Adapt: dynamic forms

Table of Contents

Component JSON schemas

These tables describe the JSON schemas for each individual component, providing example values, types, descriptions, whether properties are optional and their default values.

<input>

Descriptions for all `<input>` types available in Adapt.

input:text

Prop	Example Value	Type	Description	Optional	Default
type	"input:text"	String	Type of rendered input		
label	"Username"	String	Text for label	Yes	""
state	["State1"]	String|Array	State for when to show component	Yes	All States
placeholder	"Enter a username"	String	Text for placeholder	Yes	""
description	"Username will be your login"	String	Description/help text	Yes	""

input:date

Prop	Example Value	Type	Description	Optional	Default
type	input:date	String	Type of rendered input		
label	"Shipping date"	String	Text for label	Yes	""
state	["State1"]	String|Array	State for when to show component	Yes	All States
placeholder	"Enter your DOB"	String	Text for placeholder	Yes	""
description	"Select a date for shipping orders"	String	Description/help text	Yes	""

input:checkbox

Prop	Example Value	Type	Description	Optional	Default
type	input:checkbox	String	Type of rendered input		
label	"Shipping date"	String	Text for label	Yes	""
state	["State1"]	String|Array	State for when to show component	Yes	All States
description	"Select a date for shipping orders"	String	Description/help text	Yes	""
options	{ optionOne: "Option One" }	Object	Object with key/value pairs for each checkbox	Yes	""

input:radio

Prop	Example Value	Type	Description	Optional	Default
type	input:date	String	Type of rendered input		
label	"Shipping date"	String	Text for label	Yes	""
state	["State1"]	String|Array	State for when to show component	Yes	All States
description	"Select a date for shipping orders"	String	Description/help text	Yes	""
options	{ optionOne: "Option One" }	Object	Object with key/value pairs for each radio item	Yes	""

<select>

Descriptions for all `<select>` types available in Adapt.

select

Renders a single select dropdown component

Component JSON

Prop	Example Value	Type	Description	Optional	Default
type	select	String	Type of rendered input		
options	[{}]	Array of objects		Yes	""
state	["State1"]	String|Array	State for when to show component	Yes	All States
label	*	String	Text for label	Yes	' '
placeholder	*	String	Text for placeholder	Yes	' '
description	*	String	Description/help text	Yes	' '

Options JSON

Prop	Example Value	Type	Description	Optional	Default
value	1	String	Value of select item	No	
label	One	String	Label to display select item	No	

select:multiple

Renders a multiple select component.

Component JSON

Prop	Example Value	Type	Description	Optional	Default
type	select:multiple	String	Type of rendered input		
label	*	String	Text for label	Yes	' '
options	[{}]	Array of objects		Yes	" "
state	["State1"]	String|Array	State for when to show component	Yes	All States
placeholder	*	String	Text for placeholder	Yes	' '
description	*	String	Description/help text	Yes	' '

Options JSON

Prop	Example Value	Type	Description	Optional	Default
value	1	String	Value of select item	No	
label	One	String	Label to display select item	No	

<textarea>

Renders a textarea

Prop	Example Value	Type	Description	Optional	Default
type	textarea	String	Type of rendered input		
state	["State1"]	String|Array	State for when to show component	Yes	All States
label	*	String	Text for label	Yes	' '
placeholder	*	String	Text for placeholder	Yes	' '
description	*	String	Description/help text	Yes	' '

<button>

Renders a textarea

Prop	Example Value	Type	Description	Optional	Default
type	button	String	Type of rendered input		
text	Submit!	String	Text for button		''
state	["State1"]	String|Array	State for when to show component	Yes	All States
label	Submit the form	String	Text for label	Yes	''
description	Press this button to submit	String	Description/help text	Yes	''

Element JSON schemas

These tables describe the JSON schemas for each individual component, providing example values, types, descriptions, whether properties are optional and their default values.

Column

Descriptions for all `column` types available in Adapt.

column

Prop	Example Value	Type	Description	Optional	Default
type	"column"	String	Type of rendered element		
items	[{}, {}]	Array	Array of objects for each column		""
state	["State1"]	String|Array	State for when to show component	Yes	All States
span	[3, 1]	String or Array	Width for each column		

The span property can be either an array or string. If all columns are the same width, you can use a string, but if you have columns with varying widths, use an array. If the array does not have values for each column, it will be repeated - i.e. for 4 columns, `[2, 3]` would be

the equivilent of `[2, 3, 2, 3]`

column:rows

Instead of having fixed columns, these are purely rows at whatever width specified, allowing you to have dynamically added fields display correctly on the page.

Prop	Example Value	Type	Description	Optional	Default
type	"column:rows"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
items	{}	Object	Object of items		""
span	[3, 1]	String or Array	Width for each column		

Label

The label item. This is mainly used by fields, but you can use it if you wish.

Prop	Example Value	Type	Description	Optional	Default
type	"label"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
label	"Hello"	String	The label's text		

Description

The description item. This is mainly used by fields, but you can use it if you wish.

Prop	Example Value	Type	Description	Optional	Default
type	"label"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
desc	"Hello"	String	The label's text		

Accordion

The accordion item.

Prop	Example Value	Type	Description	Optional	Default
type	"accordion"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
model	{}	Object	The model view for each accordion created		
title	"Accordion"	String	The title for each accordion		
accordionTitle	"Item {index}"	String	The title for each accordion		

With the `accordionTitle` property, you can use `{variableName}` to display a value within that accordion. If you wish to display the accordion number, use `{index}`.

Header

The header item. This is mainly used by components, but you can use it if you wish.

Prop	Example Value	Type	Description	Optional	Default
type	"header:h1"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
text	"Hello"	String	The titles text		

You can use `header:h1` all the way through to `header:h6`.

Tabcordion

The tabcordion item.

Component JSON

Prop	Example Value	Type	Description	Optional	Default
type	"tabcordion"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
items	{}	Object	The object of items to display		

Items JSON

Prop	Example Value	Type	Description	Optional	Default
tabType	"tab"	String	Type of tab, either tab or accordion		
state	["State1"]	String|Array	State for when to show component	Yes	All States
title	"Processes"	String	The title of the tab		
items	{}	Object	The object of items to in the tab		
addText	"Add Item"	String	The wording for the button to add a new accordion item		
accordionTitle	"Item {index}"	String	The title for each accordion		
accordionSubtitle	""	String	The subtitle for each accordion		

With the `accordionTitle` and `accordionSubtitle` properties, you can use `{variableName}` to display a value within that accordion. If you wish to display the

accordion number, use `{index}` .

HR

The hr item, used to split sections of the form up.

Prop	Example Value	Type	Description	Optional	Default
type	"hr"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States

Table

The table item.

Prop	Example Value	Type	Description	Optional	Default
type	"table"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
model	{}	Object	The model view for each table row created		

The table will create an index column for each item. For each column header, it will go through the model provided and use the `label` property. Labels and descriptions on component within the model object will not be displayed above/below the component.

Tabs

The tabs item.

Component JSON

Prop	Example Value	Type	Description	Optional	Default
type	"table"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show component	Yes	All States
items	{}	Object	Items object containing each tab		

Items JSON

Prop	Example Value	Type	Description	Optional	Default
type	"table"	String	Type of rendered element		
state	["State1"]	String|Array	State for when to show tab	Yes	All States
title	"Tab One"	String	The title of the tab		
items	{}	Object	Items for the tab content		

The table will create an index column for each item. For each column header, it will go through the model provided and use the `label` property. Labels and descriptions on component within the model object will not be displayed above/below the component.