MyTooth Forms

Complete User Guide

Table of Contents

1. Introduction to Forms

MyTooth allows you to create digital forms for your practice. These forms can be used for:

- New patient registration
- Patient demographic information
- Consent forms
- Treatment plans
- General questionnaires and acknowledgments

Forms can be filled out by staff in the office or sent to patients to complete remotely.

2. Understanding Form Types

When creating a form, you'll need to select a **Form Type**. This helps organize your forms and ensures data is stored correctly.

Form Type	When to Use It	Examples
Medical Alerts	For forms capturing patient allergies, conditions, or alerts that must appear prominently in the patient chart	Allergy information, medical conditions, health alerts
Medical Questions	For forms gathering patient medical history, medications, or systemic health information	Health history questionnaire, current medications, medical conditions
Dental Questions	Use for dental health questionnaires — for example, forms that gather information about oral hygiene, previous dental treatments, or current dental issues	Oral hygiene habits, previous dental treatments, current dental issues
Patient Demographic	For patient information and registration details	Contact info, emergency contacts, insurance

Form Type	When to Use It	Examples
Other	Most general forms	HIPAA forms, treatment consents, office policies, financial agreements, questionnaires

Note: Most forms you create will use the "Other" type unless you're specifically collecting patient demographic data.

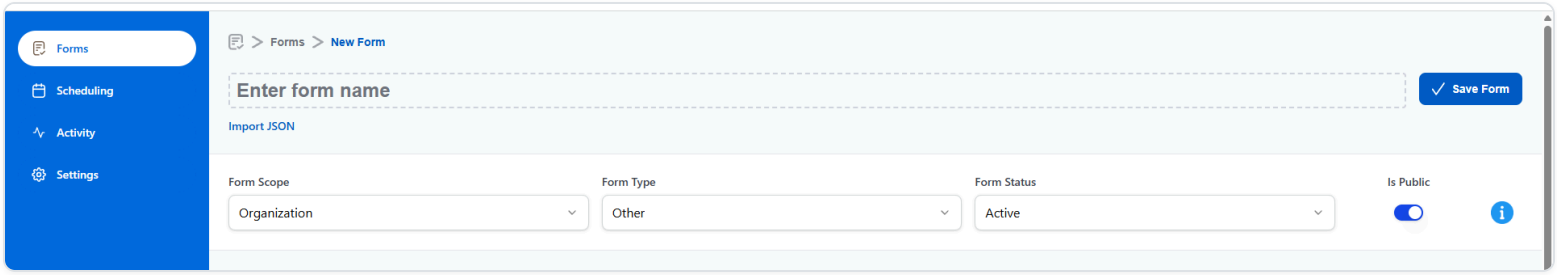
3. Getting Started

Opening the Form Builder

1. Log into MyTooth
2. Click on **Forms** in the left menu
3. Click **New Form**
4. Click **Build**

Setting Up Your Form

Before you start building, you'll need to configure a few basic settings:



Form configuration settings panel

Form Name

Give your form a clear name that describes its purpose.

> **Good examples:** "New Patient Welcome Form", "Treatment Consent", "HIPAA Acknowledgment"

> **Avoid vague names like:** "Form 1", "Test", "New Form"

Form Scope

Choose where this form will be available:

- **Organization:** Available at all your practice locations
- **Location:** Only available at specific locations

> **Recommendation:** Use "Organization" for forms that apply to all locations.

Form Type

Select from the types shown in the table above. When in doubt, choose "Other".

Status

- **Active:** Form is ready to use
- **Draft:** Form is still being built
- **Inactive:** Form is hidden but saved

Start with "Draft" while building, then change to "Active" when ready.

Is Public

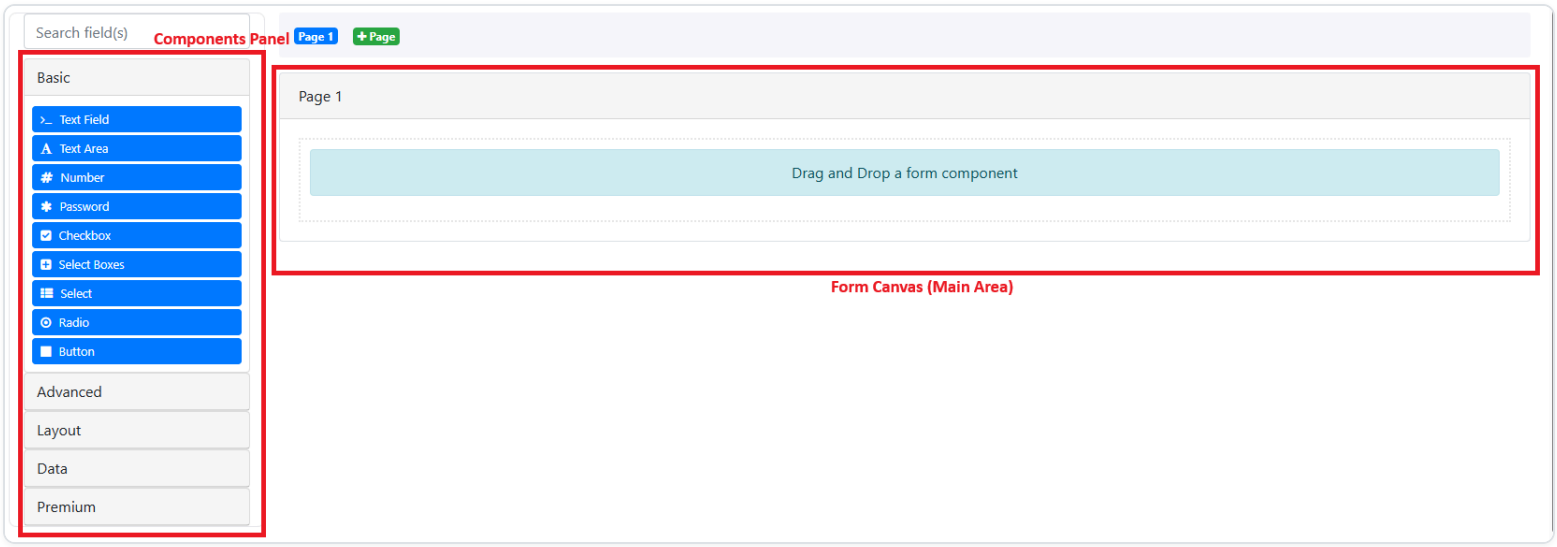
Turn this ON if patients should be able to access and fill out this form remotely (via a link you send them).

4. Building Your First Form

Understanding the Screen

When you open the form builder, you'll see two main areas:

1. **Components Panel (Left Sidebar):** Contains all drag-and-drop fields such as Text Field, Number, Select, Checkbox, Radio, File Upload, etc.
2. **Form Canvas (Main Area):** The workspace where you drag and drop components to build your form.



Form builder interface showing Components Panel and Form Canvas

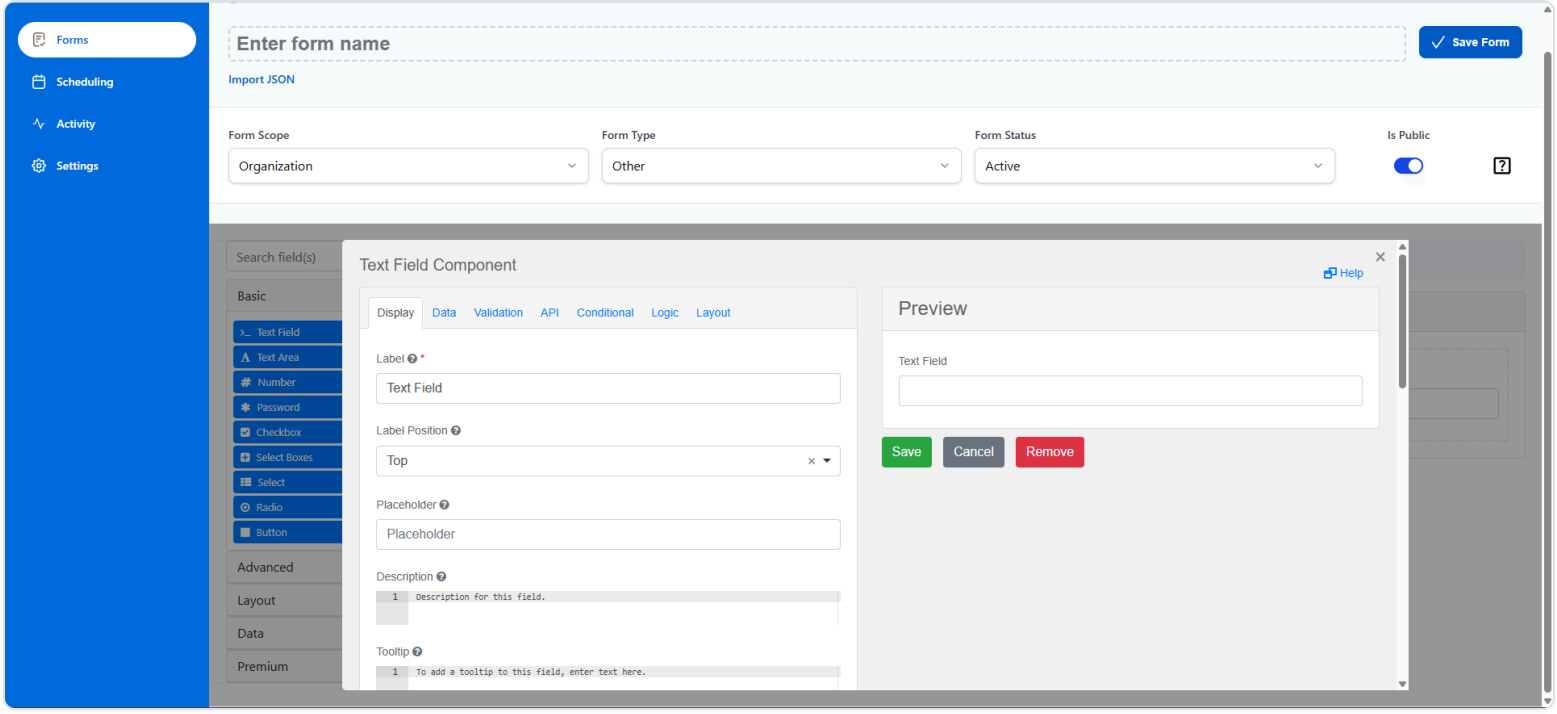
Adding Your First Field

Let's add a simple text field for a patient's name:

1. On the left panel, find **Text Field** under the "Basic" section
2. Click and drag it to the center canvas
3. Drop it where you want it to appear
4. A settings panel will open on the right

Customizing the Field

In the settings panel, you'll see several tabs. The most important one is **Display**:



Component settings panel with Display tab options

- **Label:** Type "Patient's First Name" - this is what users will see
- **Placeholder:** Type "Enter your first name" - this shows inside the empty field
- **Description:** Add any helpful instructions (optional)

Click **Save** at the bottom.

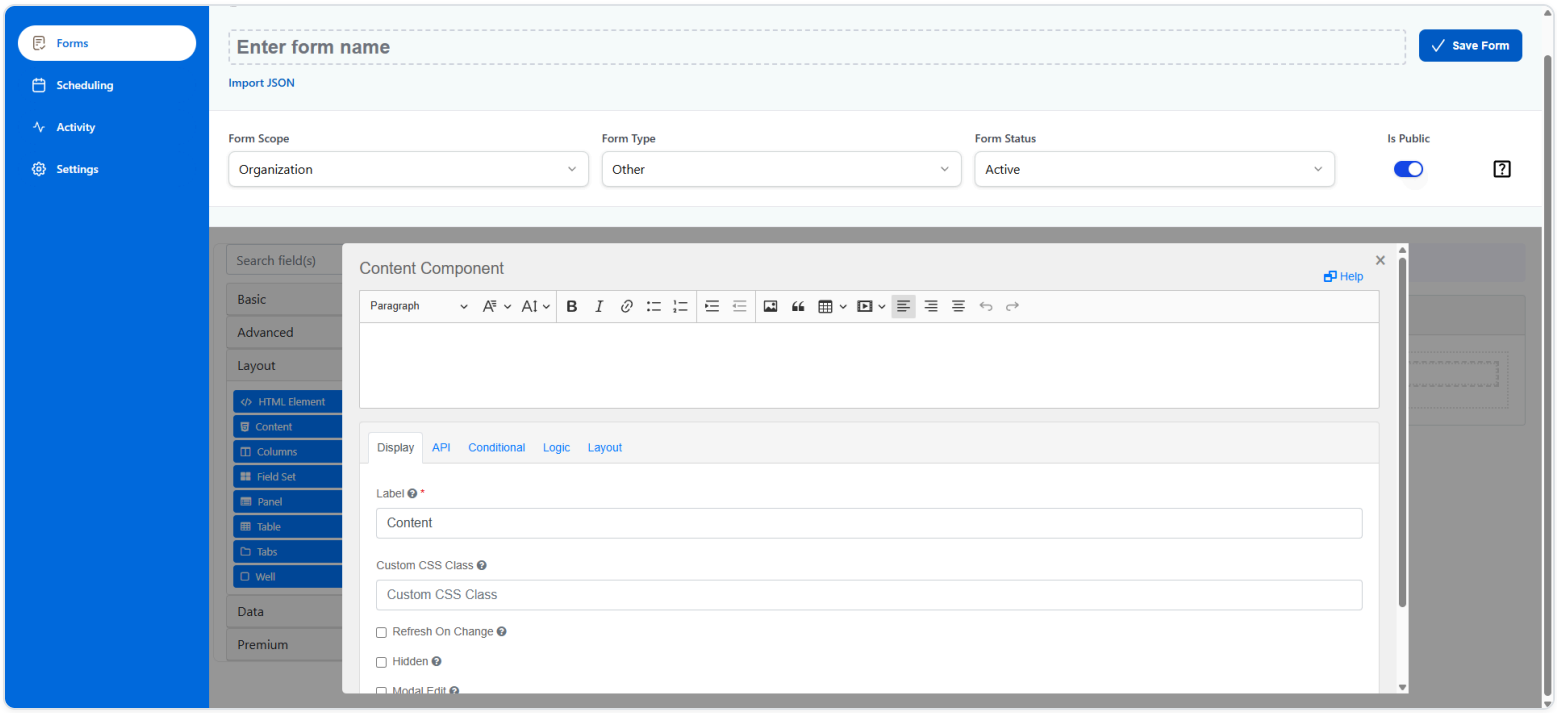
> **Congratulations!** You've added your first field. Continue adding more fields following the same process.

Adding Headers, Paragraphs, and Text Content

To organize your form with headers and add instructional text, use the **Content** field. This versatile component allows you to add section headers, normal paragraphs, or any descriptive text to guide users through your form.

1. Find **Content** under "Layout" in the left panel
2. Drag it to your form
3. Use the text editor to type a header like "Personal Information" or add a paragraph of instructions
4. Format your text using the formatting toolbar:
    - **Bold** your text for emphasis
    - Make text *larger or smaller* for hierarchy
    - **Center align** text for headers
    - Add **bullet points** or numbered lists

5. Click **Save**



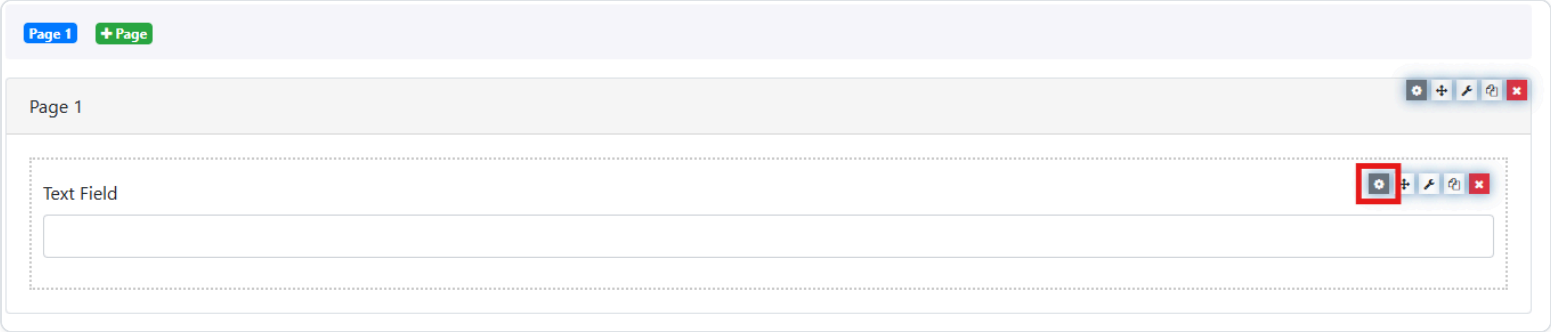
Content component with text editor and formatting options

> **Tip:** Use the Content field to add normal paragraphs with instructions like "Please fill out all required fields" or "This information will be kept confidential" throughout your form.

5. Working with Fields

Making Changes to a Field

Click on any field in your form to edit it. The settings panel will open on the right.



Click on a field to open the settings panel for editing

Important Field Settings

Display Tab

Label: The title of your field

```
Example: "Home Phone Number"
```

Placeholder: Helper text inside the field

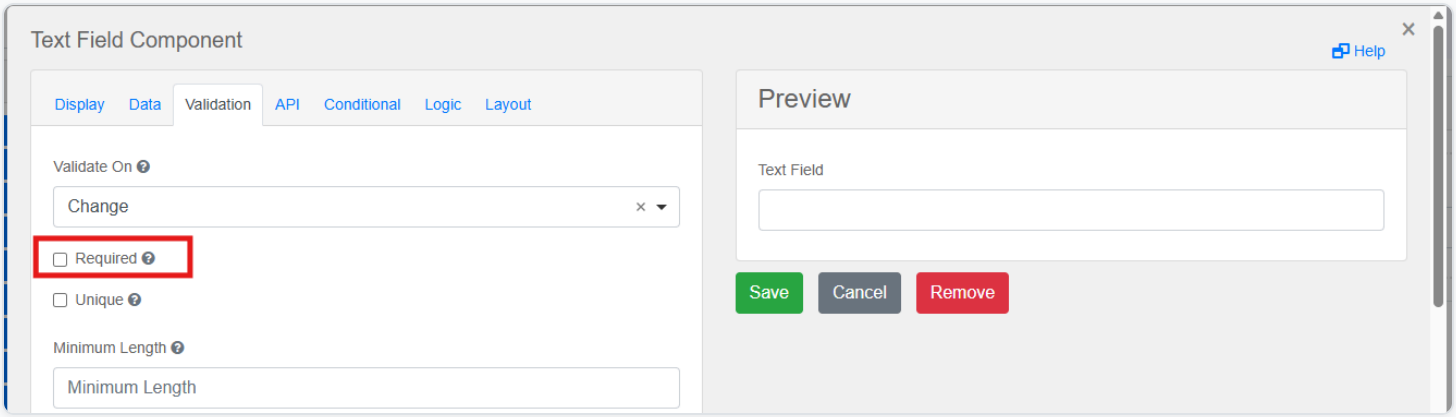
```
Example: "555-123-4567"
```

Description: Additional instructions below the field

```
Example: "Please include area code"
```

Disabled: Turn ON to make a field read-only (users can see it but can't change it). This option can be found at the bottom of the Display tab.

Validation Tab



Validation tab with Required, length limits, and error message options

Required: Turn ON to make this field mandatory. Users can't submit the form without filling it out.

Minimum/Maximum Length: Set limits on how many characters can be entered

Error Message: What users see if they fill it out incorrectly

```
Example: "Please enter a valid phone number"
```

API Tab

Property Name: A unique identifier for this field (required for data to save correctly)

- Use simple names without spaces
- Examples: `firstName` , `homePhone` , `dateOfBirth`

Important: In order to pre-fill patient data automatically, use these exact property names for the following fields:

- `patientFirstName` - for Patient First Name
- `patientLastName` - for Patient Last Name
- `patientDOB` - for Patient Date of Birth

Using these exact property names enables automatic pre-filling of patient information when patients verify their identity.

Rearranging Fields

You can move fields around by clicking and dragging them to a new position on your form.

Deleting Fields

To delete a field, first click on it to select it, then press the **Delete** button on your keyboard.

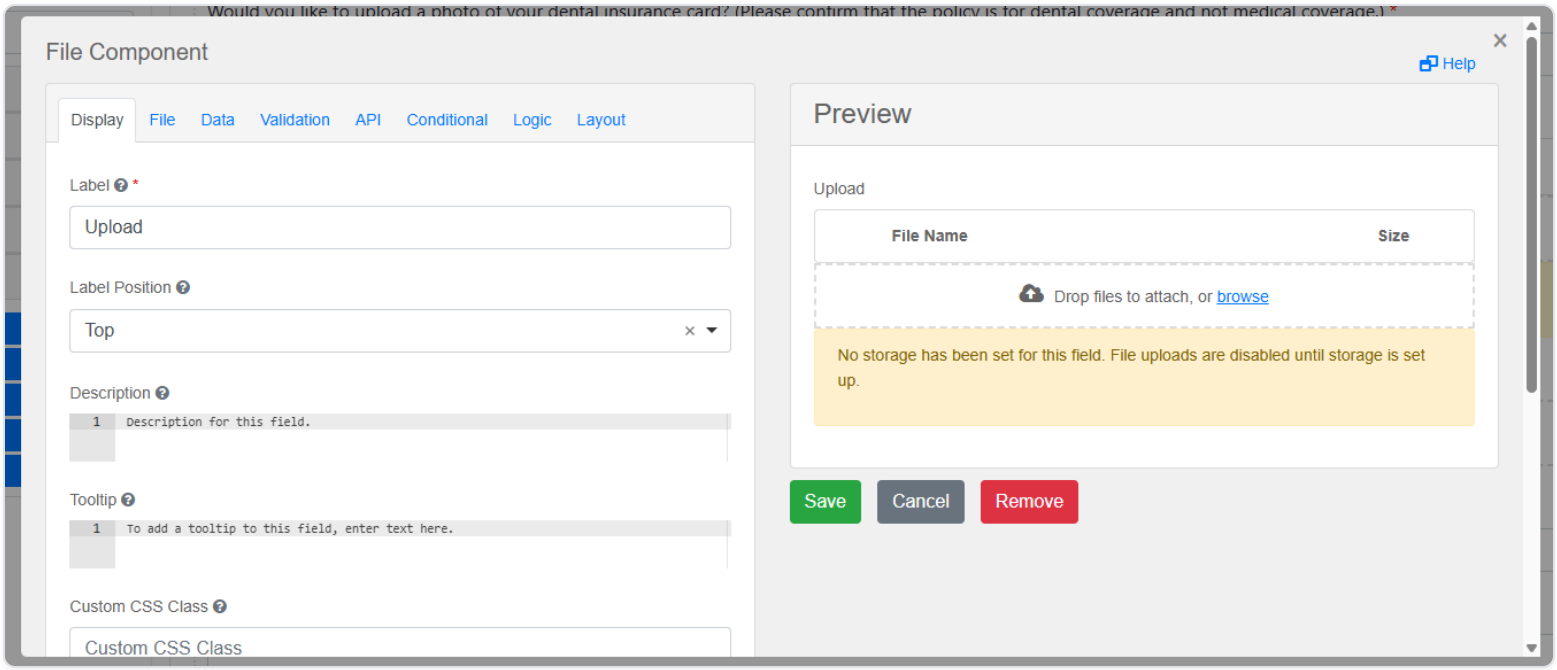
Adding a File Upload Field

To add a file upload field (for insurance cards, ID photos, X-rays, etc.), follow these steps:

Step 1: Add the File Component

Drag and drop **File** from the **Premium** section on the left panel to your form.

The File Component modal will open:



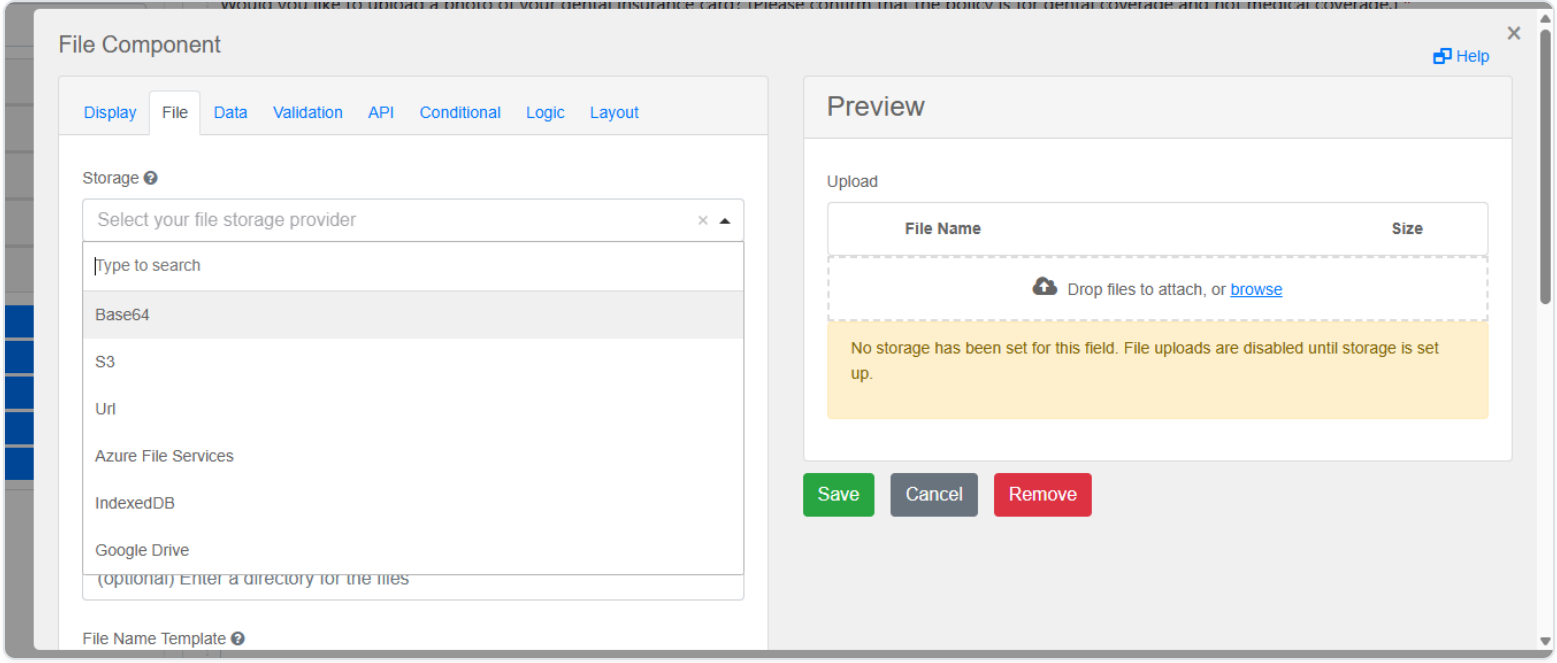
File Component modal with basic settings

From here, you can configure basic information like:

- **Label:** Give your field a descriptive name (e.g., "Upload Insurance Card")
- **Position:** Choose where the label appears

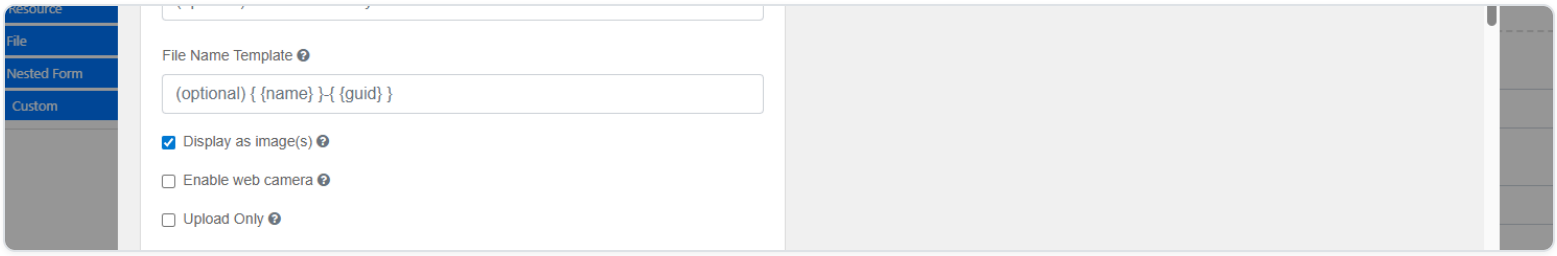
Step 2: Configure the File Tab

Click on the **File** tab to access storage and display options:



File tab with storage and display options

1. Set the **Storage** option to **Base64**
2. Check the **"Display as image(s)"** checkbox - this will show the uploaded image on the submitted PDF form



Display as image(s) checkbox selected

Step 3: Save Your Field

Click the **Save** button. Your form will now accept image uploads!

> **Result:** When patients upload images (like insurance cards or IDs), the images will be displayed directly on the submitted PDF form.

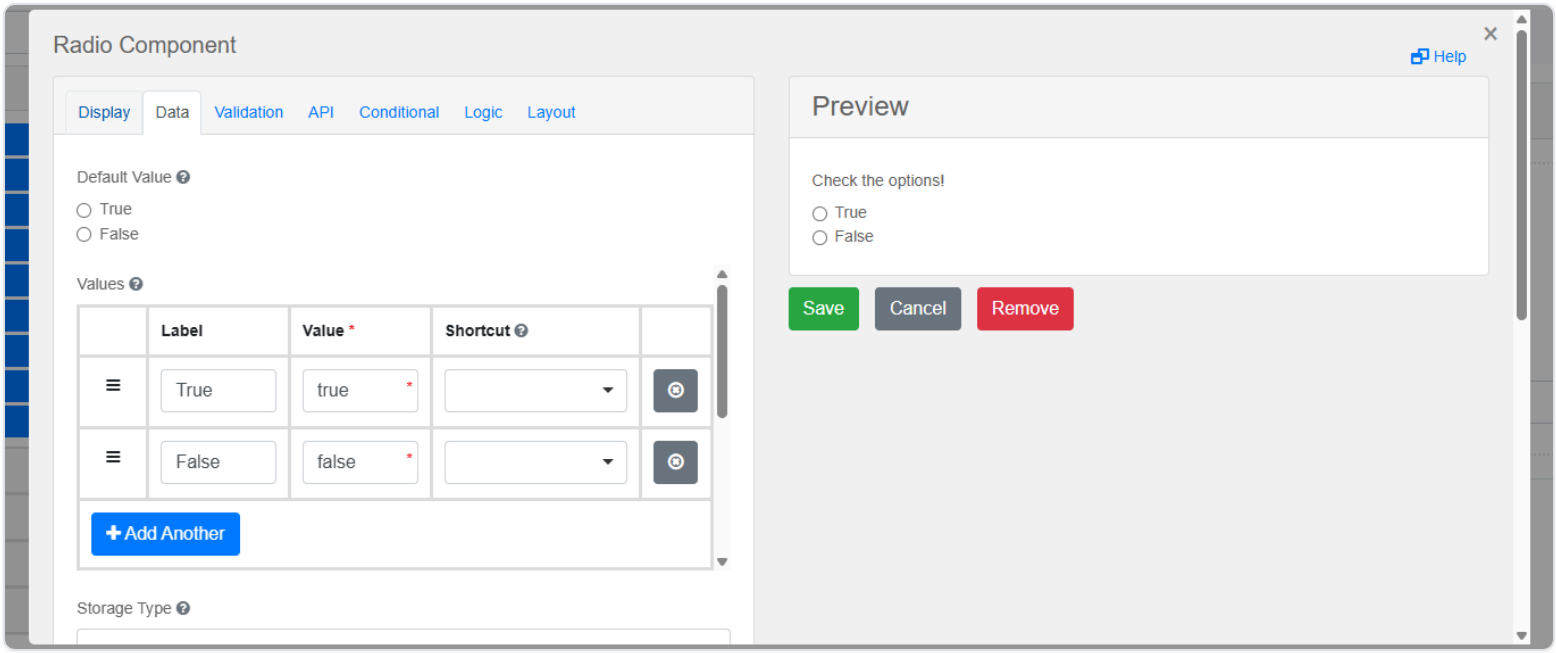
Creating Conditional Fields

Conditional fields allow you to show or hide fields based on a user's previous answer. For example, if a patient selects "Yes" to having allergies, you can display additional fields asking them to list their allergies.

Step 1: Add the Trigger Field

First, add a field that will control when other fields appear. This can be a Radio button, Checkbox, or Select dropdown. For this example, we'll use a **Radio** field.

1. Drag and drop **Radio** from the **Basic** section to your form
2. The Radio component modal will open
3. Go to the **Data** tab



Radio component Data tab with values table

From here, you can add options inside the Values table by clicking the **Add Another** button.

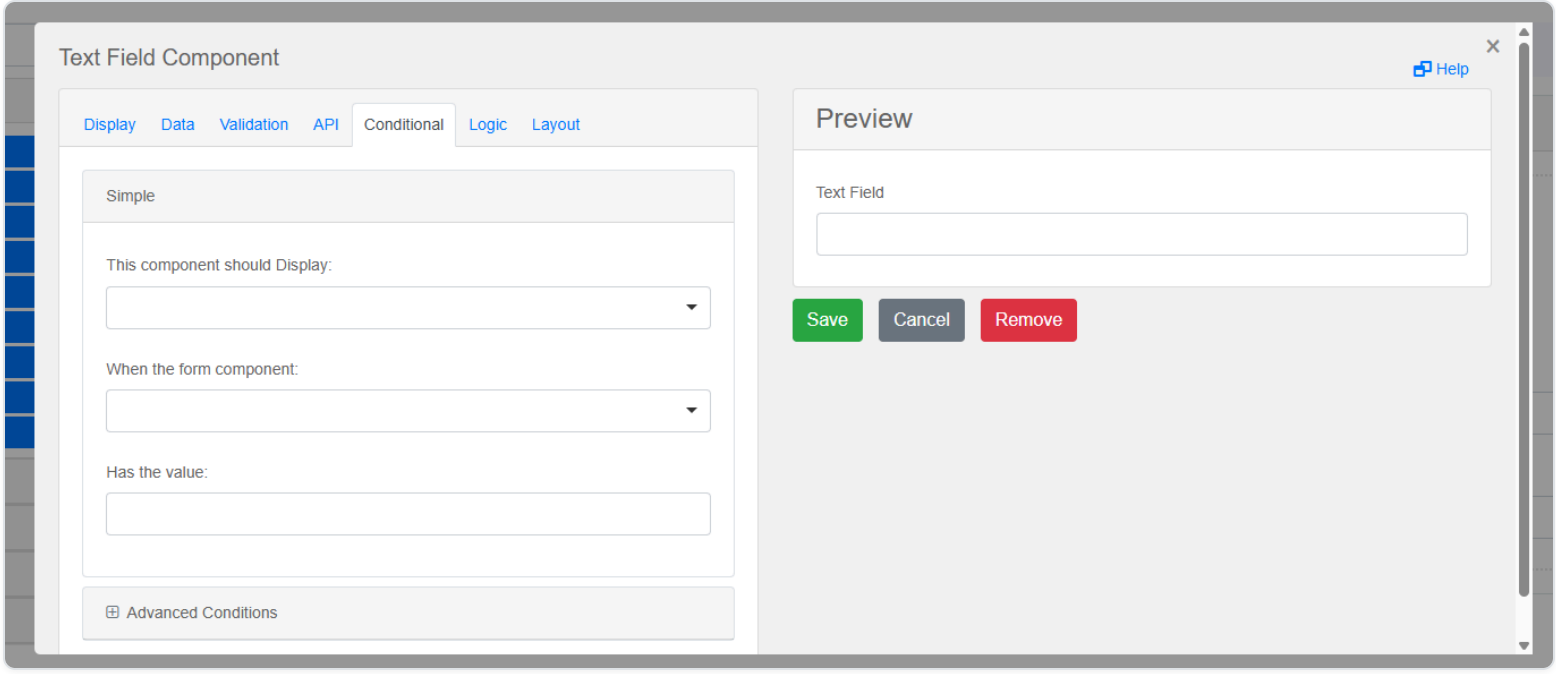
> **Important:** Note down the **Value** fields! The Label is what users see, but the Value is what you'll use to set up the conditional logic. These values are linked with their labels.

Once you're satisfied with your radio options, click the **Save** button.

Step 2: Add the Conditional Field

Now add another field that will show or hide based on the radio button selection. This can be any field type (Text Field, Text Area, etc.).

1. Drag and drop a field (e.g., **Text Field**) to your form
2. The component modal will open
3. Go to the **Conditional** tab



Conditional tab settings

Step 3: Configure the Conditional Logic

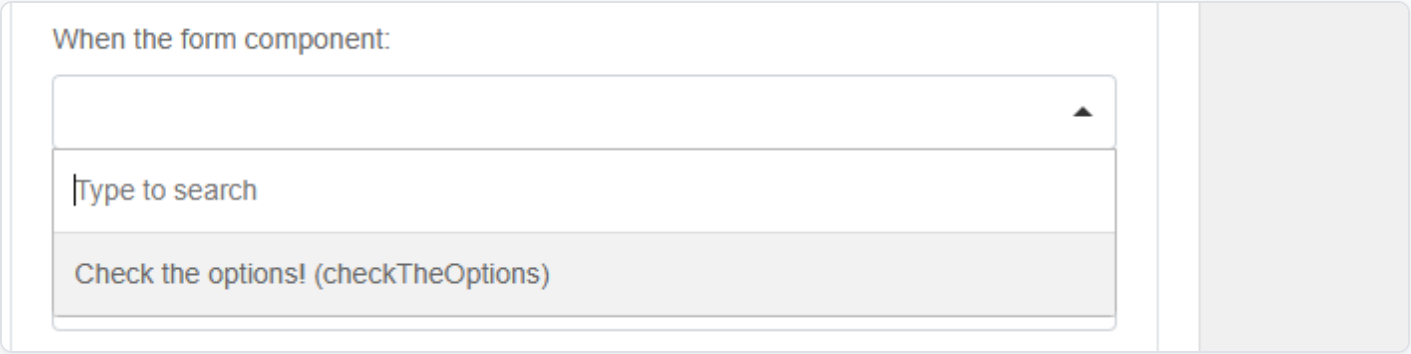
Set up the logic using these three fields:

1. This component should Display:

- **True** - The field will be SHOWN when the condition is met
- **False** - The field will be HIDDEN when the condition is met

2. When the form component:

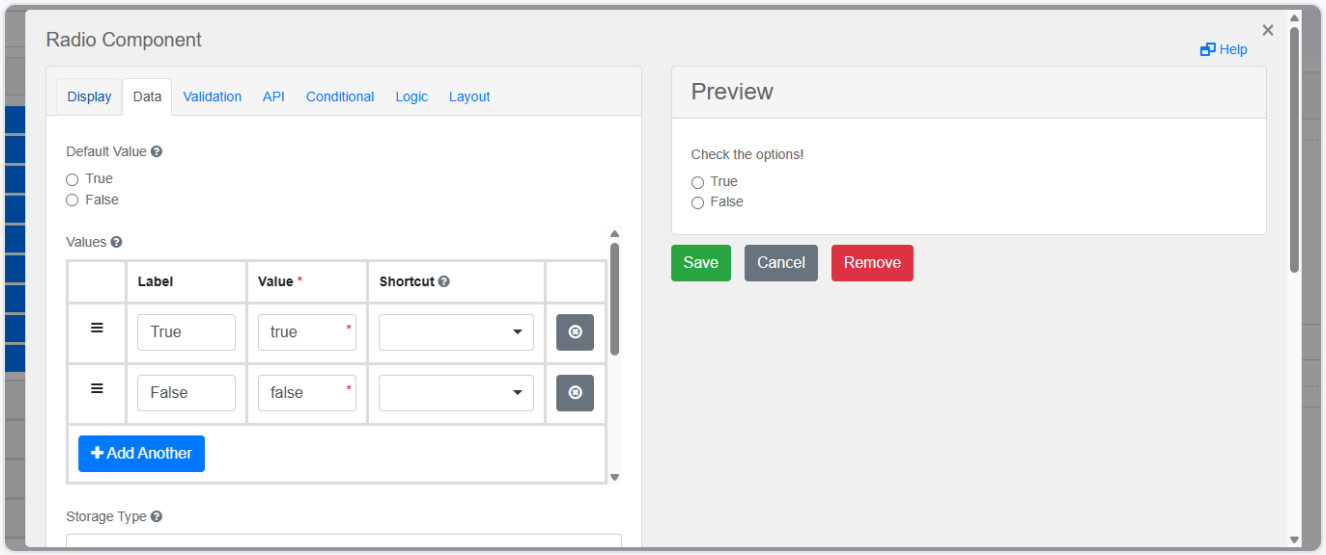
This dropdown lists all other components in your form. Select the trigger field (e.g., your Radio button).



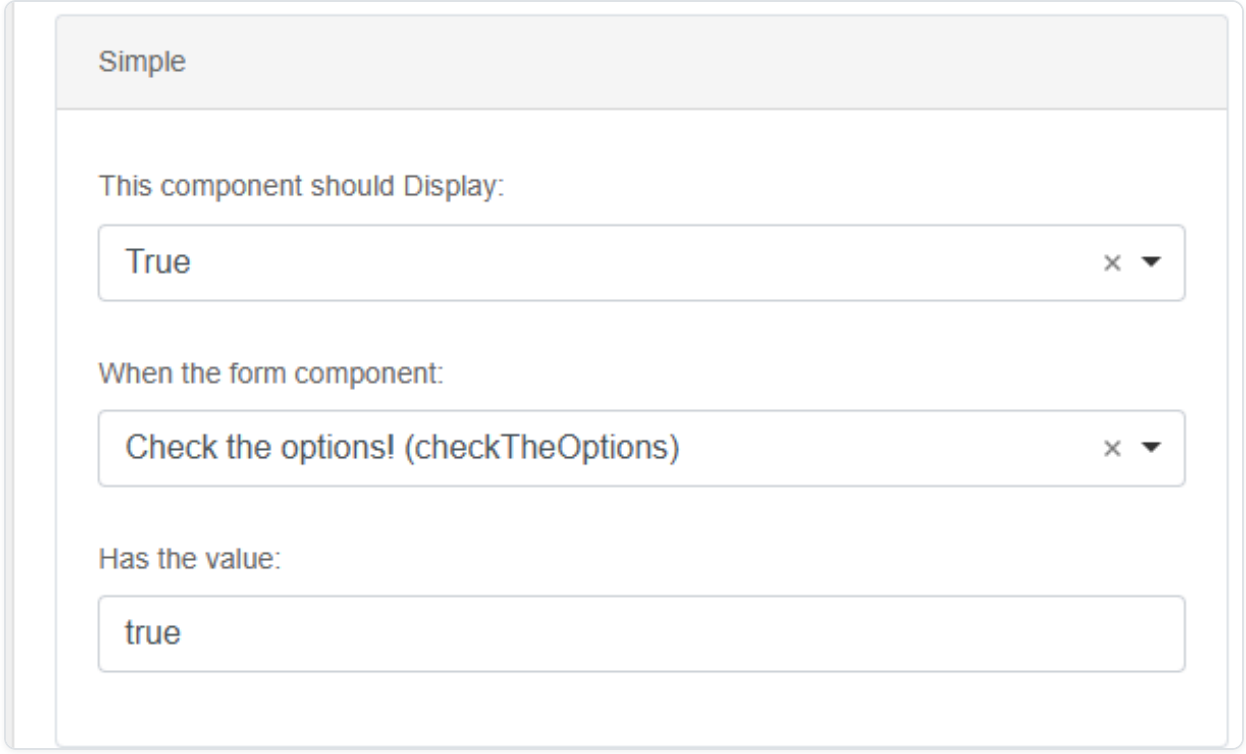
Selecting the trigger field from the dropdown

3. Has the Value:

Enter the exact **Value** (not the Label) from your trigger field. Refer back to the values you set in the Data tab.



Reference: Use the Value field (e.g., "true" or "false"), not the Label



Complete reference: All three conditional settings configured together

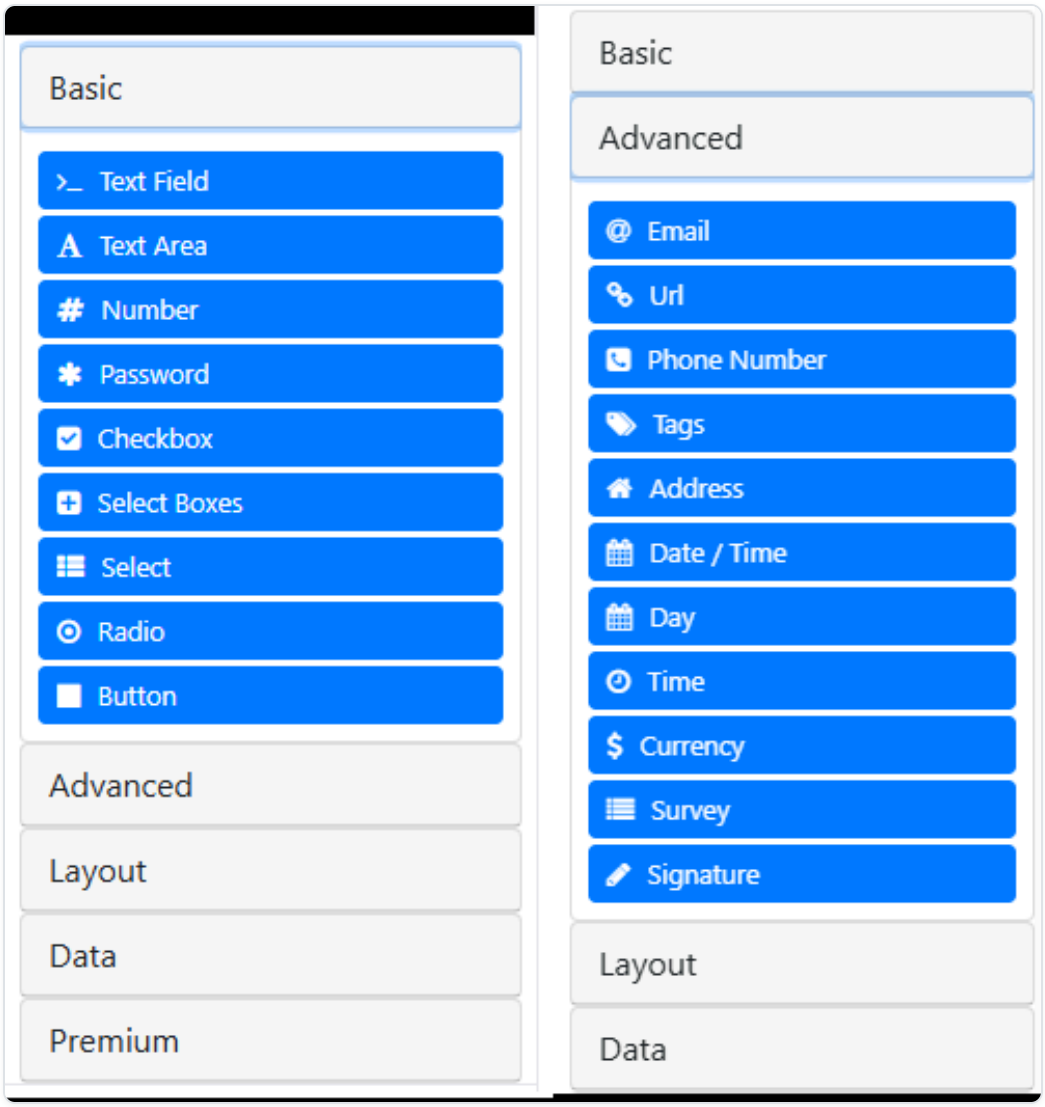
Step 4: Save and Preview

1. Click **Save** to save the conditional field
2. Save your entire form
3. Click **Preview** to test the conditional logic

Result: When users select a specific option on the trigger field, the conditional field will automatically show or hide based on your settings!

Tip: You can make multiple fields conditional on the same trigger field. Just repeat Steps 2-4 for each additional field you want to show/hide.

6. Common Field Types



Overview of common field types available in the form builder

Text Field

For single-line answers like names, addresses, or short responses.

Best for: First name, Last name, City, Email

> **Tip:** Add input mask:
>
> - `999-999-9999` for phone numbers
> - `99/99/9999` for dates

Text Area

For longer, multi-line answers like comments or explanations.

Best for: "Please explain...", "Additional comments", "Describe symptoms"

Number

For numeric values only.

Best for: Age, Number of children, Years at address

Email

Automatically checks that the email address is formatted correctly.

Best for: Contact email, Emergency contact email

Select (Dropdown)

A dropdown menu with options to choose from.

Best for: State, Gender, Insurance provider

To add options: Go to Data tab → Click "+ Add Another" → Add Label and Value

Checkbox

For selecting multiple options from a list.

Best for: "Select all that apply", Multiple acknowledgments

Radio Buttons

For choosing one option from a list (like multiple choice).

Best for: Yes/No questions, "Select one option"

Date / Time

A calendar picker for selecting dates.

Best for: Date of Birth, Appointment date, Last visit date

> **Tip:** For "today's date" that fills automatically, use a Text Field instead (see Quick Reference: Common Fields Setup).

Signature

Captures a digital signature. Found under the **Advanced** section in the left component panel.

Best for: Patient signature, Guardian signature, Consent signature

> **Important:** Always use the Signature field type for signatures, never use Text Area or Text Field.

File Upload

Allows users to upload documents or photos. Found under the **Premium** section in the left component panel as "File".

Best for: Insurance card photo, ID copy, X-rays

> **Tip:** Works with both camera (for mobile devices) and file uploads from computers. See Adding a File Upload Field for detailed setup instructions.

7. Pre-filled Patient Information

For certain forms, you can have patient information automatically filled in when patients verify their identity.

Which Information Can Be Pre-filled?

Only these three fields can be automatically filled:

- Patient First Name
- Patient Last Name
- Patient Date of Birth

How to Set This Up

Step 1: Create a form with any Form Type (this feature works with all form types)

Step 2: Add text fields for patient information with these exact property names:

For First Name:

1. Add a Text Field
2. Label: "Patient First Name"
3. Go to API tab
4. Property Name: type `patientFirstName` (exactly as shown)
5. Go to Display tab
6. Turn ON "Disabled" (so users can't change it)
7. Save

For Last Name:

1. Add a Text Field
2. Label: "Patient Last Name"
3. Go to API tab
4. Property Name: type `patientLastName` (exactly as shown)
5. Go to Display tab
6. Turn ON "Disabled"
7. Save

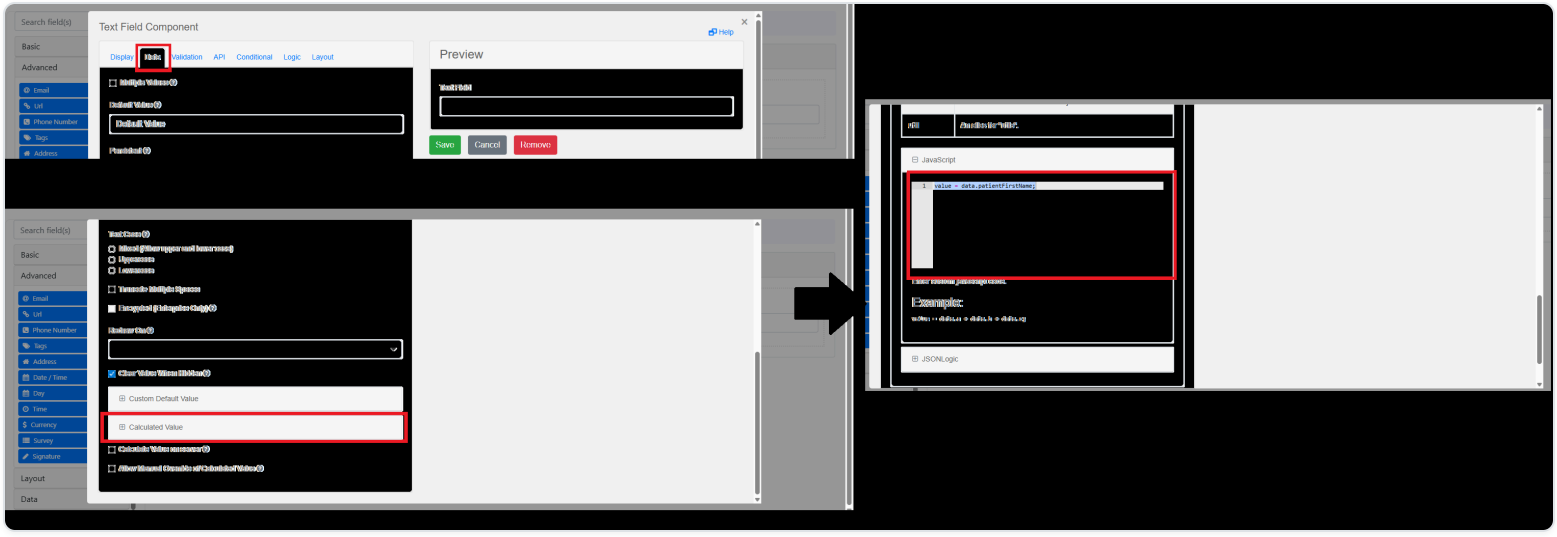
For Date of Birth:

1. Add a Text Field
2. Label: "Patient Date of Birth"
3. Go to Display tab
4. Placeholder: "MM/DD/YYYY"
5. Input Mask: `99/99/9999`
6. Turn ON "Disabled"
7. Go to API tab
8. Property Name: type `patientDOB` (exactly as shown)
9. Save

> **Important:** The property names must be spelled exactly as shown above for the pre-fill feature to work.

Using the Same Patient Info Multiple Times

Why do this? Sometimes you need the patient's information to appear in different sections of your form (for example, at the top for identification and again at the bottom for signature). Instead of asking patients to type their name twice, you can make it auto-copy from the first field.



Example of using patient information in multiple locations on a form

Step-by-Step Instructions:

1. Create your first field normally: Follow the setup steps in the section above (use property names like `patientFirstName`, `patientLastName`, or `patientDOB`). These fields will automatically pre-fill with patient information when patients verify their identity.

2. For the second instance of the same information, create a new field with a different property name:

Examples:

- **First Name:** First field = `patientFirstName` → Second field = `patientFirstNameSection2`
- **Last Name:** First field = `patientLastName` → Second field = `patientLastNameSection2`
- **Date of Birth:** First field = `patientDOB` → Second field = `patientDOBSection2`

3. Configure the second field to auto-copy from the first:

1. Add a Text Field to your form
2. Go to API tab → Set Property Name (e.g., `patientFirstNameSection2`)
3. Go to Data tab → In "Calculated Value" box, type the formula:
   - For First Name: `value = data.patientFirstName;`
   - For Last Name: `value = data.patientLastName;`
   - For Date of Birth: `value = data.patientDOB;`
4. Go to Display tab → Turn ON "Disabled" (this prevents users from manually editing this field, ensuring it always matches the original field)
5. Click **Save**

Result: The second field will automatically display the same information as the first field.

8. Dynamic Location Information

You can add your practice location's information to forms, and it will automatically update based on which location the form is being used at.

What Information Can Be Added?

- Location name
- Address
- City, State, Zip
- Phone number
- Logo

How to Add Location Information

For text information (address, phone, etc.):

1. Add a **Content** field from the Layout section (see Section 4: Adding Headers, Paragraphs, and Text Content for detailed instructions on how to add a Content field)
2. In the text editor, type your text and include these special codes:

```
Our Location: {{data.locationName}} Address: {{data.locationAddress1}} City: {{data.locationCity}}, {{data.locationState}} {{data.locationZip}} Phone: {{data.locationPhone}}
```

The information between `{{` and `}}` will automatically be replaced with your actual location information.

For adding your logo:

1. Add a **Content** field from the Layout section (see Section 4: Adding Headers, Paragraphs, and Text Content for detailed instructions) and save it blank
2. Click the **Edit Json** button

Edit Json button in the component settings

3. Copy and paste this code:

JSON code for adding location logo

```
{ "html": "<figure class=\"image\"><img src=\"{{ data.locationLogoUrl }}\" alt=\"Location Logo\" style=\"max-width: 200px; height: auto;\"></figure>", "label": "Content", "refreshOnChange": false, "key": "locationLogo", "type": "content", "input": false, "tableView": false }
```

4. Click the **Save** button

> **Tip:** Place your logo at the very top of the form for a professional look.

9. Creating Form Packets

A **packet** is a collection of multiple forms grouped together. Instead of sending patients five separate forms, you can send one packet that includes all five.

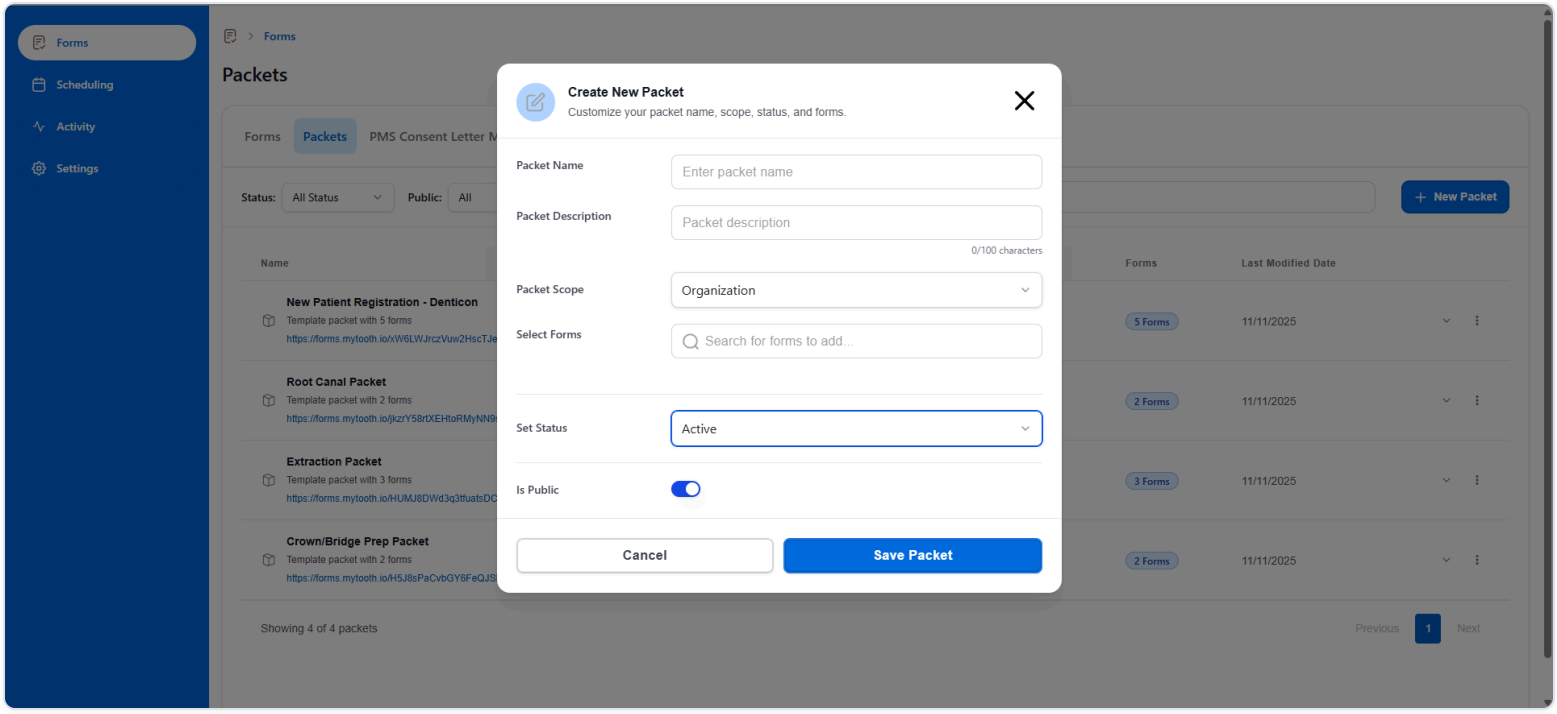
When to Use Packets

Use packets when you need to send multiple forms together:

- New patient registration
- Pre-appointment requirements
- Annual updates
- Procedure-specific forms

Creating a Packet

1. Go to **Forms** → **Packets**
2. Click **New Packet**
3. Fill out the following information:



New Packet creation modal with all fields

Packet Name

Give it a clear name

```
Example: "New Patient Registration Packet"
```

Packet Description *(optional)*

Brief explanation

```
Example: "Forms required for all new patients"
```

Packet Scope

- **Organization:** Available to all locations
- **Location:** Only visible at specific locations

Status

- Draft (while building)
- Active (ready to use)
- Inactive (hidden)

Select Forms

- Click in the box
- Search for forms you want
- Click each form to add it
- **Note:** You can add multiple forms together

Is Public

Turn ON if patients will complete this remotely

4. Click **Save Packet**

Example Packet

Packet Name: New Patient Welcome Packet

Forms Included:

1. Patient Demographics
2. HIPAA Consent
3. Financial Policy Agreement
4. Treatment Consent
5. Office Policies Acknowledgment

10. Advanced: Patient Demographic Sync

When you create a Patient Demographic form, it can automatically update patient information directly in your Practice Management System (PMS). This powerful feature ensures that patient data stays synchronized across your entire practice.

> **Important:** This section is for advanced users who need to create forms that sync patient demographic data with the PMS. For most forms, you won't need this level of integration.

How It Works

When patients fill out a Patient Demographic form, the submitted data automatically updates their records in the PMS. This eliminates manual data entry and keeps information current across all systems.

Setting Up Sync-Enabled Forms

Step 1: Set Form Type to "Patient Demographic"

When creating your form, you **must** set the Form Type to **Patient Demographic**. This tells the system that this form will sync with the PMS.

> **Tip:** You can find this setting in the Form Settings panel when you first create or edit a form (see Section 3: Getting Started).

Step 2: Use Exact Property Names

Each field in your form must use a specific Property Name that exactly matches the PMS field. Even small differences in spelling or capitalization will prevent the sync from working.

> **Critical:** Property Names are case-sensitive and must match exactly. For example, `firstName` will work, but `firstname` or `FirstName` will not.

Property Name Mapping Guide

Below is a comprehensive list of PMS fields and their corresponding Property Names. Use these when setting up your Patient Demographic form.

Basic Information

First Name:	`firstName`
Last Name:	`lastName`
Middle Initial:	`mi`
Title:	`title`
Preferred Name:	`nickName`
Pronouns:	`preferredpronouns`
Date of Birth:	`dateOfBirth`
Sex:	`sex`
Marital Status:	`maritalStatus`

Contact Information

Email:	`email`
Home Phone:	`homePhone`
Cell Phone:	`cellPhone`
Work Phone:	`workPhone`
Preferred Contact:	`prefCntMethod`
Preferred Language:	`preferredlang`

Address Information

Address:	`address`
Address Line 2:	`address2`
City:	`city`
State:	`state`
Zip Code:	`zip`

Emergency & Guardian

Emergency Contact:	`emergencyContact`
Emergency Phone:	`emergencyPhone`
Guardian Name:	`poaName`
Guardian Phone:	`poaPhone`

Identification

Chart Number:	`chartNo`
SSN:	`socialSecurityNumber`
Driver License:	`driverLicense`
Medicaid ID:	`mediId`

Additional Information

Student Status:	`studentStatus`
School Name:	`schoolName`
Height:	`height`
Weight:	`weight`
Patient Type:	`patientType`
HIPAA Notes:	`hipaaNotes`

System Flags (Checkbox Fields)

Active Patient:	`active`
Assign Benefits:	`isAssignBenefits`
HIPAA Agreement:	`isHIPAA`
No Auto SMS:	`noAutoSms`
No Auto Email:	`noAutoEmail`
No Correspondence:	`isCorrespondence`

Responsible Party

Relation to Responsible Party:	`relationToResponsibleParty`
RP First Name:	`responsiblePartyFirstName`
RP Last Name:	`responsiblePartyLastName`
RP Phone:	`responsiblePartyPhone`
RP Email:	`responsiblePartyEmail`

HIPAA Notes: `hipaaNotes` writes to the Denticon HIPAA Notes field (up to 1000 characters) and is Denticon-only — it has no equivalent in Cloud 9 and is ignored there.

Example: Creating a Basic Patient Registration Form

Here's a simple example of how to set up a patient registration form that syncs with the PMS:

1. **Create a new form** and set Form Type to "Patient Demographic"
2. **Add a Text Field** for First Name
    - Label: "First Name"
    - Go to API Tab → Property Name: `firstName`
    - Go to Validation Tab → Turn ON "Required"
3. **Add a Text Field** for Last Name
    - Label: "Last Name"
    - Go to API Tab → Property Name: `lastName`
    - Go to Validation Tab → Turn ON "Required"
4. **Add an Email Field**
    - Label: "Email Address"
    - Go to API Tab → Property Name: `email`
5. **Continue adding fields** as needed, using the Property Names from the table above

> **Result:** When patients fill out and submit this form, their information will automatically update in the PMS!

Common Mistakes to Avoid

- **Forgetting to set Form Type:** The form must be set to "Patient Demographic" type
- **Incorrect capitalization:** `firstname` will NOT work; use `firstName`
- **Adding spaces:** `first name` will NOT work; use `firstName`
- **Using custom names:** You must use the exact property names from the mapping table
- **Typos:** Even one wrong letter will break the sync

Testing Your Form

After creating your Patient Demographic form:

1. Save and publish the form
2. Test it by submitting sample data

3. Check the PMS to verify the data synced correctly

4. If data doesn't appear, double-check all Property Names match exactly

> **Pro Tip:** Start with a few essential fields (First Name, Last Name, Email) and test the sync before adding more complex fields. This makes troubleshooting easier.

11. Medical History Forms

MyTooth automatically syncs with your Practice Management System (PMS) and creates three medical history forms for your organization:

- **Medical Alerts** - Critical health warnings, allergies, and conditions
- **Dental Questionnaire** - Dental-specific health questions and oral care history
- **Medical Questionnaire** - Comprehensive medical history and current health status

> **Good News:** These forms are automatically generated based on your PMS data. You don't need to create them from scratch!

Customizing Your Medical History Forms

If you need to add or remove fields from the automatically generated forms:

1. Go to **Forms** in the left menu
2. Find the medical history form you want to edit (Medical Alerts, Dental Questionnaire, or Medical Questionnaire)
3. Click **Edit** or **Build**
4. Add new fields by dragging components from the left panel, or delete existing fields by selecting them and pressing the Delete key
5. When adding custom fields, make sure to set proper Property Names in the API tab
6. Click **Save** when done

> **Important for Custom Fields:** If you add custom fields that need to sync with your PMS, the Property Names must match the normalized camelCase format of the PMS field labels. For example: "Do you smoke?" becomes `doYouSmoke`, and "Date of Last X-ray" becomes `dateOfLastXRay`.

Understanding Property Name Normalization

When adding custom fields, property names must be normalized to camelCase:

- Remove all spaces

- First word lowercase, capitalize subsequent words
- Remove special characters

```
"Latex Allergy" → latexAllergy
"No Known Allergies" → noKnownAllergies
"Date of Last X-ray" → dateOfLastXRay
```

Need the exact Denticon labels? Check with your Denticon system administrator to get the precise list of medical alert, dental question, and medical question labels used in your system. Property names must be normalized versions of these exact labels.

12. Advanced: Consent Forms

Used inside Consent type forms. When the user checks these, the values are synced to Denticon, and Voice, SMS, and Email are sent to the patient accordingly.

Consent Opt-ins

SMS opt-in:	`smsConsent`
Email opt-in:	`emailConsent`
Voice opt-in:	`voiceConsent`

Contact Fields (optional)

Phone number:	`consentPhoneNumber`
Email address:	`consentEmailAddress`

> **Opt-in vs. contact fields:** The three opt-in checkboxes are the consent itself — include at least one, or no consent is synced. The two contact fields are optional: they record which phone number or email the consent applies to, and never trigger a sync on their own.

Example: Creating a Consent Form

Here's how to set up a Consent form that records communication consent in the PMS:

1. **Create a new form** and set Form Type to "Consent"
2. **Add a Checkbox** for text-message consent
    - Label: "I agree to receive text messages (SMS)"
    - Go to API Tab → Property Name: `smsConsent`
3. **Add a Checkbox** for email consent
    - Label: "I agree to receive emails"
    - Go to API Tab → Property Name: `emailConsent`
4. **Add a Checkbox** for voice-call consent
    - Label: "I agree to receive phone (voice) calls"
    - Go to API Tab → Property Name: `voiceConsent`

5. **(Optional) Add a Text Field** for the phone number the consent applies to
    - Label: "Phone number for these messages"
    - Go to API Tab → Property Name: `consentPhoneNumber`
6. **(Optional) Add a Text Field** for the email address the consent applies to
    - Label: "Email address for these messages"
    - Go to API Tab → Property Name: `consentEmailAddress`
7. **Save and publish** the form

> **Result:** When a patient checks a box and submits, that consent is saved in Denticon. At least one opt-in checkbox must be present for anything to sync.

13. Merge Fields

What are merge fields?

Merge fields let you embed patient, provider, or office information directly into a form at the point it is opened by the patient. Instead of a generic form, the patient sees their own name, date of birth, address, and other details already filled in.

In the form builder, merge fields appear as a dedicated **Merge Fields** group in the left-hand component palette — right alongside the built-in *Basic*, *Advanced*, and *Layout* groups. You add one by dragging it onto the form, exactly like any other field. Each merge field is pre-configured, pre-filled with the patient's value, and locked so it cannot be edited.

> **Feature availability:** The Merge Fields palette group only appears when the feature is enabled for your organization. If you do not see it, contact your administrator.

Under the hood, each merge field uses **Form.io's native interpolation** (`{{ data.<key> }}`). When a patient opens a form via a personalised magic-link, the MyTooth backend injects their data into the form, so the value resolves automatically at render time — including in generated PDFs. The same mechanism that resolves `{{ data.locationLogoUrl }}` (the practice logo) resolves all merge fields.

Adding a merge field

1. Drag from the Merge Fields group (recommended)

In the builder's left palette, open the **Merge Fields** group. Drag the field you want (for example *Patient First Name*) onto the form canvas, just like a Text Field or Checkbox.

The dropped field arrives already keyed to the patient value, pre-filled, and *locked* — the patient sees their data and cannot accidentally overwrite it. No copying, pasting, or token typing required.

> Communication-preference fields (No SMS, No Voice, No Email) are added as **checkboxes**; all other merge fields are added as text fields.

2. Inline tokens in text (advanced)

To weave a value into a sentence or heading, you can still type the raw token `{{ data.<key> }}` directly into an **HTML Element** or **Content** component, or into a **Panel** title. Form.io resolves it at render time. Example:

```
Dear {{ data.patientFirstName }} {{ data.patientLastName }},
```

Use the Available merge fields table below for the exact token of each value.

Identified vs. anonymous forms

Merge fields only resolve when a form is opened via a **personalised (magic-link) URL** sent directly to an identified patient. The backend attaches the patient's data to that specific link.

> **Anonymous / public links:** If a form is configured as a public URL (accessible without a personal invitation), merge fields will not pre-populate. A dragged merge field renders empty, and an inline token typed into content shows the raw text — for example, `{{ data.patientFirstName }}` — instead of the patient's name. The form builder shows a warning banner whenever a form is set to public to remind you of this.

To avoid confusion, either:

- Avoid merge fields on forms intended for public/anonymous use, or
- Switch the form to identified-patient delivery (disable the public URL toggle).

Available merge fields

The table below lists every merge field available in the **Merge Fields** palette group. The **Token** column is the underlying interpolation value — useful if you want to embed it inline in an HTML/Content component or a panel title (see method 2 above). The **PMS** note under each group shows which Practice Management Systems supply the value.

Patient

Patient First Name	`{{ data.patientFirstName }}`
Patient Last Name	`{{ data.patientLastName }}`
Patient Date of Birth	`{{ data.patientDOB }}`
Patient Email	`{{ data.patientEmail }}`
Patient Phone	`{{ data.patientPhone }}`
Patient Address	`{{ data.patientAddress }}`
Patient City	`{{ data.patientCity }}`
Patient State	`{{ data.patientState }}`
Patient ZIP Code	`{{ data.patientZip }}`

All Patient tokens are supported on both Denticon and Cloud 9.

Responsible Party

Responsible Party First Name	`{{ data.responsiblePartyFirstName }}`
Responsible Party Last Name	`{{ data.responsiblePartyLastName }}`
Responsible Party Phone	`{{ data.responsiblePartyPhone }}`
Responsible Party Email	`{{ data.responsiblePartyEmail }}`
Responsible Party Address	`{{ data.responsiblePartyAddress }}`

All Responsible Party tokens are supported on both Denticon and Cloud 9.

Provider

Provider Name	`{{ data.providerName }}`

Supported on both Denticon and Cloud 9.

Office

Office Name	`{{ data.officeName }}`

Supported on both Denticon and Cloud 9.

Communication Preferences

No SMS	`{{ data.noSMS }}`
No Voice	`{{ data.noVoice }}`
No Email	`{{ data.noEmail }}`

These resolve to `true` or `false` based on the patient's stored communication opt-outs. Supported on both Denticon and Cloud 9.

> **Address line 2 not available:** There is no merge field for a second address line (`patientAddress2`) because neither Denticon nor Cloud 9 provides this value in the patient payload at this time.

Common use cases

1. Personalised greeting in a consent form

Add an **HTML/Content** component at the top of the form with the body:

```
Dear {{ data.patientFirstName }} {{ data.patientLastName }},

Please review and sign the following consent form for your upcoming appointment at {{
data.officeName }}.
```

The patient will see their full name and your office name filled in when they open the link.

2. Pre-filled demographic form

From the **Merge Fields** palette group, drag the fields you want onto the form:

```
Drag: Patient First Name, Patient Last Name, Patient Email, Patient Phone
```

Each lands pre-filled and locked with the patient's stored value. The patient reviews their existing details at a glance. (To let the patient *edit* a value, add a normal Text Field instead and reference the token in its Default Value.)

3. Provider context in a treatment consent

Add a paragraph in an **HTML/Content** component:

```
Your treating provider is {{ data.providerName }}.
```

This is especially useful for multi-provider practices where the same form template is used by all providers.

Quick Reference: Field Settings

Quick guide to the most commonly used field settings.

Make Field Required

Location: Validation Tab

Setting: Turn ON "Required"

Forces users to fill out this field before submitting the form.

Make Field Disabled

Location: Display Tab

Setting: Turn ON "Disabled"

Makes field read-only. Useful for auto-filled or calculated fields.

Add Input Mask

Location: Display Tab

Setting: Input Mask field

Examples:

- `999-999-9999` for phone
- `99/99/9999` for dates

Set Property Name

Location: API Tab

Setting: Property Name field

Use camelCase format like `patientFirstName` for integration with other systems.

Add Placeholder Text

Location: Display Tab

Setting: Placeholder field

Shows example text inside empty field. Example: "Enter your full name"

Set Min/Max Length

Location: Validation Tab

Setting: Minimum Length / Maximum Length

Controls how many characters users can enter. Useful for limiting text length.

Make Field Hidden

Location: Display Tab

Setting: Turn ON "Hidden"

Hides field from view. Can still store data for calculations or integrations.

Set Default Value

Location: Data Tab

Setting: Default Value field

Pre-fills field with a default value. Users can change it if needed.

Quick Reference: Common Fields Setup

Ready-to-use field configurations for common form elements.

Name Field

Type: Text Field

Label: "Patient's First Name"

Property Name: `patientFirstName`

Required: YES

Phone Field

Type: Text Field

Label: "Cell Phone"

Property Name: `cellPhone`

Input Mask: `999-999-9999`

Placeholder: "___-___-____"

Email Field

Type: Email

Label: "Email Address"

Property Name: `email`

Required: YES

Date of Birth

Type: Text Field

Label: "Date of Birth"

Property Name: `patientDOB`

Input Mask: `99/99/9999`

Placeholder: "MM/DD/YYYY"

Required: YES

Today's Date (Auto-filled)

Type: Text Field

Label: "Today's Date"

Property Name: `todaysDate`

Input Mask: `99/99/9999`

Disabled: YES

Calculated Value:

```
value = moment().format('MM/DD/YYYY');
```

Signature

Type: Signature

Label: "Patient Signature"

Property Name: `patientSignature`

Required: YES

Yes/No Question

Type: Radio

Label: "Do you have insurance?"

Property Name: `hasInsurance`

Options:

- Yes (value: yes)
- No (value: no)

File Upload

Type: File

Label: "Upload Insurance Card"

Property Name: `insuranceCard`

Image: YES

File Max Size: 50MB

Storage: `base64`

> **See detailed setup instructions →**
