

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

See the Release History

← File

Gallery →

FileExplorer

gradio.FileExplorer(...)

Description

Creates a file explorer component that allows users to browse and select files on the machine hosting the Gradio app.

Behavior

As input: passes the selected file or directory as a str path (relative to root) or list[str] depending on file_count

As output: expects function to return a str path to a file, or List[str] consisting of paths to files.

Initialization

Parameter Description

glob str

default: "**/*.*"

The glob-style pattern used to select which files to display, e.g. "" to match all files, ".png" to match all .png files, "/*.txt" to match any .txt file in any subdirectory, etc. The default value matches all files and folders recursively. See the Python glob documentation at https://docs.python.org/3/library/glob.html for more information.

value

str | list[str] | Callable | None

default: None

The file (or list of files, depending on the file_count parameter) to show as "selected" when the component is first loaded. If a callable is provided, it will be called when the app loads to set the initial value of the component. If not provided, no files are shown as selected.

imeter	Description
file_count Literal[('single', 'multiple')] default: "multiple"	Whether to allow single or multiple files to be selected. If "single", the component will return a single absolute file path as a string. If "multiple", the component will return a list of absolute file paths as a list of strings.
root_dir str Path default: "."	Path to root directory to select files from. If not provided, defaults to current working directory.
ignore_glob str None default: None	The glob-tyle pattern that will be used to exclude files from the list. For example, "*.py" will exclude all .py files from the list. See the Python glob documentation at https://docs.python.org/3/library/glob.html for more information.
label str None default: None	The label for this component. Appears above the component and is also used as the header if there are a table of examples for this component. If None and used in a <code>gr.Interface</code> , the label will be the name of the parameter this component is assigned to.
every float None default: None	If value is a callable, run the function 'every' number of seconds while the client connection is open. Has no effect otherwise. Queue must be enabled. The event can be accessed (e.g. to cancel it) via this component's .load_event attribute.
show_label bool None default: None	if True, will display label.
container bool default: True	If True, will place the component in a container - providing some extra padding around the border.

ımeter	Description
scale	relative width compared to adjacent Components in a Row. For
int None	example, if Component A has scale=2, and Component B has
default: None	scale=1, A will be twice as wide as B. Should be an integer.
min_width	minimum pixel width, will wrap if not sufficient screen space to
int	satisfy this value. If a certain scale value results in this
default: 160	Component being narrower than min_width, the min_width
	parameter will be respected first.
height	The maximum height of the file component, specified in pixels
int float None	a number is passed, or in CSS units if a string is passed. If more
default: None	files are uploaded than can fit in the height, a scrollbar will
	appear.
interactive	if True, will allow users to upload a file; if False, can only be use
bool None	to display files. If not provided, this is inferred based on whether
default: None	the component is used as an input or output.
visible	If False, component will be hidden.
bool	
default: True	
elem_id	An optional string that is assigned as the id of this component i
str None	the HTML DOM. Can be used for targeting CSS styles.
default: None	
elem_classes	An optional list of strings that are assigned as the classes of this
list[str] str None	component in the HTML DOM. Can be used for targeting CSS
default: None	styles.
render	If False, component will not render be rendered in the Blocks
bool	context. Should be used if the intention is to assign event
	listeners now but render the component later.

= imeter	Description
root	
None	
default: None	

Shortcuts

Class	Interface String Shortcut	Initialization	
gradio.FileExplorer	"fileexplorer"	Uses default values	

Demos

Event Listeners

Description

Event listeners allow you to capture and respond to user interactions with the UI components you've defined in a Gradio Blocks app. When a user interacts with an element, such as changing a slider value or uploading an image, a function is called.

Supported Event Listeners

The FileExplorer component supports the following event listeners. Each event listener takes the same parameters, which are listed in the Event Arguments table below.

gradio.FileExplorer.change(fn,

...)

Event Arguments

Parameter

Description

fn

Callable | None | Literal['decorator']

default: "decorator"

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.

inputs

Component | list[Component] |
set[Component] | None

default: None

List of gradio.components to use as inputs. If the function takes no inputs, this should be an empty list.

outputs

Component | list[Component] | None

default: None

List of gradio.components to use as outputs. If the function returns no outputs, this should be an empty list.

api_name

str | None | Literal[False]

default: None

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

scroll_to_output

bool

default: False

If True, will scroll to output component on completion

Parameter	Description
show_progress Literal[('full', 'minimal', 'hidden')] default: "full"	If True, will show progress animation while pending
queue bool None default: None	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.
batch bool default: False	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 max_batch_size). 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.
<pre>max_batch_size int default: 4</pre>	Maximum number of inputs to batch together if this is called from the queue (only relevant if batch=True)
preprocess bool default: True	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).
postprocess	If False, will not run postprocessing of component data

bool

default: True

before returning 'fn' output to the browser.

Parameter	Description
cancels dict[str, Any] list[dict[str, Any]] None default: None	A list of other events to cancel when this listener is triggered. For example, setting cancels=[click_event] will cancel the click_event, where click_event is the return value of another components .click 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.
every float None default: None	Run this event 'every' number of seconds while the client connection is open. Interpreted in seconds. Queue must be enabled.
<pre>trigger_mode Literal[('once', 'multiple', 'always_last')] None default: None</pre>	If "once" (default for all events except .change()) 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 .change() event) would allow a second submission after the pending event is complete.
js str None default: None	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.
<pre>concurrency_limit int None Literal['default'] default: "default"</pre>	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 default_concurrency_limit parameter in Blocks.queue () , which itself is 1 by default).
concurrency_id str None default: None	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.



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.

← File

Gallery →



