tfdbg.GrpcDebugWrapperSession

Class **`GrpcDebugWrapperSession`**

Defined in `tensorflow/python/debug/wrappers/grpc_wrapper.py`.

Debug Session wrapper that send debug data to gRPC stream(s).

Properties

`graph`

`graph_def`

`run_call_count`

`sess_str`

`session`

Methods

`__init__`

```
__init__(
    sess,
    grpc_debug_server_addresses,
    watch_fn=None,
    thread_name_filter=None,
    log_usage=True
)
```

Constructor of DumpingDebugWrapperSession.

Args:

- `sess`: The TensorFlow `Session` object being wrapped.
- `grpc_debug_server_addresses`: (`str` or `list` of `str`) Single or a list of the gRPC debug server addresses, in the format of , without the "grpc://" prefix. For example: "localhost:7000", ["localhost:7000", "192.168.0.2:8000"]

- `watch_fn`: (`Callable`) A Callable that can be used to define per-run debug ops and watched tensors. See the doc of `NonInteractiveDebugWrapperSession.__init__()` for details.
- `thread_name_filter`: Regular-expression white list for threads on which the wrapper session will be active. See doc of `BaseDebugWrapperSession` for more details.
- `log_usage`: (`bool`) whether the usage of this class is to be logged.

Raises:

- `TypeError`: If `grpc_debug_server_addresses` is not a `str` or a `list` of `str`.

`__enter__`

```
__enter__()
```

`__exit__`

```
__exit__(
    exec_type,
    exec_value,
    exec_tb
)
```

`as_default`

```
as_default()
```

`close`

```
close()
```

`increment_run_call_count`

```
increment_run_call_count()
```

`invoke_node_stepper`

```
invoke_node_stepper(
    node_stepper,
    restore_variable_values_on_exit=True
)
```

See doc of BaseDebugWrapperSession.invoke_node_stepper.

`list_devices`

```
list_devices(
    *args,
    **kwargs
)
```

`make_callable`

```
make_callable(
    fetches,
    feed_list=None,
    accept_options=False
)
```

`on_run_end`

```
on_run_end(request)
```

See doc of BaseDebugWrapperSession.on_run_end.

`on_run_start`

```
on_run_start(request)
```

See doc of BaseDebugWrapperSession.on_run_start.

`on_session_init`

```
on_session_init(request)
```

See doc of BaseDebugWrapperSession.on_run_start.

`partial_run`

```
partial_run(
    handle,
    fetches,
    feed_dict=None
)
```

`partial_run_setup`

```
partial_run_setup(
    fetches,
    feeds=None
)
```

Sets up the feeds and fetches for partial runs in the session.

`prepare_run_debug_urls`

```
prepare_run_debug_urls(
    fetches,
    feed_dict
)
```

Implementation of abstract method in superclass.

See doc of `NonInteractiveDebugWrapperSession.prepare_run_debug_urls()` for details.

Args:

- `fetches` : Same as the `fetches` argument to `Session.run()`
- `feed_dict` : Same as the `feed_dict` argument to `Session.run()`

Returns:

- `debug_urls` : (`str` or `list` of `str`) file:// debug URLs to be used in this `Session.run()` call.

`reset`

```
reset(
    *args,
    **kwargs
)
```

`run`

```
run(
    fetches,
    feed_dict=None,
    options=None,
    run_metadata=None,
    callable_runner=None,
    callable_runner_args=None
)
```

Wrapper around Session.run() that inserts tensor watch options.

Args:

- `fetches` : Same as the `fetches` arg to regular `Session.run()` .
- `feed_dict` : Same as the `feed_dict` arg to regular `Session.run()` .
- `options` : Same as the `options` arg to regular `Session.run()` .
- `run_metadata` : Same as the `run_metadata` arg to regular `Session.run()` .
- `callable_runner` : A `callable` returned by `Session.make_callable()` . If not `None` , `fetches` and `feed_dict` must both be `None` .
- `callable_runner_args` : An optional list of arguments to `callable_runner` .

Returns:

Simply forwards the output of the wrapped `Session.run()` call.

Raises:

- `ValueError` : On invalid `OnRunStartAction` value. Or if `callable_runner` is not `None` and either or both of `fetches` and `feed_dict` is `None` .

`should_stop`

```
should_stop()
```