
  "notUseA": function(value) {  
    return (value.indexOf('A') < 0);  
  }  
}
```

実装はjsで条件式を書くだけ

```
<div>バリデータテスト</div>  
<div><input type="text" name="test" class="espfDemo-notUseA" /></div>  
<div class="espfDemo-e-test-notUseA" style="color:red;">文字列にAを含めないで下さい</div>
```

inputやselect要素に独自条件をクラス指定し、対応するエラー要素を定義する

# ファイル添付も可能

```
<div>添付ファイル</div>
<input type="file" id="attachment" name="attachment" class="espfDemo-maxsize-1m espfDemo-maxfiles-2 espfDemo-filetype-image" multiple>
<div class="espfDemo-e-attachment" style="color:red">
<span class="espfDemo-e-attachment-maxsize">ファイルのサイズが1メガバイトを超えています。</span>
<span class="espfDemo-e-attachment-maxfiles">ファイルは3つ以上添付できません。</span>
<span class="espfDemo-e-attachment-filetype">ファイルは画像形式のみです。</span>
</div>
```

添付ファイルの数・形式・サイズ等の自動チェックやエラー表示もhtml/cssだけで実現可

## 添付ファイル

ファイルの選択 20210329.docx

ファイルは画像形式のみです。

## 添付ファイル

ファイルの選択 4 ファイル

ファイルのサイズが1メガバイトを超えています。

## 添付ファイル

ファイルの選択 4 ファイル

ファイルは3つ以上添付できません。

# メールの送信先やタイトル・本文を設定する専用管理画面

espar メール ログアウト

## メールテンプレート編集

お問い合わせメール({{ .espfid }})

### 通知メール

**メールヘッダ (必須)**

from: 株式会社フィードテ일러<info@feedtailor.jp>  
to: info@feedtailor.jp  
subject: お問い合わせメール({{ .espfid }})

**メール本文 (必須)**

問い合わせページから、以下のお問い合わせを頂きました。

■ 入力情報

-----  
会社名: {{ .company }}  
お名前: {{ .name }}  
メールアドレス: {{ .mail }}  
ウェブサイト: {{ .uri }}  
種別: {{ .inquiry\_type }}  
内容:  
{{ .content }}  
-----

■ 入力元情報

-----  
入力日時: {{ .espf\_date }} {{ .espf\_time }}  
リクエストIP: {{ .espf\_remote\_addr }}  
ユーザーエージェント: {{ .espf\_user\_agent }}  
リファラ: {{ .espf\_referer }}  
-----

### 返信メール

**メールヘッダ**

from: 株式会社フィードテ일러<info@feedtailor.jp>  
to: {{ .mail }}  
#cc: info@feedtailor.jp

**メール本文**

こちらは株式会社フィードテ일러のお問い合わせ窓口です。

この度は、御連絡を頂き誠にありがとうございます。  
以下の内容でお問い合わせを受付させて頂きました。

弊社担当者から追って御連絡致しますので今暫くお待ち下さい。

# メール送信先や内容の柔軟な設定

**メール本文 (必須)**

問い合わせページから、以下のお問い合わせを頂きました。

■ 入力情報

-----

会社名 : {{ .company }}  
お名前 : {{ .name }}  
メールアドレス : {{ .mail }}  
ウェブサイト : {{ .url }}  
種別 : {{ .inquiry\_type }}  
内容 :  
{{ .content }}

-----

フォーム入力値に置き換わる変数

差出人 (必須)

返信先

宛先 (必須)

Cc

Bcc

To, Cc, Bcc, From, Reply-To を自由に設定

■ 入力元情報

-----

入力日時 : {{ .espf\_date }} {{ .espf\_time }}  
リクエストIP : {{ .espf\_remote\_addr }}  
ユーザエージェント : {{ .espf\_user\_agent }}  
リファラ : {{ .espf\_referer }}

-----

日時やIP・ユーザエージェント等の標準変数

.espf_date	フォーム入力日付
.espf_time	フォーム入力時間
.espf_remote_addr	フォーム入力ユーザの IPアドレス
.espf_user_agent	フォーム入力ユーザの UserAgent
.espf_referer	フォームの URL

**メール本文 (必須)**

```

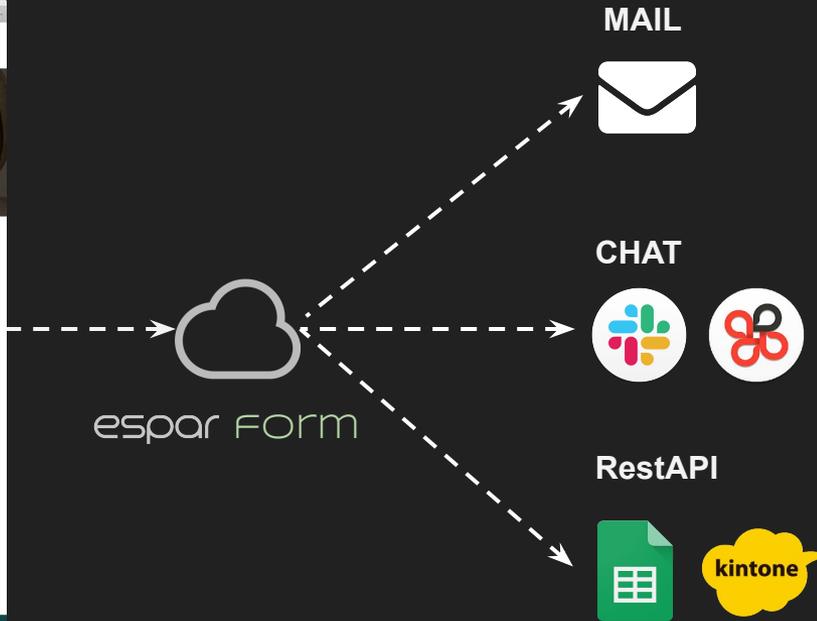
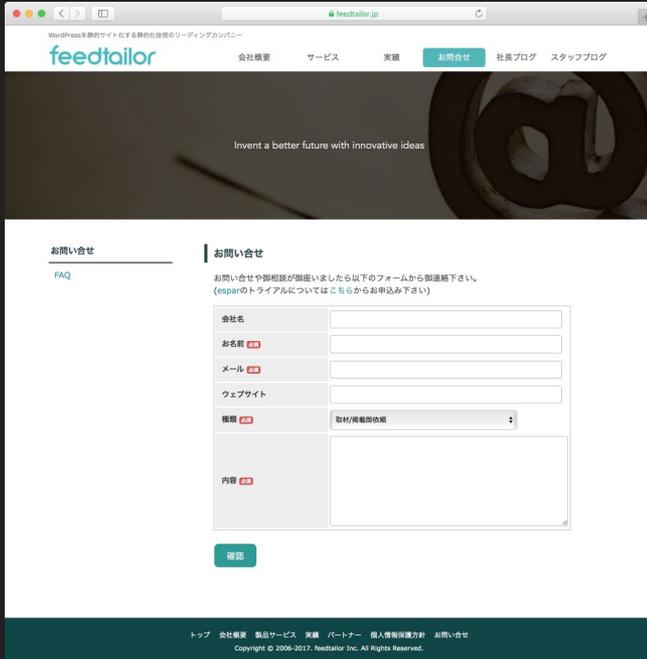
{{ if .meisho2 }}
-----
会議の名称 : {{ .meisho2 }}
会議の頻度 : {{ .hindo2 }}
会議の参加人数 : {{ .ninzu2 }}
会議資料の枚数 : {{ .maisuu2 }}
-----
{{ end }}
{{ if .meisho3 }}
-----

```

if 構文を使って条件分岐が可能



# Google SpreadSheet 連携 / kintone 連携



A1	G	H	I	J	K	L	M	N	O
1	お名前	フリガナ	メールアドレス	郵便番号	都道府県	市区部	町名番地	建物名	電話番号(携帯)
2	nameChara	nameKana		1100000	東京都	台東区	streetAf	building	2234567890
3	nameChara	nameKana		1100000	東京都	台東区	streetAf	building	2234567890
4	nameChara	nameKana		1100000	東京都	台東区	streetAf	building	2234567890
5	nameChara	nameKana		1000000	北海道	local	street	building	2234567890
6	nameChara	nameKana		1000000	北海道	local	street	building	2234567890
7	テスト太郎	テストタロウ		5300011	大阪府	大阪市北区大深町	1-1-1	ビル	09099999999
8	タッチ太郎	タッチタロウ		5400011	大阪府	大阪市中央区農人橋	1-2-3	ビルビル	09088888999
9	テスト太郎	テストタロウ		5300011	大阪府	大阪市北区大深町	1-1-1	ビル6階	0612345678
10	単価タロウ	タンカタロウ		5300011	大阪府	大阪市北区大深町	9-8-7	ビル1階	09099998888
11	テスト潮平	テスト		5320011	大阪府	大阪市淀川区西中島	1-1-1		09012345678

Google SpreadSheet 連携。入力内容を自動保存。あとからCSVダウンロードも可能

# Google SpreadSheet 連携

Apps Script esapr\_form\_api

デプロイ

以前のエディタを使用

```
1 function doPost(e){
2   var ss      = SpreadsheetApp.getActiveSpreadsheet();
3   var sheet   = ss.getSheetByName('data');
4   var postData = JSON.parse(e.postData.contents);
5
6   // 行の最後に値を追加
7   sheet.appendRow([
8     postData.company,
9     postData.name,
10    postData.mail,
11    postData.url,
12    postData.inquiry_type,
13    postData.content,
14    postData.espf_date,
15    postData.espf_time,
16    postData.espf_remote_addr,
17    postData.espf_user_agent,
18    postData.espf_referer,
19    postData.espf_version
20  ]);
21 }
```

SpreadSheetのスクリプトエディタを使ってGASで連携コードを作成する

# Google SpreadSheet 連携

デプロイを管理

アクティブ	設定
espar form の標準...	バージョン バージョン 4 (10月1日 14:51)
アーカイブ済み	説明
initial version	デプロイ ID
	コピー
	ウェブアプリ
	URL <a href="https://script.google.com/macros/s/">https://script.google.com/macros/s/</a>

キャンセル    デプロイ

RestAPIとしてデプロイし、URLを espar form の外部API連携に指定する  
(詳細はオンラインマニュアル参照)

# 振る舞いを自由にカスタマイズできる豊富なコールバック

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.17.0/jquery.validate.min.js"></script>
<script src="https://bridge.espar.biz/v1/form.js"></script>
<script>
  var espar_form = {
    espfSim: {
      api_key: "ZdLwU7s5SIfAPK4x7XteFScaFxm1Sd7nL6N3REP";
      submit_success_callback: function() {
        console.log("success");
      }
    }
  }
</script>
```

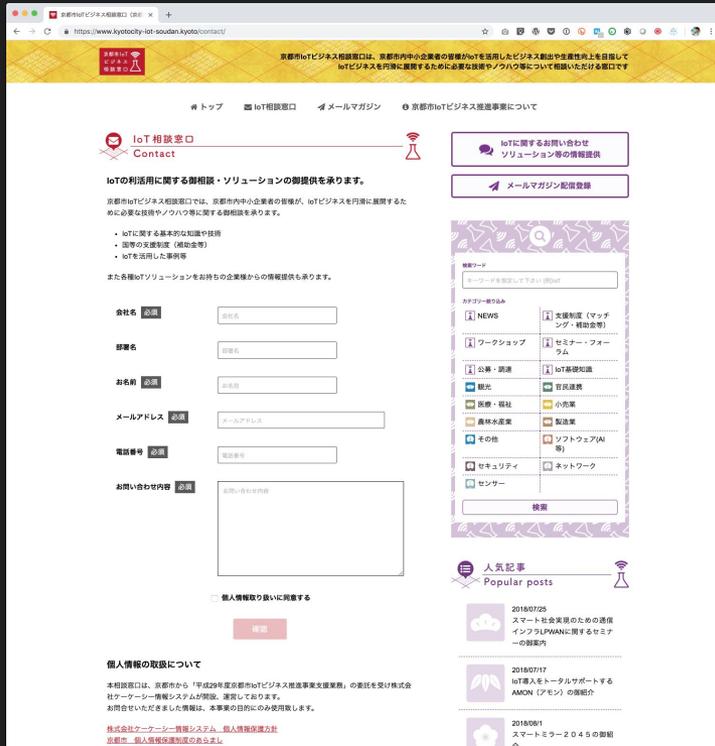
← submitが成功した後に呼ばれる独自の挙動

コールバック	コールバックが呼ばれるタイミング
input_ready_callback	入力画面が表示されきったあと
confirm_ready_callback	確認画面が表示されきったあと
validation_success_callback	入力画面で条件チェックエラーがなかったとき
validation_failure_callback	入力画面で条件チェックエラーがあったとき
submit_success_callback	送信が成功して完了したあと
submit_failure_callback	送信が失敗して終了したあと
back_clicked_callback	確認画面で戻るボタンがクリックされた直後
submit_clicked_callback	送信ボタンがクリックされた直後
confirm_clicked_callback	確認ボタンが押された直後

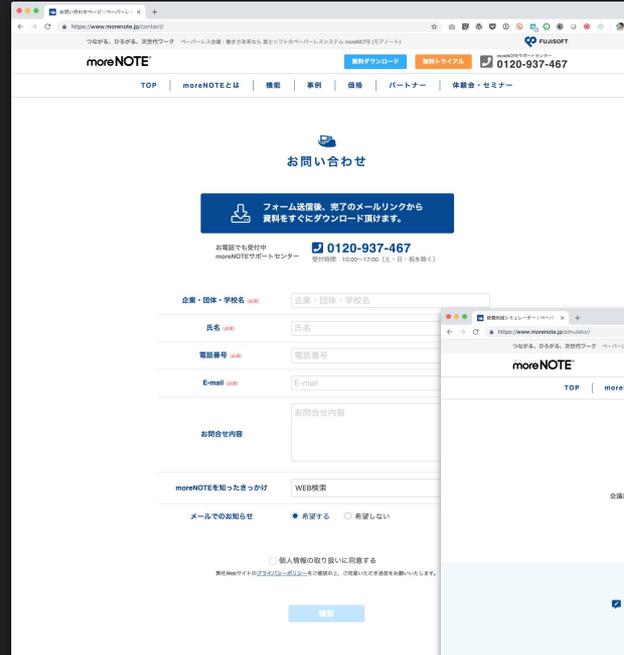
# espar form の導入事例

## 富士ソフト(株)様 moreNOTE製品サイト

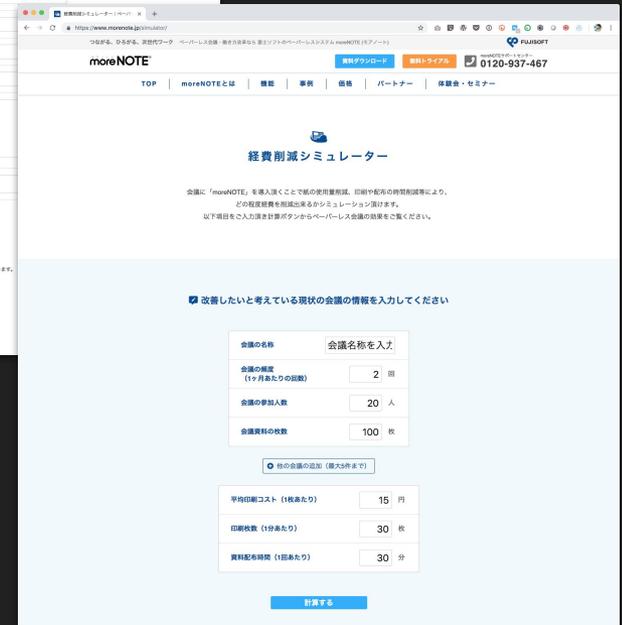
### 京都市様 京都市IoTビジネス相談窓口



<https://www.kyotocity-iot-soudan.kyoto/contact/>



<https://www.morenote.jp/contact/>



<https://www.morenote.jp/simulator/>