Anvil's new Personal Plan is here: custom domains, Full Python and more for $15/mo! See the announcement >>

Start building

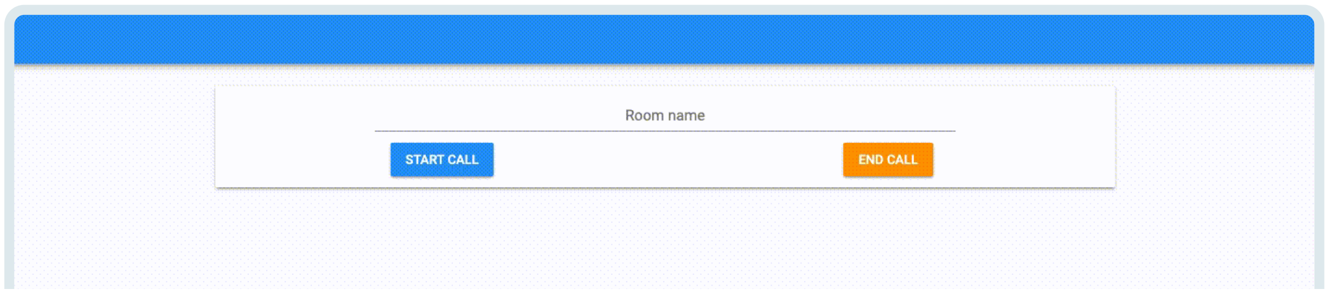
Using Daily's video chat API with Anvil

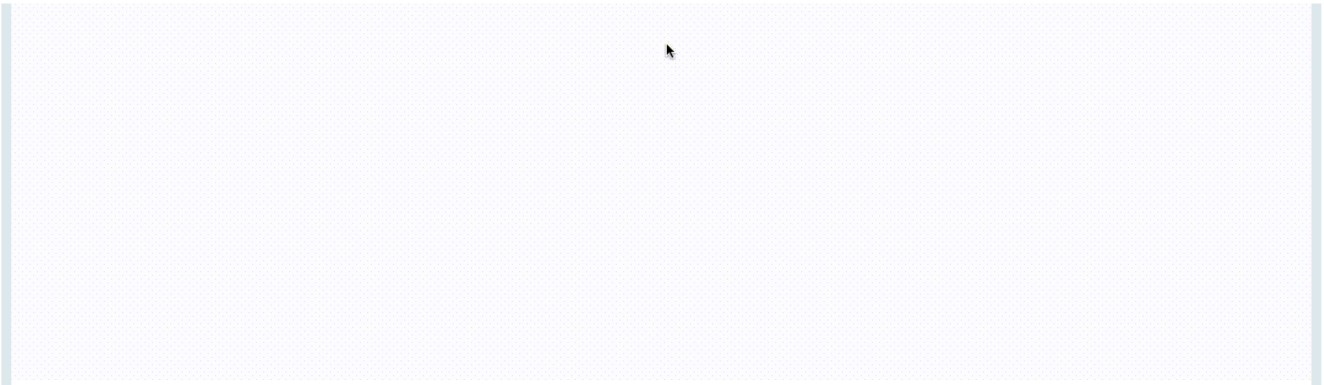
By Ryan

Adding video calls to your app with a few lines of code

Daily's API lets you add real time video calls to any app with just a few lines of code. Anvil makes it easy to build web apps entirely in Python – no Javascript required. In this post, I'm going to show you how to use them together and show you how to get started with Daily's API.

We're going to build an app that lets your users join a video call. You could integrate it with anything – communication for web based games, video-based collaboration software, or live customer support. (Click here to see some examples of apps you can build with Anvil.)




Daily call being started

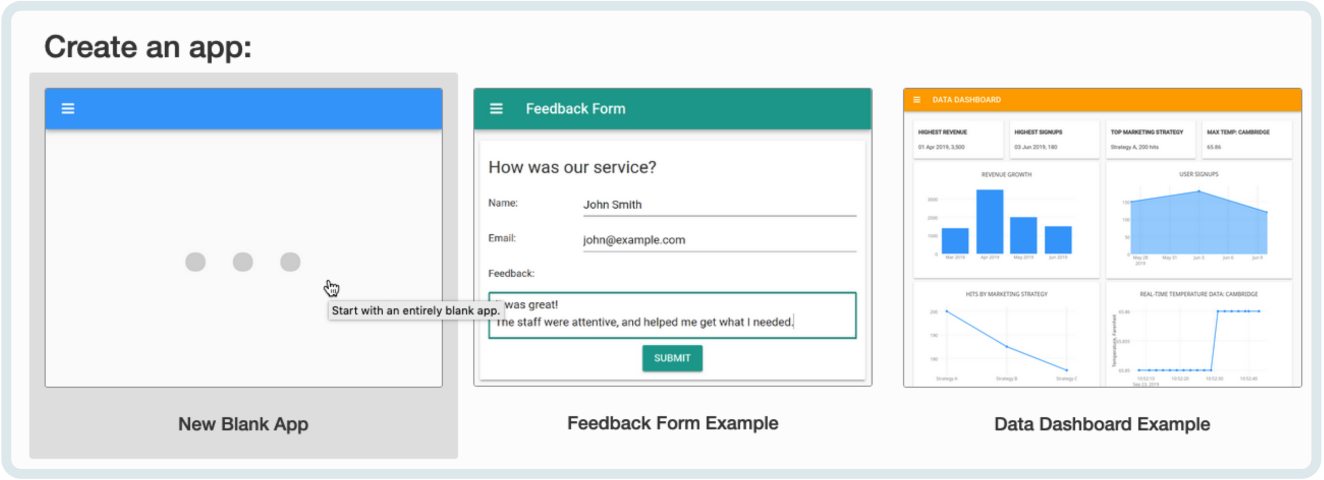
Wait – isn't Daily a JavaScript API? That's right, but with Anvil **you can import Javascript libraries into your Python front end**. How? Read on...

Create a form to join video calls

Creating an app

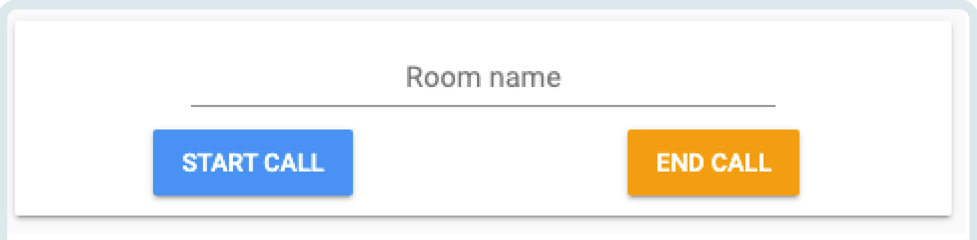
Creating web apps with Anvil is simple. We'll create one to get started.

[Log in](#) to Anvil and click 'New Blank App'. Choose the Material Design theme.



Creating a UI

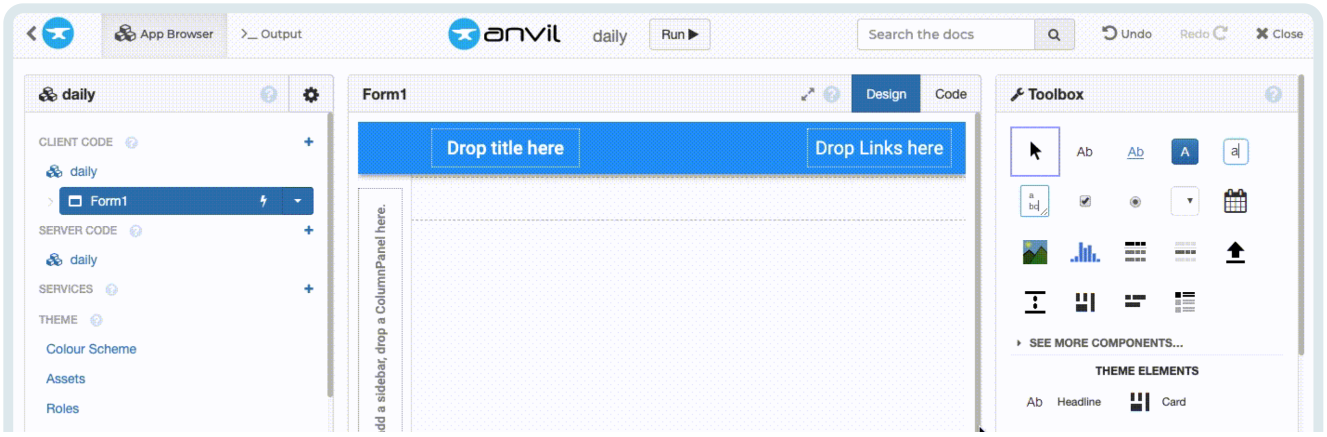
We need a form with a text box to enter the name of our meeting room and buttons to start and stop the call. Our finished form will look something like this:

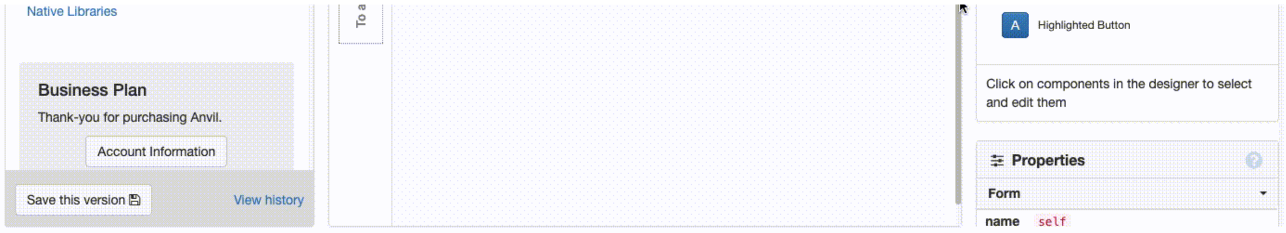


Finished call form

We construct the form by dragging-and-dropping [components](). Let's start by dropping a Card into our form. Then drag a [TextBox]() into the Card and, in the properties panel on the right, change the name of the component to `room_name_textbox`.

Underneath the TextBox, drag and drop two [Buttons](). Change the first Button's name to `start_call_button` and its text to `Start call`. Change the second Button's name to `end_call_button` and its text to `End call`.





Building our call form

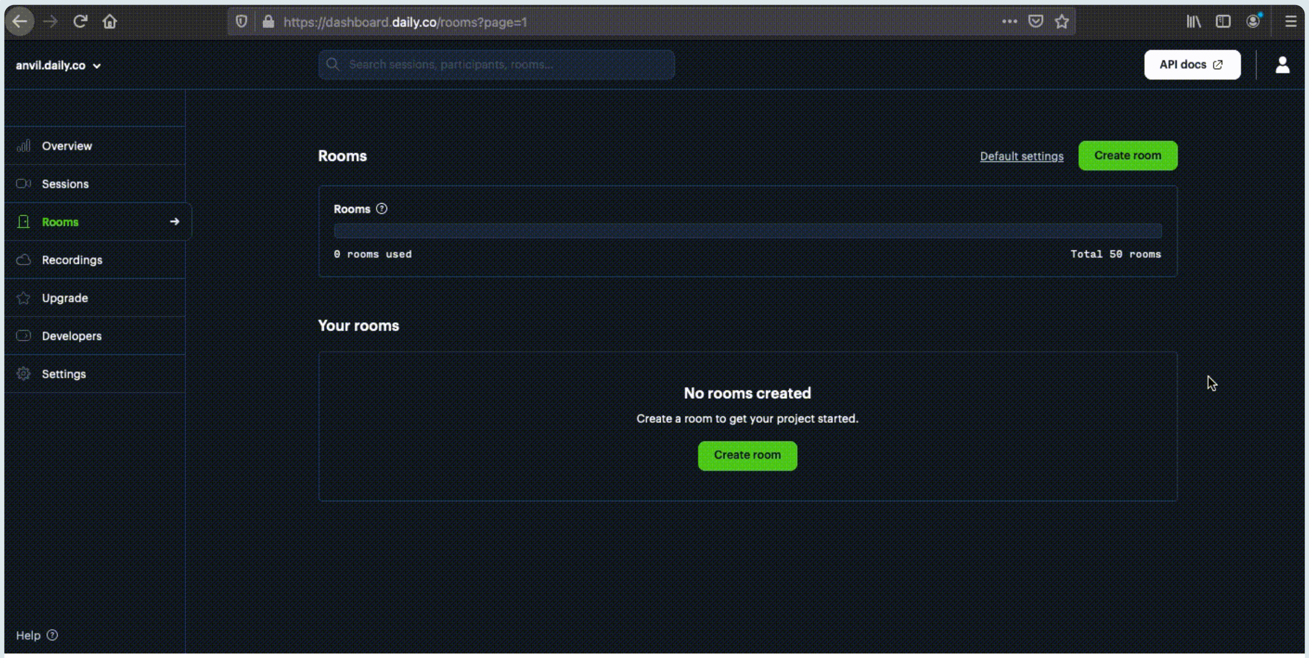
That's it! Our user interface is finished.

Using the Daily API

Creating a Daily video call room

We need to create a video call room our Anvil app users can join.

Let's do this by going to https://dashboard.daily.co/rooms, clicking `Create room` and giving our room a name.



Building our call form

For more information on creating rooms, Daily has a useful guide [here](#).

Importing the Daily API library

To import Daily's library, navigate to our app's Native Libraries and add the following line of code:

```
<script crossorigin src="https://unpkg.com/@daily-co/daily-js'
```

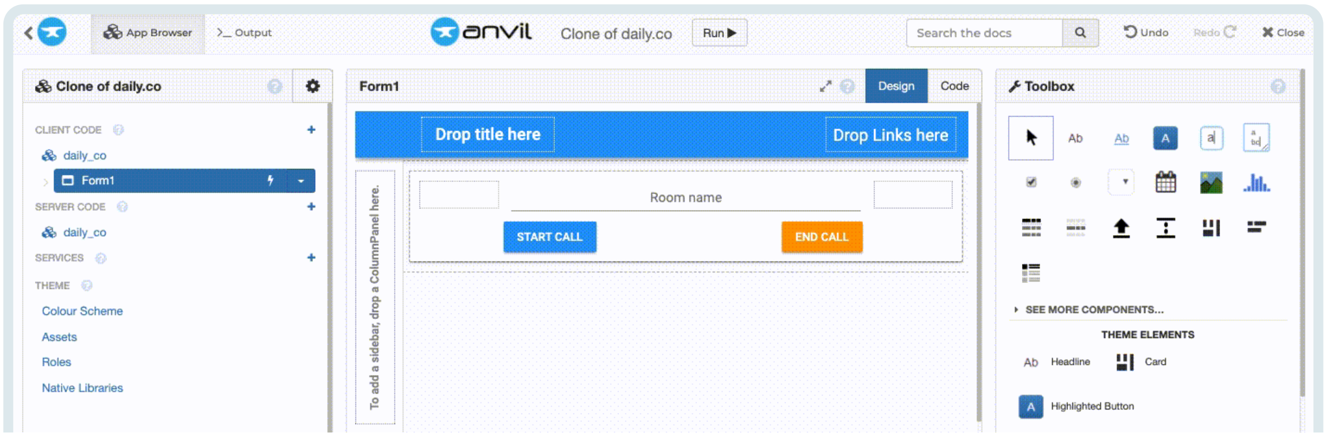
Starting a call

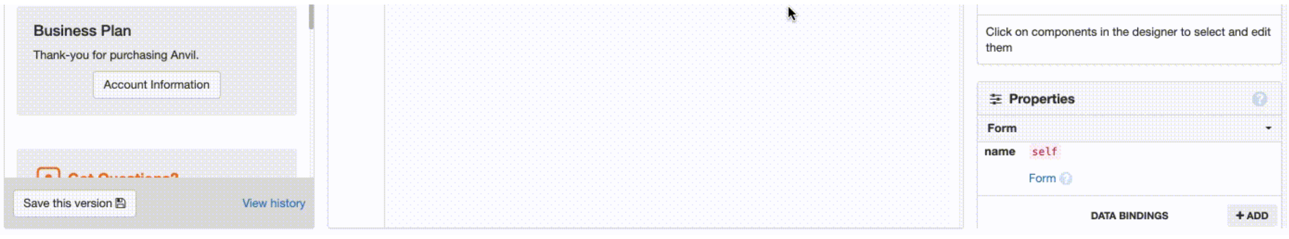
Anvil lets you import and use JavaScript functions in Python code - handling all the conversion for you. Let's write the code our user interface needs to start a call.

Navigate back to our **`Form1`** and, at the top of our Form's Code View, import the **`DailyIframe`** class.

```
from anvil.js.window import DailyIframe
```

Then, back in the Form's Design View, create a click event handler for our **`start_call_button`**.





Adding a click event to our ‘Start call’ button

The **`start_call_button_click`** function will be called every time the button is clicked. In the function, we’ll check if the user has entered a room name and then create an instance of a **`DailyIframe`** called **`call_frame`**. Then, we’ll call the **`DailyIframe`**’s [**`join()`**] method, passing it our meeting room link plus the **`room_name`** as a parameter.

*(Replace **`'https://your-team.daily.co/'`** with your own meeting room link, which you can find in your [Daily dashboard].)*

→

```
def start_call_button_click(self, **event_args):
    """This method is called when the button is clicked'
    room_name = self.room_name_textbox.text
    if room_name:
      self.call_frame = DailyIframe.createFrame()
      self.call_frame.join({ 'url': 'https://anvil.daily
```

What if that room doesn’t exist?

If the specified room doesn’t exist, the **`join()`** method will throw an exception. We can catch this the usual way, with a [**`try`**] block. Then we’ll use Anvil’s [**`alert()`**] function to pop up a dialog:.

→

```
def start_call_button_click(self, **event_args):
    """This method is called when the button is clicked'
    room_name = self.room_name_textbox.text
```

```
        if room_name:
          self.call_frame = DailyIframe.createFrame()
          try:
            self.call_frame.join({ 'url': 'https://anvil.da
          except Exception as e:
            if "The meeting you're trying to join does not 
              alert("That room doesn't exist.")
              self.end_call_button_click()
            else:
              raise
```

That’s our start call functionality finished. Let’s write the functionality that will end the call.

Ending a call

When the user clicks “End Call”, we want to end the call, by calling the `DailyIframe`’s `leave()` and `destroy()` methods.

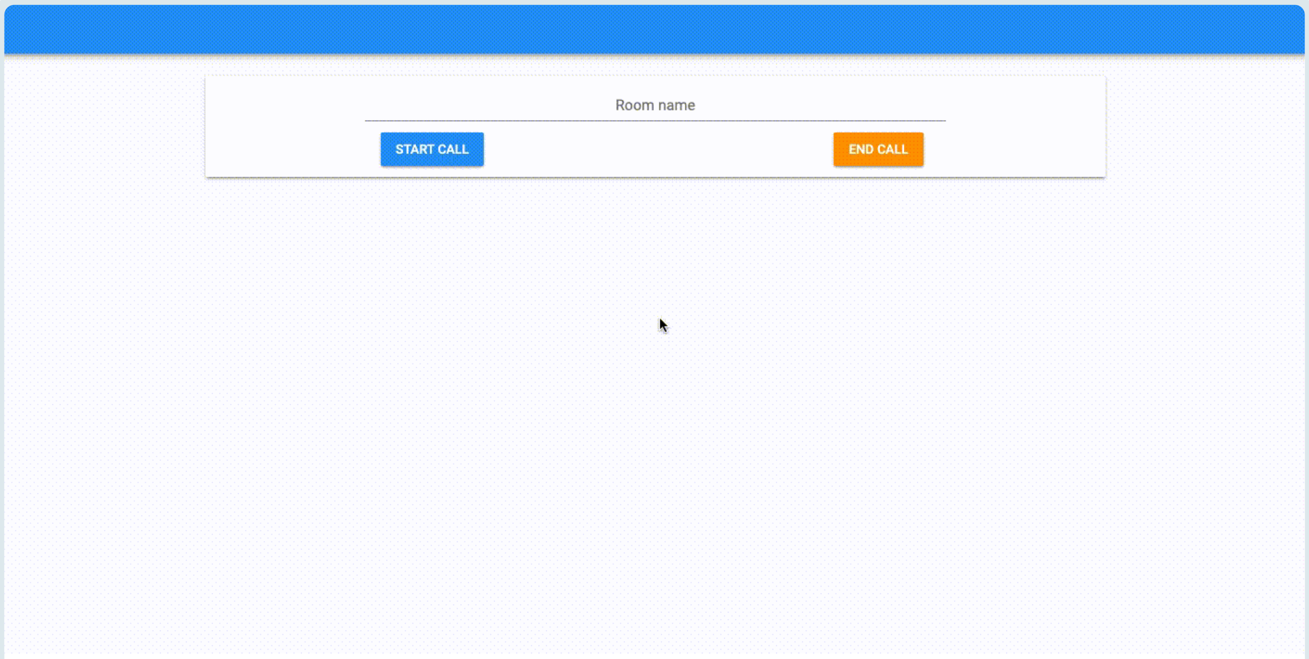
Like we did with **`start_call_button`**, create an event handler function for **`end_call_button`** called **`end_call_button_click`**. Then call those methods:

```
    def end_call_button_click(self, **event_args):
        """This method is called when the button is clicked
        self.call_frame.leave()
        self.call_frame.destroy()
```

Great work! Our app’s users can now enter the name of the meeting they want to join, start and end the call all without ever leaving your Anvil app.

That’s it!

We've just created a web app with nothing but Python; integrated it with Daily, and had it start a video call with the click of a button. Pretty cool, huh?



Daily call being started

Clone the App

For those of you who want to see the finished source code for this app:

See the finished app

New to Anvil?

If you're new here, welcome! Anvil is a platform for building full-stack web apps with nothing but Python. No need to wrestle with JS, HTML, CSS, Python, SQL and all their frameworks – just **build it all in Python**.

Yes – Python that runs in the browser. Python that runs on the server. Python that builds your UI. A drag-and-drop UI editor. We even have a built-in Python database, in case you don't have your own.

Why not have a play with the app builder? **It's free!** Click here to get started:

Get building ⇢

Want to try another tutorial? Learn about databases in Anvil with our Feedback Form tutorial:

Fastest

Data Dashboard

Go

Build Database-Backed Apps

Go

Build a Simple Feedback Form

Go

Develop

Build

Features

Pricing

Support

Docs

Forum

Learning Centre

Company

About us

Jobs

Press

Contact

Privacy Terms of Service