epiphan video

Precondition

Given a video recording device accessible through SSH using:

ssh operator@virtual-device.local

The device has the following API endpoints (example attached - see device_api.py):

- **POST /channel/<id>/recording/config**
  - **Description**: Sets the recording configuration for a specific channel.
  - **Request Body:** JSON object containing configuration settings, like: *{"format": "mp4", "resolution": "1080p"}*
  - **Response**: JSON object indicating the status of the operation.

- **GET /channel/<id>/recording/config**
  - **Description**: Retrieves the recording configuration for a specific channel.
  - **Response**: JSON object containing the current configuration settings for the specified channel.

- **POST /channel/<id>/recording/start**
  - **Description**: Starts recording on a specific channel.
  - **Response**: JSON object indicating the status of the operation.

- **POST /channel/<id>/recording/stop**
  - **Description**: Stops recording on a specific channel.
  - **Response**: JSON object indicating the status of the operation and the file path of the recorded file.

- **GET /channel/<id>/recording/files**
  - **Description**: Retrieves the list of recording files for a specific channel.

- **Response**: JSON array containing the file paths of the recordings for the specified channel.

After using the /start and /stop endpoints, the device stores files locally (accessible via SSH) at:

/tmp/recordings/<channel-id>/<yyyy-mm-dd>/<file-number>-<timestamp_start>-<timestamp_end>.<file_format>

Examples:

/tmp/recordings/1/2024-09-01/1-1725386400-1725390000.mp4
/tmp/recordings/2/2024-09-01/2-1725393600-1725395400.avi

Task

Create automation tests using Python and Pytest to validate the device recording and encoding functionality using the following scenario:

Note: Use ***@pytest*** fixtures and organize your code to be modular and reusable.

1. Start a mock device API (in attachments) that mimics the behavior of the actual device.
2. Use different channel resolutions and formats to test, such as:

   `{"format": "mp4", "resolution": "1080p"}`

   `{"format": "avi", "resolution": "720p"}`

   `{"format": "mov", "resolution": "1080i"}`

3. Start recording.

4. Wait for a few seconds and then stop recording on the same channel.
5. Get the file location and retrieve it using a “fake” SSH connection to your local PC (use video file from the sample_video folder for this task).
6. Validate that the recorded file exists and has the expected format and resolution.

   Note: In your case, the test should fail for some durations/formats because you will use only one sample file, so you can mark it as expected to fail.
7. Validate that the duration of the recorded file is as recorded.