

New to Gradio? Start here: **Getting Started**

See the **Release History**

ChatInterface →

Interface

```
gradio.Interface(fn, inputs, outputs, ...)
```

Description

Interface is Gradio's main high-level class, and allows you to create a web-based GUI / demo around a machine learning model (or any Python function) in a few lines of code. You must specify three parameters: (1) the function to create a GUI for (2) the desired input components and (3) the desired output components. Additional parameters can be used to control the appearance and behavior of the demo.

Example Usage

```
import gradio as gr
```

```
def image_classifier(inp):  
    return {'cat': 0.3, 'dog': 0.7}
```

```
demo = gr.Interface(fn=image_classifier, inputs="image", outputs="label")  
demo.launch()
```

Initialization

Parameter

Description

fn

Callable

required

The function to wrap an interface around. Often a machine learning model's prediction function. Each parameter of the function corresponds to one input component, and the function should return a single value or a tuple of values, with each element in the tuple corresponding to one output component.

meter	Description
<div>inputs</div> <div><i>str</i> <i>Component</i> <i>list[<i>str</i> <i>Component</i>]</i> <i>None</i></div> <div><i>required</i></div>	<p>A single Gradio component, or list of Gradio components.</p> <p>Components can either be passed as instantiated objects, or referred to by their string shortcuts. The number of input components should match the number of parameters in fn. If set to None, then only the output components will be displayed.</p>
<div>outputs</div> <div><i>str</i> <i>Component</i> <i>list[<i>str</i> <i>Component</i>]</i> <i>None</i></div> <div><i>required</i></div>	<p>A single Gradio component, or list of Gradio components.</p> <p>Components can either be passed as instantiated objects, or referred to by their string shortcuts. The number of output components should match the number of values returned by fn. If set to None, then only the input components will be displayed.</p>
<div>examples</div> <div><i>list[Any]</i> <i>list[list[Any]]</i> <i>str</i> <i>None</i></div> <div>default: None</div>	<p>Sample inputs for the function; if provided, appear below the UI components and can be clicked to populate the interface. Should be nested list, in which the outer list consists of samples and each inner list consists of an input corresponding to each input component. A string path to a directory of examples can also be provided, but it should be within the directory with the python file running the gradio app. If there are multiple input components and a directory is provided, a log.csv file must be present in the directory to link corresponding inputs.</p>
<div>cache_examples</div> <div><i>bool</i> <i>None</i></div> <div>default: None</div>	<p>If True, caches examples in the server for fast runtime in examples. If <code>fn</code> is a generator function, then the last yielded value will be used as the output. The default option in HuggingFace Spaces is True. The default option elsewhere is False.</p>
<div>examples_per_page</div> <div><i>int</i></div> <div>default: 10</div>	<p>If examples are provided, how many to display per page.</p>

meter	Description
<div>live</div> <div><i>bool</i></div> <div>default: False</div>	Whether the interface should automatically rerun if any of the inputs change.
<div>title</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	A title for the interface; if provided, appears above the input and output components in large font. Also used as the tab title when opened in a browser window.
<div>description</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	A description for the interface; if provided, appears above the input and output components and beneath the title in regular font. Accepts Markdown and HTML content.
<div>article</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	An expanded article explaining the interface; if provided, appears below the input and output components in regular font. Accepts Markdown and HTML content.
<div>thumbnail</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	String path or url to image to use as display image when the web demo is shared on social media.
<div>theme</div> <div><i>Theme</i> <i>str</i> <i>None</i></div> <div>default: None</div>	Theme to use, loaded from <code>gradio.themes</code> .
<div>css</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	Custom css as a string or path to a css file. This css will be included in the demo webpage.

meter	Description
<div>allow_flagging</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	One of "never", "auto", or "manual". If "never" or "auto", users will not see a button to flag an input and output. If "manual", users will see a button to flag. If "auto", every input the user submits will be automatically flagged (outputs are not flagged). If "manual", both the input and outputs are flagged when the user clicks flag button. This parameter can be set with environmental variable GRADIO_ALLOW_FLAGGING; otherwise defaults to "manual".
<div>flagging_options</div> <div><i>list[str]</i> <i>list[tuple[str, str]]</i> <i>None</i></div> <div>default: None</div>	If provided, allows user to select from the list of options when flagging. Only applies if allow_flagging is "manual". Can either be a list of tuples of the form (label, value), where label is the string that will be displayed on the button and value is the string that will be stored in the flagging CSV; or it can be a list of strings ["X", "Y"], in which case the values will be the list of strings and the labels will ["Flag as X", "Flag as Y"], etc.
<div>flagging_dir</div> <div><i>str</i></div> <div>default: "flagged"</div>	What to name the directory where flagged data is stored.
<div>flagging_callback</div> <div><i>FlaggingCallback</i> <i>None</i></div> <div>default: None</div>	None or an instance of a subclass of FlaggingCallback which will be called when a sample is flagged. If set to None, an instance of gradio.flagging.CSVLogger will be created and logs will be saved to a local CSV file in flagging_dir. Default to None.
<div>analytics_enabled</div> <div><i>bool</i> <i>None</i></div> <div>default: None</div>	Whether to allow basic telemetry. If None, will use GRADIO_ANALYTICS_ENABLED environment variable if defined, or default to True.

meter	Description
<div>batch</div> <div>bool</div> <div>default: False</div>	If True, then the function should process a batch of inputs, meaning that it should accept a list of input values for each parameter. The lists should be of equal length (and be up to length <code>max_batch_size</code>). The function is then <i>required</i> to return a tuple of lists (even if there is only 1 output component), with each list in the tuple corresponding to one output component.
<div>max_batch_size</div> <div>int</div> <div>default: 4</div>	Maximum number of inputs to batch together if this is called from the queue (only relevant if batch=True)
<div>api_name</div> <div>str Literal[False] None</div> <div>default: "predict"</div>	Defines how the endpoint appears in the API docs. Can be a string, None, or False. If set to a string, the endpoint will be exposed in the API docs with the given name. If None, the name of the prediction function will be used as the API endpoint. If False, the endpoint will not be exposed in the API docs and downstream apps (including those that <code>gr.load</code> this app) will not be able to use this event.
<div>allow_duplication</div> <div>bool</div> <div>default: False</div>	If True, then will show a 'Duplicate Spaces' button on Hugging Face Spaces.
<div>concurrency_limit</div> <div>int None Literal['default']</div> <div>default: "default"</div>	If set, this is the maximum number of this event that can be running simultaneously. Can be set to None to mean no concurrency_limit (any number of this event can be running simultaneously). Set to "default" to use the default concurrency limit (defined by the <code>default_concurrency_limit</code> parameter in <code>.queue()</code> , which itself is 1 by default).
<div>js</div> <div>str None</div> <div>default: None</div>	Custom js or path to js file to run when demo is first loaded. This javascript will be included in the demo webpage.



<div>head</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	<p>Custom html to insert into the head of the demo webpage. This can be used to add custom meta tags, scripts, stylesheets, etc. to the page.</p>
<div>additional_inputs</div> <div><i>str</i> <i>Component</i> <i>list[str Component]</i> <i>None</i></div> <div>default: None</div>	<p>A single Gradio component, or list of Gradio components.</p> <p>Components can either be passed as instantiated objects, or referred to by their string shortcuts. These components will be rendered in an accordion below the main input components. By default, no additional input components will be displayed.</p>
<div>additional_inputs_accordion</div> <div><i>str</i> <i>Accordion</i> <i>None</i></div> <div>default: None</div>	<p>If a string is provided, this is the label of the <code>gr.Accordion</code> to use to contain additional inputs. A <code>gr.Accordion</code> object can be provided as well to configure other properties of the container holding the additional inputs. Defaults to a <code>gr.Accordion(label="Additional Inputs", open=False)</code>. This parameter is only used if <code>additional_inputs</code> is provided.</p>
<div>submit_btn</div> <div><i>str</i> <i>Button</i></div> <div>default: "Submit"</div>	<p>The button to use for submitting inputs. Defaults to a <code>gr.Button("Submit", variant="primary")</code>. This parameter does not apply if the Interface is output-only, in which case the submit button always displays "Generate". Can be set to a string (which becomes the button label) or a <code>gr.Button</code> object (which allows for more customization).</p>
<div>stop_btn</div> <div><i>str</i> <i>Button</i></div> <div>default: "Stop"</div>	<p>The button to use for stopping the interface. Defaults to a <code>gr.Button("Stop", variant="stop", visible=False)</code>. Can be set to a string (which becomes the button label) or a <code>gr.Button</code> object (which allows for more customization).</p>
<div>clear_btn</div> <div><i>str</i> <i>Button</i></div> <div>default: "Clear"</div>	<p>The button to use for clearing the inputs. Defaults to a <code>gr.Button("Clear", variant="secondary")</code>. Can be set to a string (which becomes the button label) or a <code>gr.Button</code> object (which allows for more customization).</p>



```
import gradio as gr

def greet(name):
    return "Hello " + name + "!"

demo = gr.Interface(fn=greet, inputs="textbox", outputs="textbox")

if __name__ == "__main__":
    demo.launch()
```

Methods

launch

```
gradio.Interface.launch(...)
```

Description

Launches a simple web server that serves the demo. Can also be used to create a public link used by anyone to access the demo from their browser by setting share=True.

Example Usage

```
import gradio as gr
def reverse(text):
    return text[::-1]
demo = gr.Interface(reverse, "text", "text")
demo.launch(share=True, auth=("username", "password"))
```

Agruments

Parameter

Description



Parameter	Description
<div>inline</div> <div><i>bool</i> <i>None</i></div> <div>default: None</div>	whether to display in the interface inline in an iframe. Defaults to True in python notebooks; False otherwise.
<div>inbrowser</div> <div><i>bool</i></div> <div>default: False</div>	whether to automatically launch the interface in a new tab on the default browser.
<div>share</div> <div><i>bool</i> <i>None</i></div> <div>default: None</div>	whether to create a publicly shareable link for the interface. Creates an SSH tunnel to make your UI accessible from anywhere. If not provided, it is set to False by default every time, except when running in Google Colab. When localhost is not accessible (e.g. Google Colab), setting share=False is not supported.
<div>debug</div> <div><i>bool</i></div> <div>default: False</div>	if True, blocks the main thread from running. If running in Google Colab, this is needed to print the errors in the cell output.
<div>max_threads</div> <div><i>int</i></div> <div>default: 40</div>	the maximum number of total threads that the Gradio app can generate in parallel. The default is inherited from the starlette library (currently 40).
<div>auth</div> <div><i>Callable</i> <i>tuple[str, str]</i> <i>list[tuple[str, str]]</i> <i>None</i></div> <div>default: None</div>	If provided, username and password (or list of username-password tuples) required to access interface. Can also provide function that takes username and password and returns True if valid login.
<div>auth_message</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	If provided, HTML message provided on login page.



Parameter	Description
<div>prevent_thread_lock</div> <div><i>bool</i></div> <div>default: False</div>	If True, the interface will block the main thread while the server is running.
<div>show_error</div> <div><i>bool</i></div> <div>default: False</div>	If True, any errors in the interface will be displayed in an alert modal and printed in the browser console log
<div>server_name</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	to make app accessible on local network, set this to "0.0.0.0". Can be set by environment variable GRADIO_SERVER_NAME. If None, will use "127.0.0.1".
<div>server_port</div> <div><i>int</i> <i>None</i></div> <div>default: None</div>	will start gradio app on this port (if available). Can be set by environment variable GRADIO_SERVER_PORT. If None, will search for an available port starting at 7860.
<div>height</div> <div><i>int</i></div> <div>default: 500</div>	The height in pixels of the iframe element containing the interface (used if inline=True)
<div>width</div> <div><i>int</i> <i>str</i></div> <div>default: "100%"</div>	The width in pixels of the iframe element containing the interface (used if inline=True)
<div>favicon_path</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	If a path to a file (.png, .gif, or .ico) is provided, it will be used as the favicon for the web page.
<div>ssl_keyfile</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	If a path to a file is provided, will use this as the private key file to create a local server running on https.



Parameter	Description
<div>ssl_certfile</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	If a path to a file is provided, will use this as the signed certificate for https. Needs to be provided if ssl_keyfile is provided.
<div>ssl_keyfile_password</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	If a password is provided, will use this with the ssl certificate for https.
<div>ssl_verify</div> <div><i>bool</i></div> <div>default: True</div>	If False, skips certificate validation which allows self-signed certificates to be used.
<div>quiet</div> <div><i>bool</i></div> <div>default: False</div>	If True, suppresses most print statements.
<div>show_api</div> <div><i>bool</i></div> <div>default: True</div>	If True, shows the api docs in the footer of the app. Default True.
<div>allowed_paths</div> <div><i>list[str]</i> <i>None</i></div> <div>default: None</div>	List of complete filepaths or parent directories that gradio is allowed to serve (in addition to the directory containing the gradio python file). Must be absolute paths. Warning: if you provide directories, any files in these directories or their subdirectories are accessible to all users of your app.
<div>blocked_paths</div> <div><i>list[str]</i> <i>None</i></div> <div>default: None</div>	List of complete filepaths or parent directories that gradio is not allowed to serve (i.e. users of your app are not allowed to access). Must be absolute paths. Warning: takes precedence over <code>allowed_paths</code> and all other directories exposed by Gradio by default.



Parameter	Description
<div>root_path</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	The root path (or "mount point") of the application, if it's not served from the root ("/") of the domain. Often used when the application is behind a reverse proxy that forwards requests to the application. For example, if the application is served at "https://example.com/myapp", the <code>root_path</code> should be set to "/myapp". Can be set by environment variable GRADIO_ROOT_PATH. Defaults to "".
<div>app_kwargs</div> <div><i>dict[str, Any]</i> <i>None</i></div> <div>default: None</div>	Additional keyword arguments to pass to the underlying FastAPI app as a dictionary of parameter keys and argument values. For example, <code>"docs_url": "/docs"</code>
<div>state_session_capacity</div> <div><i>int</i></div> <div>default: 10000</div>	The maximum number of sessions whose information to store in memory. If the number of sessions exceeds this number, the oldest sessions will be removed. Reduce capacity to reduce memory usage when using gradio.State or returning updated components from functions. Defaults to 10000.
<div>share_server_address</div> <div><i>str</i> <i>None</i></div> <div>default: None</div>	Use this to specify a custom FRP server and port for sharing Gradio apps (only applies if share=True). If not provided, will use the default FRP server at https://gradio.live. See https://github.com/huggingface/frp for more information.
<div>share_server_protocol</div> <div><i>Literal[('http', 'https')]</i> <i>None</i></div> <div>default: None</div>	Use this to specify the protocol to use for the share links. Defaults to "https", unless a custom share_server_address is provided, in which case it defaults to "http". If you are using a custom share_server_address and want to use https, you must set this to "https".

load

```
gradio.Interface.load(block, ...)
```



Description

This listener is triggered when the Interface initially loads in the browser.

Agruments

Parameter	Description
<div>block</div> <div><i>Block</i> <i>None</i></div> <div>required</div>	
<div>fn</div> <div><i>Callable</i> <i>None</i> <i>Literal['decorator']</i></div> <div>default: "decorator"</div>	the function to call when this event is triggered. Often a machine learning model's prediction function. Each parameter of the function corresponds to one input component, and the function should return a single value or a tuple of values, with each element in the tuple corresponding to one output component.
<div>inputs</div> <div><i>Component</i> <i>list[Component]</i> <i>set[Component]</i> <i>None</i></div> <div>default: None</div>	List of gradio.components to use as inputs. If the function takes no inputs, this should be an empty list.
<div>outputs</div> <div><i>Component</i> <i>list[Component]</i> <i>None</i></div> <div>default: None</div>	List of gradio.components to use as outputs. If the function returns no outputs, this should be an empty list.
<div>api_name</div> <div><i>str</i> <i>None</i> <i>Literal[False]</i></div> <div>default: None</div>	defines how the endpoint appears in the API docs. Can be a string, None, or False. If set to a string, the endpoint will be exposed in the API docs with the given name. If None (default), the name of the function will be used as the API endpoint. If False, the endpoint will not be exposed in the API docs and downstream apps (including those that <code>gr.load</code> this app) will not be able to use this event.



Parameter	Description
<div>scroll_to_output</div> <div><i>bool</i></div> <div>default: False</div>	If True, will scroll to output component on completion
<div>show_progress</div> <div><i>Literal[('full', 'minimal', 'hidden')]</i></div> <div>default: "full"</div>	If True, will show progress animation while pending
<div>queue</div> <div><i>bool None</i></div> <div>default: None</div>	If True, will place the request on the queue, if the queue has been enabled. If False, will not put this event on the queue, even if the queue has been enabled. If None, will use the queue setting of the gradio app.
<div>batch</div> <div><i>bool</i></div> <div>default: False</div>	If True, then the function should process a batch of inputs, meaning that it should accept a list of input values for each parameter. The lists should be of equal length (and be up to length <code>max_batch_size</code>). The function is then <i>required</i> to return a tuple of lists (even if there is only 1 output component), with each list in the tuple corresponding to one output component.
<div>max_batch_size</div> <div><i>int</i></div> <div>default: 4</div>	Maximum number of inputs to batch together if this is called from the queue (only relevant if batch=True)
<div>preprocess</div> <div><i>bool</i></div> <div>default: True</div>	If False, will not run preprocessing of component data before running 'fn' (e.g. leaving it as a base64 string if this method is called with the <code>Image</code> component).
<div>postprocess</div> <div><i>bool</i></div> <div>default: True</div>	If False, will not run postprocessing of component data before returning 'fn' output to the browser.



Parameter	Description
<div>cancels</div> <div><i>dict[str, Any] list[dict[str, Any]] None</i></div> <div>default: None</div>	<p>A list of other events to cancel when this listener is triggered.</p> <p>For example, setting <code>cancels=[click_event]</code> will cancel the <code>click_event</code>, where <code>click_event</code> is the return value of another components <code>.click</code> method. Functions that have not yet run (or generators that are iterating) will be cancelled, but functions that are currently running will be allowed to finish.</p>
<div>every</div> <div><i>float None</i></div> <div>default: None</div>	<p>Run this event 'every' number of seconds while the client connection is open. Interpreted in seconds. Queue must be enabled.</p>
<div>trigger_mode</div> <div><i>Literal[('once', 'multiple', 'always_last')] None</i></div> <div>default: None</div>	<p>If "once" (default for all events except <code>.change()</code>) would not allow any submissions while an event is pending. If set to "multiple", unlimited submissions are allowed while pending, and "always_last" (default for <code>.change()</code> event) would allow a second submission after the pending event is complete.</p>
<div>js</div> <div><i>str None</i></div> <div>default: None</div>	<p>Optional frontend js method to run before running 'fn'. Input arguments for js method are values of 'inputs' and 'outputs', return should be a list of values for output components.</p>
<div>concurrency_limit</div> <div><i>int None Literal['default']</i></div> <div>default: "default"</div>	<p>If set, this is the maximum number of this event that can be running simultaneously. Can be set to None to mean no concurrency_limit (any number of this event can be running simultaneously). Set to "default" to use the default concurrency limit (defined by the <code>default_concurrency_limit</code> parameter in <code>Blocks.queue()</code>, which itself is 1 by default).</p>
<div>concurrency_id</div> <div><i>str None</i></div> <div>default: None</div>	<p>If set, this is the id of the concurrency group. Events with the same concurrency_id will be limited by the lowest set concurrency_limit.</p>



Parameter

Description

`show_api`

bool

default: True

whether to show this event in the "view API" page of the Gradio app, or in the ".view_api()" method of the Gradio clients. Unlike setting `api_name` to False, setting `show_api` to False will still allow downstream apps to use this event. If `fn` is None, `show_api` will automatically be set to False.

from_pipeline

```
gradio.Interface.from_pipeline(pipeline, ...)
```

Description

Class method that constructs an Interface from a Hugging Face transformers.Pipeline object. The input and output components are automatically determined from the pipeline.

Example Usage

```
import gradio as gr
from transformers import pipeline
pipe = pipeline("image-classification")
gr.Interface.from_pipeline(pipe).launch()
```

Agruments

Parameter

Description

`pipeline`

Pipeline

required

the pipeline object to use.

integrate

Description

A catch-all method for integrating with other libraries. This method should be run after `launch()`

Agruments

Parameter	Description
<code>comet_ml</code> <i><class 'inspect._empty'></i> default: None	If a <code>comet_ml</code> Experiment object is provided, will integrate with the experiment and appear on Comet dashboard
<code>wandb</code> <i>ModuleType None</i> default: None	If the <code>wandb</code> module is provided, will integrate with it and appear on WandB dashboard
<code>mlflow</code> <i>ModuleType None</i> default: None	If the <code>mlflow</code> module is provided, will integrate with the experiment and appear on ML Flow dashboard

queue

`gradio.Interface.queue(...)`

Description

By enabling the queue you can control when users know their position in the queue, and set a limit on maximum number of events allowed.

Example Usage

```
demo = gr.Interface(image_generator, gr.Textbox(), gr.Image())
demo.queue(max_size=20)
demo.launch()
```




Parameter	Description
<div>status_update_rate</div> <div><i>float Literal['auto']</i></div> <div>default: "auto"</div>	If "auto", Queue will send status estimations to all clients whenever a job is finished. Otherwise Queue will send status at regular intervals set by this parameter as the number of seconds.
<div>api_open</div> <div><i>bool None</i></div> <div>default: None</div>	If True, the REST routes of the backend will be open, allowing requests made directly to those endpoints to skip the queue.
<div>max_size</div> <div><i>int None</i></div> <div>default: None</div>	The maximum number of events the queue will store at any given moment. If the queue is full, new events will not be added and a user will receive a message saying that the queue is full. If None, the queue size will be unlimited.
<div>concurrency_count</div> <div><i>int None</i></div> <div>default: None</div>	Deprecated. Set the concurrency_limit directly on event listeners e.g. <code>btn.click(fn, ..., concurrency_limit=10)</code> or <code>gr.Interface(concurrency_limit=10)</code> . If necessary, the total number of workers can be configured via <code>max_threads</code> in <code>launch()</code> .
<div>default_concurrency_limit</div> <div><i>int None Literal['not_set']</i></div> <div>default: "not_set"</div>	The default value of <code>concurrency_limit</code> to use for event listeners that don't specify a value. Can be set by environment variable <code>GRADIO_DEFAULT_CONCURRENCY_LIMIT</code> . Defaults to 1 if not set otherwise.



Key Features

Sharing Your App

Interface State

Reactive Interfaces

ChatInterface →

