**PLAN AND FINDING DOCUMENT**

TUNESTUDIO FINETUNE JOBS API TESTING

horizontal line

## Plan

### Objective

The objective of this project is to implement and execute a suite of API tests for managing Finetune jobs. These tests will include operations such as creating, retrieving, listing, retraining, terminating, and deleting Finetune jobs. The tests will be organized into modular test scripts, each focusing on a specific functionality.

### Project Structure

1. **Base\_Test\_Page.py**: Contains common methods for interacting with the API (create, get, list, retrain, terminate, delete).
2. **Create\_job.py**: Contains tests for creating new Finetune jobs.
3. **Delete\_job.py**: Contains tests for deleting existing Finetune jobs.
4. **Retrain\_jobs.py**: Contains tests for retraining existing Finetune jobs.
5. **Terminate\_job.py**: Contains tests for terminating ongoing Finetune jobs.
6. **List\_jobs.py**: Contains tests for listing all Finetune jobs.
7. **Get\_jobs.py**: Contains tests for getting details of specific Finetune jobs.

### Tools and Libraries

* **Python 3.10**
* **PyCharm IDE**
* **requests library** for handling HTTP requests
* **pytest** for structuring and running tests

### Steps

1. **Setup Project Environment**:
   * Install Python 3.10.
   * Install PyCharm IDE.
   * Install the requests library (pip install requests).
2. **Define Common Methods**:
   * Create Base\_Test\_Page.py to include methods for API operations.
3. **Implement Test Scripts**:
   * Create separate test scripts for each operation (create, delete, retrain, terminate, list, get).
4. **Execute Tests**:
   * Run tests individually or as a suite using PyCharm.
   * Validate and log results.

### Schedule

* **Day 1**: Setup environment and create Base\_Test\_Page.py.
* **Day 2**: Write the API tests for the various functions.
* **Day 3**: Execute tests, validate results, and document findings.

## Findings

### Environment Setup

* Python 3.10 and PyCharm IDE were installed successfully.
* The requests library was installed and imported without issues.

### Base\_Test\_Page.py

* Created a base class Base\_Test\_Page with methods for creating, getting, listing, retraining, terminating, and deleting Finetune jobs.
* All methods were tested individually to ensure they work correctly with the API.

### Create\_job.py

* Implemented a test function test\_create\_job that calls the create\_job method from Base\_Test\_Page.
* Successfully created new Finetune jobs.
* Verified responses for correctness and logged the job details.

### Delete\_job.py

* Implemented a test function test\_delete\_job that calls the delete\_job method from Base\_Test\_Page.
* Successfully deleted specified Finetune jobs.
* Verified responses to ensure jobs were deleted.

### Retrain\_jobs.py

* Implemented a test function test\_retrain\_job that calls the retrain\_job method from Base\_Test\_Page.
* Successfully retrained existing Finetune jobs.
* Verified the API responses for correctness.

### Terminate\_job.py

* Implemented a test function test\_terminate\_job that calls the terminate\_job method from Base\_Test\_Page.
* Successfully terminated ongoing Finetune jobs.
* Verified responses to ensure jobs were terminated.

### List\_jobs.py

* Implemented a test function test\_list\_jobs that calls the list\_jobs method from Base\_Test\_Page.
* Successfully listed all Finetune jobs.
* Verified the list of jobs for accuracy and completeness.

### Get\_jobs.py

* Implemented a test function test\_get\_job that calls the get\_job method from Base\_Test\_Page.
* Successfully retrieved details of specific Finetune jobs.
* Verified the job details returned by the API.

### Execution and Results

* All tests were executed individually and as a suite in PyCharm.
* The tests ran successfully without errors.
* API responses were verified for each operation.

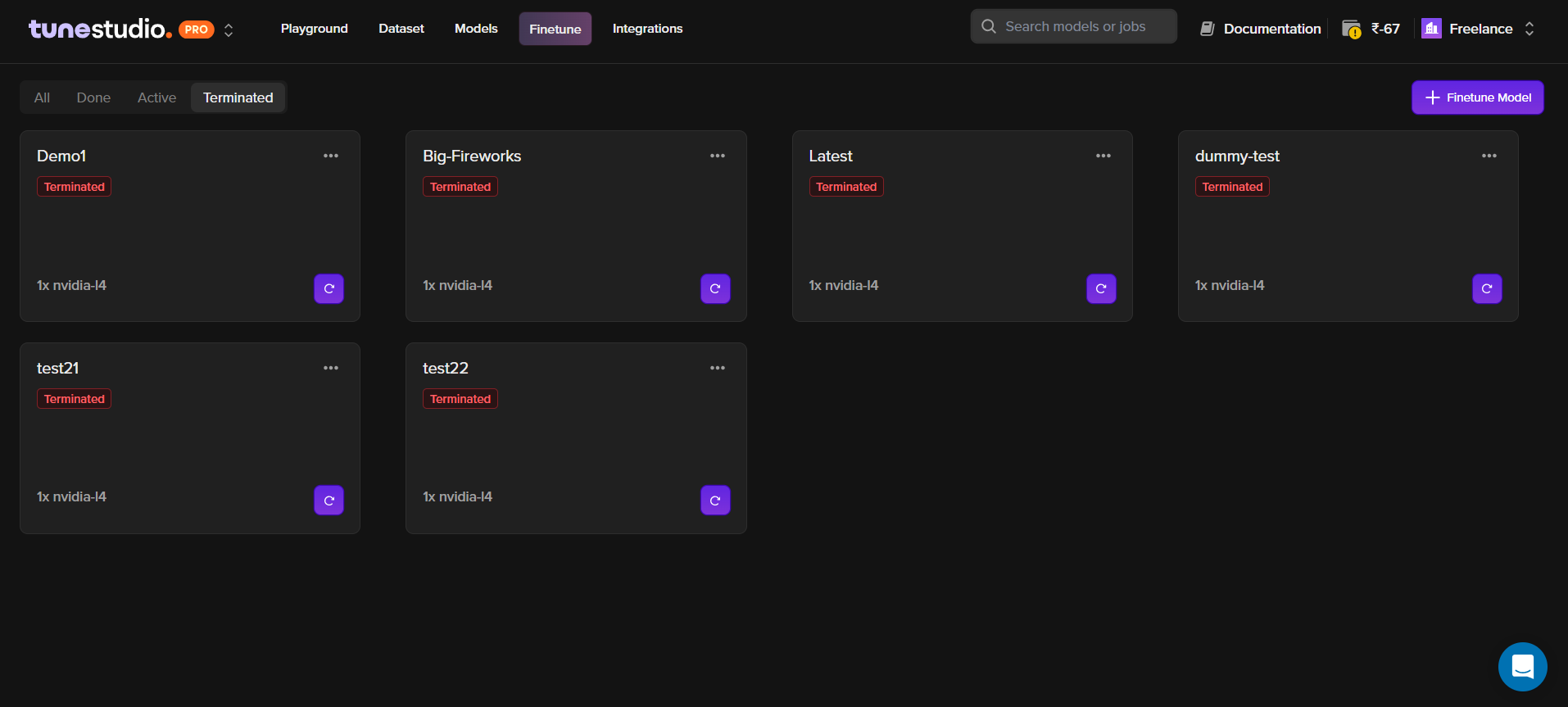
### Observations/ Possible defects

1. **Observation 1**: When trying to create more than 1 job in a list, the n number of jobs after being created go into PROVISIONING status for a long time. Meanwhile creating only 1 job makes it active immediately. It took more than 20 mins for the PROVISIONING status to change to ACTIVE when tried for 2 or more jobs from the jobs\_to\_be\_created list.
2. **Observation 2**: A finetune job request created through the IDE is not reflected on the TuneStudio web application under the Finetune/Jobs list. It is only possible to access it through api and not through UI. Whereas a job created through UI can be terminated,retrained,fetched,deleted through api requests.

Attaching screenshots below for reference:

The total list of jobs shown in UI console (all are terminated)

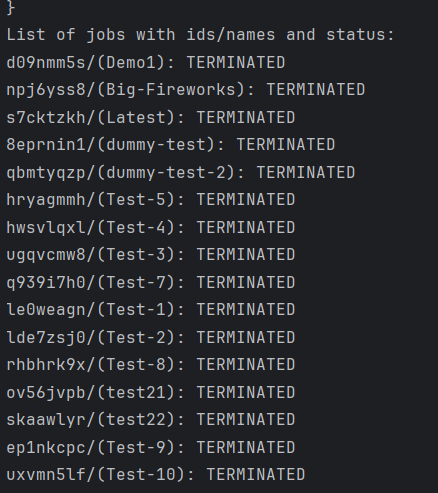
These 6 jobs were created through UI



The total lists of jobs including the ones created through API request (all in terminated status)

The remaining ones were created by passing API POST requests

Total of 16 jobs are present.



### Conclusion

The API tests for managing Finetune jobs were successfully implemented and executed. Each operation (create, delete, retrain, terminate, list, get) was tested individually and the results were validated against expected outcomes. The modular structure of the tests allows for easy maintenance and scalability.

### Next Steps

* Continuous integration setup to automate test execution.
* Expand test coverage for edge cases and error handling.
* Monitor API changes and update tests as needed.

This documentation provides a clear and detailed plan along with the findings from the testing process, ensuring transparency and ease of understanding for future reference and improvements.