FORM BUILDER QUICK START GUIDE

1. INTRO

The form builder is a tool to style and structure your <u>dynamic forms</u> without any HTML, CSS or Javascript knowledge. It also allows you to set up interactive logic, for example hiding/showing fields, validation, or something like adding a date picker etc. The following document will provide you a guide on how to use this tool.

2. INITIALIZATION

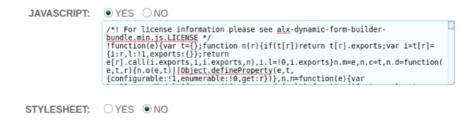
The form builder is bundled and minified javascript code that takes in a configuration (provided by you) and uses it to do it's "magic" under the hood. In order to make it part of the form you have to navigate to the dynamic form configuration and copy-paste it to the "Embedded assets – Javascript" section.

In the example shown in this quick tutorial the form is a skinny version of a sublease submission, where there are three Targets. Doctors, Stores and the Sublease (which is the Relational Target).

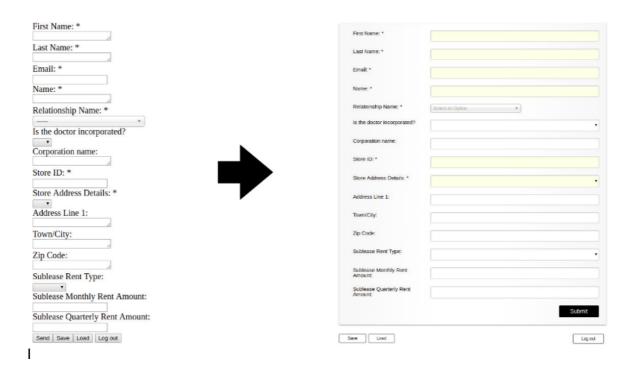
IMPORTANT!

Make sure you copy-paste the entire code. If anything is missing or you modify the code the form builder will not work.

Embedded assets



Save the settings and if you open the dynamic form you should see the form builder being initialized and applying the default styling and layout.



3. CONFIGURATION BASICS

The configuration tells the form builder what to do. It is a set of rules that gets interpreted by the form builder to set up the structure, the styling and the form's internal logic.

It also gets inserted into the "Embedded assets – Javascript" section, underneath the code that you added in the previous step.

Embedded assets

```
JAVASCRIPT:

• YES • NO

visible;\n opacity: 1; \\n",""])}]);
//# sourceMappingURL=data:application/json;charset=utf-
8;base64,eyJ2ZXJzaW9uIjozLCJzb3VyY2VzIjpbXSwibmFtZXMioltdLCJtYXBwaW5ncyI6IiIs
ImZpbGUi0iJhbHgtZHluYW1pYy1mb3JtLWJ1aWxkZXItYnVuZGxlLm1pbi5qcyIsInNvdXJjZVJvb
3Qi0iIifQ==
| window.alx_dynamic_form_config = {}

STYLESHEET: • YES • NO
```

An empty configuration (the equivalent of providing no configuration) would look like this.

```
window.alx_dynamic_form_config = {}
```

This is the Javascript Object which we are going to gradually extend to configure our form.

What is a Javascript Object?

Objects in JavaScript can be compared to objects in real life. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. A cup described as a Javascript Object could look something like this.

```
cup = {
  color: {
    handle: 'black,
    cup: 'white
  },
  design: 'ornamentic',
  material: 'porcelain',
  owners: ['Flavius', 'Rachel']
}
```

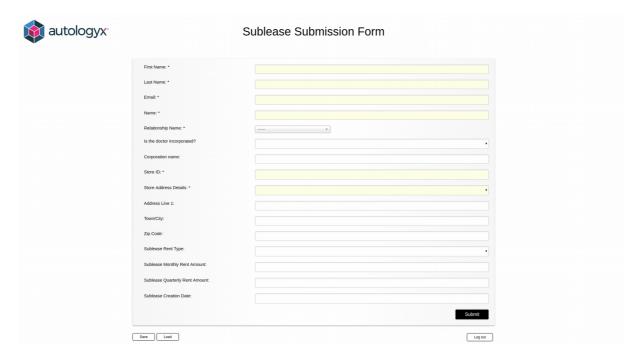
4.0. BASIC FORM EDITING

4.1. ADDING LOGO AND TITLE

As a first step let's enhance our form by adding a logo and a title to it by changing the configuration.

```
window.alx_dynamic_form_config = {
    content: {
       page: {
            logo: "https://www.autologyx.com/sites/default/files/ALX_logonostrap-R-350x96.png",
            title: "Sublease Submission Form"
       }
    }
}
```

The above configuration will yield the following result:



4.2. CHANGING FIELD LABELS

Modifying fields, like changing the label, adding a placeholder, or just changing the field type is quite a common task. The form builder gives you quick access through the configuration to do all of these. First, let's change the label for individual fields.

And if you want to add the same label to several fields at once you only have to do the following.

```
window.alx_dynamic_form_config = {
    content: {
       page: {
           logo: "https://www.autologyx.com/sites/default/files/ALX_logonostrap-R-350x96.png",
           title: "Sublease Submission Form"
        single: {
            fields: {
               id_doctor_field_corporation_description: {
                   label: "Description"
                id_field_sublease_creation_date: {
                   label: "Creation Date'
           }
        multiple: {
            fields: {
               labels: [{
    value: "Name",
                        "id_doctor_field_firstname",
                        "id_test_store_field_name"
                        "id_doctor_field_corporation_name"
                }, {
                    value: "Id",
                        "id_test_store_field_store_id"
               }]
           }
  }
```

IMPORTANT!

As you can see you target single fields by using the "single" property and multiple fields at once with the "multiple" property.

Single fields:

First you tell the form builder which field you want to modify (using its id) and then you tell it what logic apply to the field.

Multiple fields:

First you tell the form builder what is the logic you want to apply, then you list the fields you want the logic to be applied to.

Also notice, that when you are defining multiple labels the data structure the form builder is expecting is a list of items (array).

Now we got some of the labels changed.

lyx° Sublease Submission Form

_ast Name: *		
Email: *		
Name *		
Relationship Name: *		
s the doctor incorporated?		
Name		
Description		
Store ID: *		
Store Address Details: *		
Address Line 1:		
own/City:		
tip Code:		
Sublease Rent Type:		
Sublease Monthly Rent Amount:		
Sublease Quarterly Rent Amount:		
Creation Date		
		Submi

4.3. CHANGING FIELD PLACEHOLDERS

Changing placeholders of fields is almost identical to changing the labels. All you have to do is use the "placeholder" keyword instead of "label". First let's add a placeholder to one of the fields.

```
window.alx_dynamic_form_config = {
   content: {
      page: {
            logo: "https://www.autologyx.com/sites/default/files/ALX_logonostrap-R-350x96.png",
            title: "Sublease Submission Form"
      },
      single: {
            fields: {
                id_doctor_field_corporation_description: {
                    label: "Description"
            },
            id_field_sublease_creation_date: {
                    label: "Creation Date"
            },
            id_test_store_field_address_line_1: {
                    placeholder: "This is a placeholder for the address"
            }
        }
        multiple: {/* config */}
    }
}
```

And then change the placeholder for some of the name fields where we changed the label.

```
window.alx_dynamic_form_config = {
   content: {
       page: {
           logo: "https://www.autologyx.com/sites/default/files/ALX logonostrap-R-350x96.png".
           title: "Sublease Submission Form"
        single: {
           fields: {
              id_doctor_field_corporation_description: {
                   label: "Description"
               id_field_sublease_creation_date: {
                   label: "Creation Date"
               id_test_store_field_address_line_1: {
                   placeholder: "This is a placeholder for the address"
           }
        multiple: {
           fields: {
              labels: [{
                   value: "Name",
                   ids: [
                        "id_doctor_field_firstname",
                       "id_test_store_field_name"
                       "id_doctor_field_corporation_name"
               }, {
                   value: "Id",
                   ids: [
                        "id_test_store_field_store_id"
               placeholders: [{
                   value: "Placeholder for name fields",
                   ids: [
                       "id_test_store_field_name",
                       "id_doctor_field_corporation_name"
 } }
              }]
```

IMPORTANT!

Placeholders are only applied to fields that accept a text as an input.

Now with these changes we added some placeholder texts too.

јух°

Sublease Submission Form

Name *		
Last Name: *		
Email: *		
Name *	Placeholder for name fields	
Relationship Name: *		
Is the doctor incorporated?		
Name	Placeholder for name fields	
Description		
id *		
Store Address Details: *		
Address Line 1:	This is a placeholder for the address	
Town/City:		
Zip Code:		
Sublease Rent Type:		
Sublease Monthly Rent Amount:		
Sublease Quarterly Rent Amount:		
Creation Date		
	Su	ıbmit
	30	DTIIL

4.4. ADDING HELPER TEXTS TO FIELDS

One fairly common task whie modifying the forms is to add helper texts underneath the fields that guide the user how to fill the form. Using the same principle we just learned with the labels and placeholders we can quickly add helper texts. Let's add some helper text to the description field so we can see how applying multiple logic to an individual field looks like.

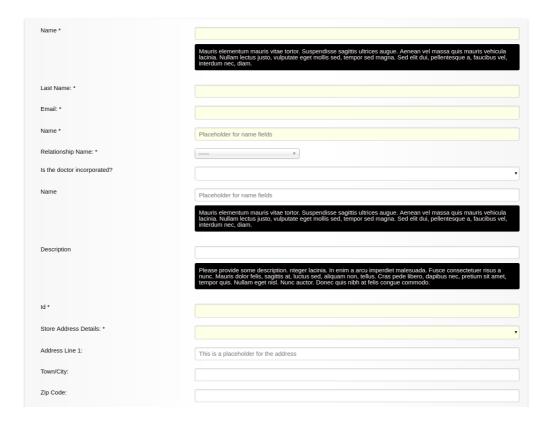
Same logic as we have seen it before for adding multiple helper texts.

```
window.alx_dynamic_form_config = {
   content: {
       page: {
            logo: "https://www.autologyx.com/sites/default/files/ALX_logonostrap-R-350x96.png",
            title: "Sublease Submission Form"
        single: {
            fields: {
               id_doctor_field_corporation_description: {
                   label: "Description"
                    helperText: "Please provide some description. nteger lacinia. In enim a arcu imperdiet malesuada. Fusce conse
                id_field_sublease_creation_date: {
                   label: "Creation Date
                id_test_store_field_address_line_1: {
                   placeholder: "This is a placeholder for the address"
           }
        multiple: {
            fields: {
                labels: [{
                   value: "Name",
                    ids: [
                         "id_doctor_field_firstname",
                        "id_test_store_field_name"
                        "id_doctor_field_corporation_name"
                    1
                    value: "Id",
                   ids: [
"id_test_store_field_store_id"
                    ]
                placeholders: [{
   value: "Placeholder for name fields",
                        "id_test_store_field_name",
                        "id_doctor_field_corporation_name"
                helperTexts: [{
                    value: "Mauris elementum mauris vitae tortor. Suspendisse sagittis ultrices augue. Aenean vel massa quis maur
                    ids: [
                        "id_doctor_field_firstname",
                        "id_doctor_field_corporation_name"
                    ]
               }]
     } }
   }
}
```

With the above changes this is the result we get.



Sublease Submission Form

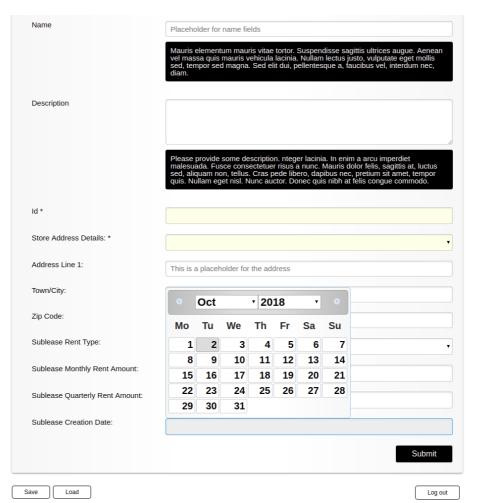


4.5 CHANING FIELD TYPE

By default a dynamic form has a limited set of field types, but what if we need a multi-line textarea or a date picker for a field? First let's add the changes.

```
window.alx_dynamic_form_config = {
    content: {
       page: {
           logo: "https://www.autologyx.com/sites/default/files/ALX_logonostrap-R-350x96.png",
            title: "Sublease Submission Form"
        single: {
            fields: {
                id_doctor_field_corporation_description: {
                    label: "Description", helperText: "Please provide some description. nteger lacinia. In enim a arcu imperdiet malesuada. Fusce consect
                    type: "textarea"
                id_field_sublease_creation_date: {
                    label: "Creation Date"
                id_test_store_field_address_line_1: {
                    placeholder: "This is a placeholder for the address"
                id_field_sublease_creation_date: {
                    type: "date"
        multiple: {/* config */}
```

And below you can see how the end results look like.



4.6 ADVANCED FIELD CONFIGURATION

Now lets stop here for a moment. In the above examples we were able to add quick changes, but soon we would run into certain limitations. What if we wanted to add more rows to textarea fields, what if we wanted to date field to use US rather than UK locale date. What about adding hyperlinks to helper texts?

The form builder allows you to do "finer" configuration. Take a look at the configurations below.

```
window.alx_dynamic_form_config = {
   content: {
   page: {/* config */},
        single: {
           fields: {
               id_doctor_field_corporation_description: {
                    label: "Description",
                    helperText: "Please provide some description. nteger lacinia. In enim a arcu imperdiet malesuada. Fusce consec
                    type: "textarea"
                id field sublease creation date: {
                   label: "Creation Date"
                id test store_field_address_line_1: {
                  placeholder: "This is a placeholder for the address"
               id_field_sublease_creation_date: {
                   type: "date"
           }
        multiple: {/* config */}
```

```
window.alx_dynamic_form_config = {
   content: {
       page: {/* config */},
        single: {
            fields: {
                id doctor field corporation description: {
                    label: "Description",
                    helperText: {
                        value: "Please provide some description. nteger lacinia. In enim a arcu imperdiet malesuada. Fusce consecte
                    type: {
                        value: "textarea",
                        config: {
    "rows": 5
                   }
                id_field_sublease_creation_date: {
                   label: "Creation Date"
                id test store_field_address_line_1: {
                    placeholder: "This is a placeholder for the address"
                id_field_sublease_creation_date: {
                        value: "date",
                        config: {
                           region: "en-GB"
                   }
               }
           }
        multiple: {/* config */}
```

These two configurations provide the same result. The difference is that this time we provided "sub-configuration objects" instead of simple texts. You will learn more about that in the advanced section, but for now all you have to know that this is the way to do things, like injecting HTML content to the form.

5.0 GROUPING FIELDS TOGETHER

Now let's do something a little bit more advanced. The form builder allows us to put fields into logical groups (or nest groups within groups). This will give us access to powerful features like pagination, or showing/hiding multiple fields at once.

If you remember here in this tutorial we have three Targets. Doctors, Stores and the Sublease (which is the Relational Target).

```
window.alx_dynamic_form_config = {
    structure: {
       groups: [{
            id: "doctor",
            fields: [
                "id_doctor_field_firstname",
                "id_doctor_field_lastname",
                "id doctor field email",
                "id\_doctor\_field\_is\_the\_doctor\_incorporated"
            children: [{
               id: "doctor-corporation",
                fields: [
                    "id_doctor_field_corporation_name",
                    "id_doctor_field_corporation_description"
           }]
        }, {
            id: "store",
           fields: [
               "id_test_store_field_name",
                "id_test_store_field_store_id",
                "id_test_store_field_store_address_details"
            children: [{
   id: "store-address",
                fields: [
                    "id test store field address line 1".
                    "id_test_store_field_towncity",
                    "id_test_store_field_zip_code
            }]
        }, {
            id: "sublease",
            fields: [
               "id_field_relationship_name",
               "id_field_sublease_rent_type",
                "id_field_sublease_monthly_rent_amount",
                "id_field_sublease_quarterly_rent_amount",
                "id_field_sublease_creation_date"
       }]
    content: {/* config */}
}
```

This setting on the surface will not change the form, but under the hood quite a few things happened. We told the form builder that we have five groups, "sublease", "doctor", "store", "store-address" and "doctor-corporation". Two of which are nested groups.

We also defined which fields belong to which group by using their ids.

IMPORTANT!

When you define groups you have to allocate all the fields, otherwise they will "fall out" from the form.

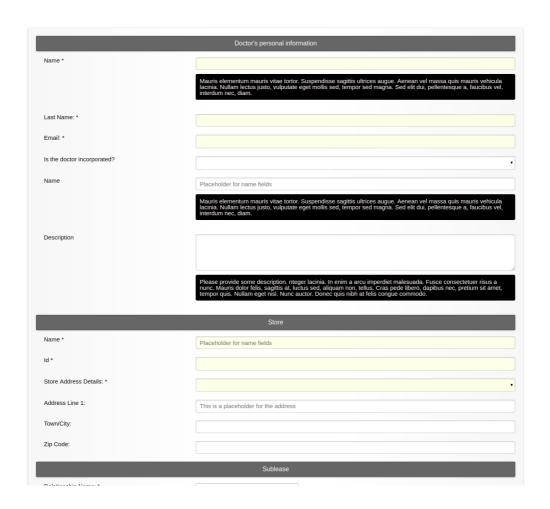
5.1. ADDING HEADINGS TO GROUPS

Now that we have groups of fields we can label them appropriately. Let's do that by adding some headings.

As you can see the logic is pretty similar to how we would configure our fields if we wanted to inject related content.

УX°

Sublease Submission Form



5.2. ADDING DESCRIPTIONS TO GROUPS

Now let's add some descriptions to some of the groups.

Now we managed to add some description to the groups.



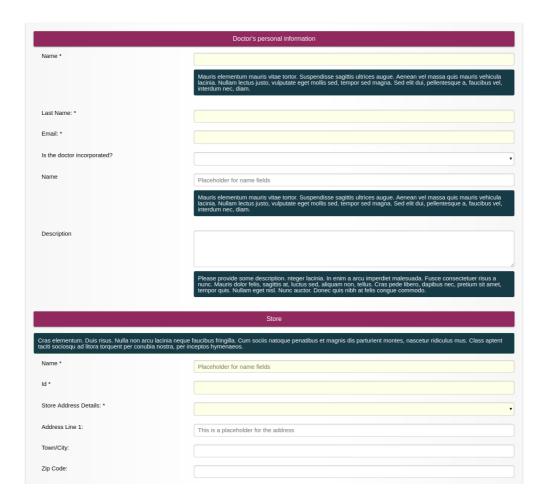
6.0. CHANGING COLOURS

Now that we added some content and our form started taking shape we can see that by default ther form builder uses variations of grey and black to colour the form. The grey is the primary and the black is the secondary by default. If we want to change that we only have to do the following.

After defining the primary and secondary colours all of that gets applied to all the elements throughout the form to get the following result.

JYX°

Sublease Submission Form



IMPORTANT!

When it comes to text the form builder runs a check against the background colour and picks white or black dependent on which one is the most contrasting.

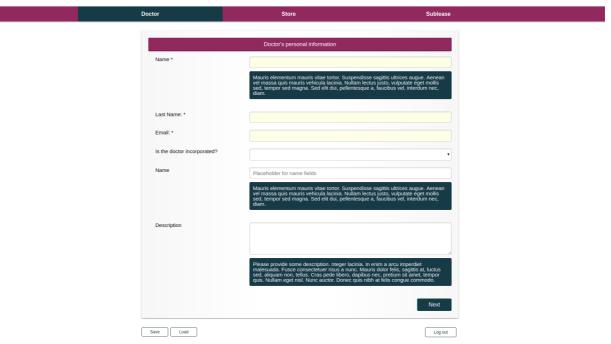
7.0 PAGINATION

An other really powerful feature of the form builder is to have the ability to split long forms into several pages. After generating the groups adding pagination is very simple.

```
window.alx_dynamic_form_config = {
    styling: /* config */,
    structure: {
        groups: {/* config */},
        pages: [{
            label: "Doctor",
            id: "doctor"
        }, {
            label: "Store",
            id: "store"
        }, {
            label: "Sublease",
            id: "sublease"
        }]
    },
    content: {/* config */}
```



Sublease Submission Form



IMPORTANT!

When you enable pagination you have to include all the top-level groups in your configuration when you label them.