



Multi-Language Account Engagement Forms



Adam Erstelle
Sercante, VP of Technology

Thanks To Our Incredible Sponsors!

SERCANTE

salesforce

stensul

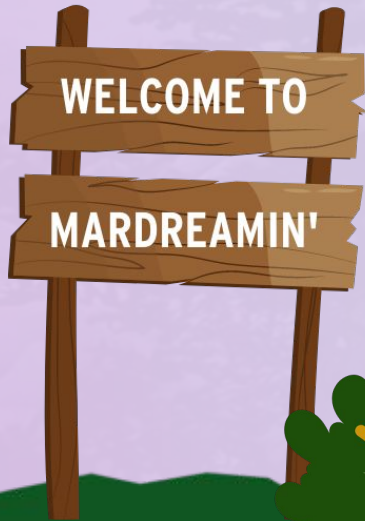
PFL

GRIDMATE

THE SPOT
by SERCANTE

FIONTA

Agenda



- Current approaches
- Challenges with current
- A different solution
- Demo
- How it Works

Current approaches




Prospect field-driven Dynamic Content








1. Create a Prospect Custom Field for Language, include in all Forms
2. Create Layout Templates & Forms in each Supported Language
3. Create Dynamic Content to choose which Form to display based on Language
Field
4. Check out our Forms' Performance



Prospect field-driven Dynamic Content

 Forms

Filter: [+ Add Form](#)

<input type="checkbox"/>	FORMS 	TOTAL VIEWS	DOMAIN TYPE	AUTORESPONDER	UPDATED AT	ACTIONS
<input type="checkbox"/>	Sign up - Spanish  /Uncategorized/Forms	0	Custom		Sep 1, 2023 7:01 AM	
<input type="checkbox"/>	Sign up - French  /Uncategorized/Forms	0	Custom		Sep 1, 2023 7:01 AM	
<input type="checkbox"/>	Sign up - English  /Uncategorized/Forms	0	Custom		Sep 1, 2023 7:00 AM	


With 0 selected: Showing of 3 Page 1 of 1



Prospect field-driven Dynamic Content

Contents


Name Sign Up Form - Multi Language


Folder  /Uncategorized/Dynamic Contents

Tracker Domain Current Primary (<http://www3.lenoxsoft.com>)

Tags

Based On Custom Field: Language

Default Content 

Variation A: Is fr 

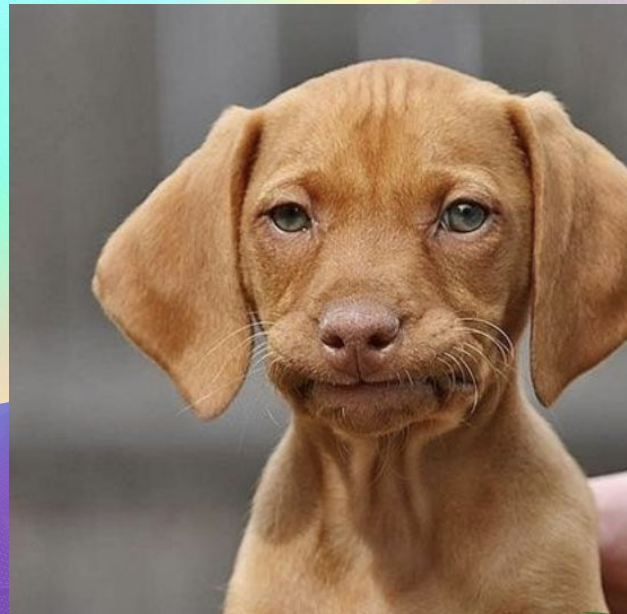


Challenges with Current Approach



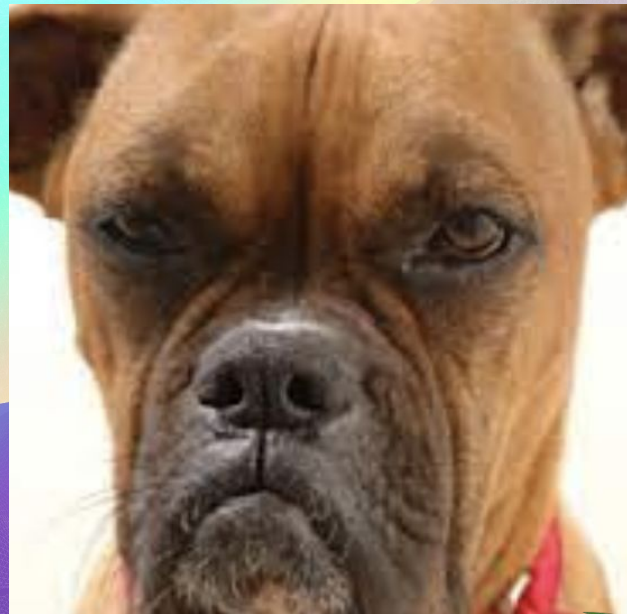
Visitor Experience

1. First time visitors get Default Language
2. Language Selection doesn't instantly apply



Visitor Experience

1. A lot of assets to create
2. Reporting is split across Forms
3. Compounding complexity with elaborate Engagement Studio Programs, automations



A different solution




Single Form, Multiple Languages


1. Account Engagement Form, with placeholders for Text
 - a. Fewer Assets to create
 - b. Supports as many languages as you want
 - c. Reporting & Automations are streamlined
2. Requires the tiniest bit of Javascript knowledge




Single Form, Multiple Languages




Language 

First Name 

Last Name 

Company

Email 

Industry

Single Form, Multiple Languages



1 Name 2 **Fields** 3 Look and Feel 4 Completion Actions 5 Confirm & Save

Form Fields Preview

[[i18n.language]] (a) *	<input type="text"/>	↕ ✎ 🗑
[[i18n.firstname]] (a) *	<input type="text"/>	↕ ✎ 🗑
[[i18n.lastname]] (a) *	<input type="text"/>	↕ ✎ 🗑
[[i18n.company]] (a)	<input type="text"/>	↕ ✎ 🗑
[[i18n.email]] (a) *	<input type="text"/>	↕ ✎ 🗑
[[i18n.industry]] (a)	<input type="text"/>	↕ ✎ 🗑

+ Add New Field

« Previous Next » Confirm & Save Cancel

Single Form, Multiple Languages



1 Name 2 Fields 3 Look and Feel 4 Completion Actions 5 Confirm & Save

Look and Feel

Layout Template Choose

Submit Button Text

Above Form Below Form Styles Advanced

B I U S x₂ x² [List Icons] [Table Icon] [Image Icon] [Link Icon] [Unlink Icon] [Font] - [Size] - [Styles] - [Format] - Merge Field

```
1 <script>
2 // get the language the Prospect has saved
3 const prospectLang = "{{{{Recipient.Language}}}"; //Pardot will replace this with the Prospect's Language
4 Custom Field Value, if they have\
5 // kick off translations
6 i18nForm(prospectLang);
7 </script>
```

« Previous Next » Confirm & Save Cancel

Single Form, Multiple Languages



1 Name 2 Fields 3 Look and Feel 4 **Completion Actions** 5 Confirm & Save

Completion Actions

Thank You Content [Thank You Code](#)

Thank You Content

B *I* U ~~S~~ ^{x²} _{x²} | [List] [List] | [Align] [Align] | [Quote] [Quote] | [List] [List] [List] [List] | [Link] [Link] [Link] [Link]

[Image] [Table] [Table] [Table] | [Omega] [Lightning] [Smiley] [Quote] [Quote]

← → [Undo] [Redo] [I_x] **A** **A** | Font - Size - Styles - Format -

🔍 Merge Field

[[11&n.thankyou]]

ⓘ Thank you content is displayed after the form is filled out

Always display form after submission
Disable the 10 minute waiting period after submitting a form

Redirect to the following URL instead of showing the form's Thank You Content

Single Form, Multiple Languages



1 Name 2 Fields 3 Look and Feel 4 **Completion Actions** 5 Confirm & Save

Completion Actions

Thank You Content Thank You Code

Thank You Code


```
<script>
// get the language the Prospect has saved
const prospectLang = "({{Recipient.Language}})"; // Pardot will replace this with the
Prospect's Language Custom Field Value if they have
// kick-off translations
i18nForm(prospectLang);
</script>
```

Thank you code is typically Javascript used to call other systems like Google Conversion Tracking

Always display form after submission
Disable the 10 minute waiting period after submitting a form

Redirect to the following URL instead of showing the form's Thank You Content

Completion Actions

+ 

Demo



How it works



Prospect field-driven Dynamic Content



handlebars



Minimal templating on steroids



How it works



Simple expressions

The following template defines two Handlebars expressions

```
<p>{{firstname}} {{lastname}}</p>
```

template 


If applied to the input object

```
{  
  firstname: "Yehuda",  
  lastname: "Katz",  
}
```

input 

the expressions will be replaced by the corresponding properties. The result is then

```
<p>Yehuda Katz</p>
```

output 

How it works



Nested input objects

Sometimes, the input objects contains other objects or arrays. For example:

```
{
  person: {
    firstname: "Yehuda",
    lastname: "Katz",
  },
}
```

input 

In such a case, you can use a dot-notation to gain access to the nested properties

```
{{person.firstname}} {{person.lastname}}
```

template 



How it works



Nested input objects

Sometimes, the input objects contains other objects or arrays. For example:

```
{
  i18n: {
    firstname: "Prénom",
    lastname: "Nom de famille",
  },
}
```

input 

In such a case, you can use a dot-notation to gain access to the nested properties

```
{{i18n.firstname}} {{i18n.lastname}}
```

template 



How it works

Nested input objects

Sometimes, the input objects contains other objects or arrays. For example:

```
{
  i18n: {
    firstname: "Prénom",
    lastname: "Nom de famille",
  },
}
```

input 

In such a case, you can use a dot-notation to gain access to the nested properties

```
[[i18n.firstname]] [[i18n.lastname]]
```

template 



Translation Placeholders as Labels

1 Name 2 Fields 3 Look and Feel 4 Completion Actions 5 Confirm & Save

Form Fields Preview

[[i18n.language]] (a) * ↕ ✎ 🗑️

[[i18n.firstname]] (a) * ↕ ✎ 🗑️

[[i18n.lastname]] (a) * ↕ ✎ 🗑️

[[i18n.company]] (a) ↕ ✎ 🗑️

[[i18n.email]] (a) * ↕ ✎ 🗑️

[[i18n.industry]] (a) ↕ ✎ 🗑️

+ Add New Field

« Previous Next » Confirm & Save Cancel

The Form's Translations

Stored in a
JS Object

1st layer:
Language
Code

2nd layer:
i18n key

3rd layer:
each
translation

The Form's Translations

```
1 ▾ const translations = {  
2 ▾   'en': {  
3 ▾     i18n: {  
4         firstname: "First Name",  
5         lastname: "Last Name"  
6     }  
7 },  
8 ▾   'fr': {  
9 ▾     i18n: {  
10         firstname: "Prénom",  
11         lastname: "Nom de famille"  
12     }  
13 },  
14 ▾   'es': {  
15 ▾     i18n: {  
16         firstname: "Primer nombre",  
17         lastname: "Apellido"  
18     }  
19 }  
20 };
```



The Form's Translations

```
1 ▾ const translations = {
2   'en': {
3     i18n: {
4       firstname: "First Name",
5       lastname: "Last Name",
6       company: "Company",
7       email: "Email",
8       industry: "Industry",
9       hospitality: "Hospitality",
10      tech: "Tech",
11      submit: "Save now",
12      language: "Language",
13      formErrors: "Please correct the errors below:",
14      fieldError: "This field is required",
15      thankyou: "Thanks for submitting!"
16    }
17  },
18 ▾ 'fr': {
19   i18n: {
20     firstname: "Prénom",
21     lastname: "Nom de famille",
```



Javascript to pull it all together

1. Replace square brackets with curly brackets
2. Determine current language
 - a. Did Prospect already specify?
 - b. Did Form previously specify?
 - c. Does Browser specify?
3. Read HTML (handlebars.js template)
4. Apply translations



Javascript to pull it all together

1. Page / Form Display
2. Error Messages
3. Thank you page



Getting the Code

Sample code is available on

THE SPOT
by SERCANTE



Thank You!

