Job

```
gradio.Job(future, ...)
```

Description

A Job is a wrapper over the Future class that represents a prediction call that has been submitted by the Gradio client. This class is not meant to be instantiated directly, but rather is created by the Client.submit() method.

A Job object includes methods to get the status of the prediction call, as well to get the outputs of the prediction call. Job objects are also iterable, and can be used in a loop to get the outputs of prediction calls as they become available for generator endpoints.

Initialization

Parameter	Description
`future` *Future* ***required***	The future object that represents the prediction call, created by the Client.submit() method
`communicator` *Communicator | None* **default: None**	The communicator object that is used to communicate between the client and the background thread running the job
`verbose` *bool* **default: True**	Whether to print any status-related messages to the console

ımeter	Description
`space_id` *str	None* **default: None**	The space ID corresponding to the Client object that created this Job object

Methods

result

`gradio.Job.result(...)`

Description

Return the result of the call that the future represents. Raises CancelledError: If the future was cancelled, TimeoutError: If the future didn't finish executing before the given timeout, and Exception: If the call raised then that exception will be raised.

Example Usage

```
from gradio_client import Client
calculator = Client(src="gradio/calculator")
job = calculator.submit("foo", "add", 4, fn_index=0)
job.result(timeout=5)
>> 9
```

Agruments

Parameter	Description
`timeout` *float	None* **default: None**	The number of seconds to wait for the result if the future isn't done. If None, then there is no limit on the wait time.

outputs

`gradio.Job.outputs(...)`

Description

Returns a list containing the latest outputs from the Job.
 If the endpoint has multiple output components, the list will contain a tuple of results. Otherwise, it will contain the results without storing them in tuples.
 For endpoints that are queued, this list will contain the final job output even if that endpoint does not use a generator function.

Example Usage

```
from gradio_client import Client
client = Client(src="gradio/count_generator")
job = client.submit(3, api_name="/count")
while not job.done():
    time.sleep(0.1)
job.outputs()
>> ['0', '1', '2']
```

status

```
gradio.Job.status(...)
```

Description

Returns the latest status update from the Job in the form of a StatusUpdate object, which contains the following fields: code, rank, queue_size, success, time, eta, and progress_data.
 progress_data is a list of updates emitted by the gr.Progress() tracker of the event handler. Each element of the list has the following fields: index, length, unit, progress, desc. If the event handler does not have a gr.Progress() tracker, the progress_data field will be None.

Example Usage

```
from gradio_client import Client
client = Client(src="gradio/calculator")
job = client.submit(5, "add", 4, api_name="/predict")
job.status()
>> <Status.STARTING: 'STARTING'>
job.status().eta
>> 43.241  # seconds
```