StreamSession Interface

Interface for managing media streaming sessions with Meta Wearables devices.

A StreamSession handles video streaming and photo capture functionality from AI glasses. Sessions are created and automatically started via Wearables.startStreamSession. Video frames are delivered through the StreamSession.videoStream Flow and photos through the StreamSession.capturePhoto suspend function.

Sessions automatically connect when a device becomes available and stop when the device session ends externally (e.g., device powered off).

Signature

```
interface StreamSession
```

Properties

state : StateFlow<StreamSessionState> [Get]

The current state of the streaming session.
State transitions:

- STOPPED -> STARTING -> STARTED -> STREAMING (when device connects and streaming begins)
- STREAMING -> STOPPING -> STOPPED (on device disconnect or error)
- Any state -> STOPPING -> STOPPED -> CLOSED (on close)

Signature

```
abstract val state:
StateFlow<StreamSessionState>
```

e>
[Get]

STREAMING state. The flow automatically handles buffer overflow by dropping the oldest frames to ensure smooth streaming.

Signature

```
abstract val videoStream:
Flow<VideoFrame>
```

Functions

capturePhoto ()

Captures a still photo during streaming.
Triggers a photo capture while video streaming is active. The captured photo is delivered as the return value of this suspend function.

Signature

```
abstract suspend fun capturePhoto():
Result<PhotoData>
```

Returns

`Result<PhotoData>` Result containing PhotoData on success, or failure if no device session is active, a capture is already in progress, or the current state is not STREAMING.

close ()

Stops video streaming and releases all resources.
Shuts down the streaming pipeline and transitions to CLOSED state. The session cannot be reused after closing. State transitions: Any state -> STOPPING -> STOPPED -> CLOSED

Signature

```
abstract fun close()
```