

Storage Module Sequence Diagrams

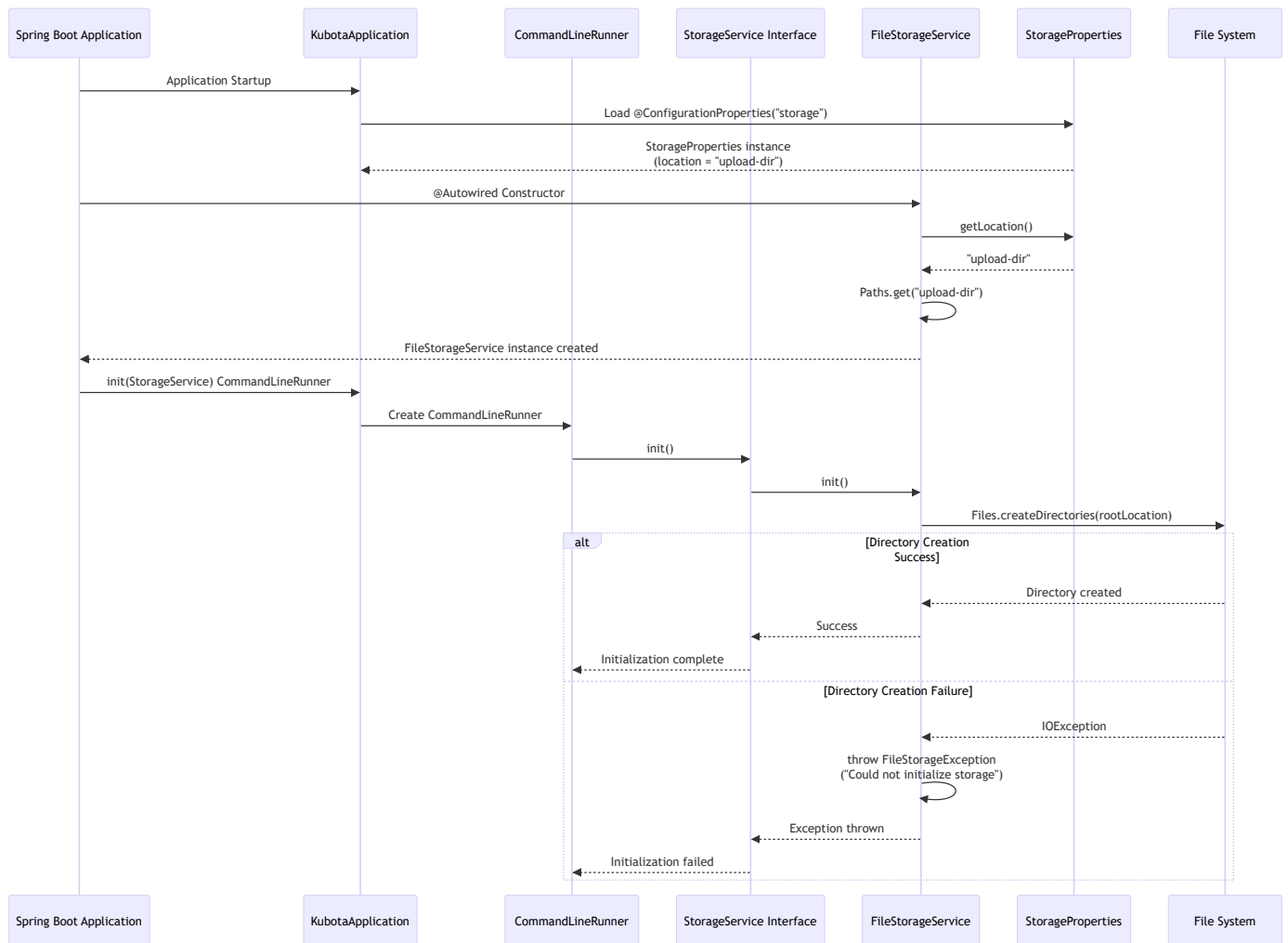
This document contains detailed sequence diagrams for the Storage Module of the KUBOTA Backend application.

Table of Contents

1. [Module Initialization Flow](#)
2. [File Upload Flow \(MultipartFile\)](#)
3. [Base64 Image Upload Flow](#)
4. [File Retrieval Flow](#)
5. [File Listing Flow](#)
6. [File Deletion Flow](#)
7. [Error Handling Flow](#)

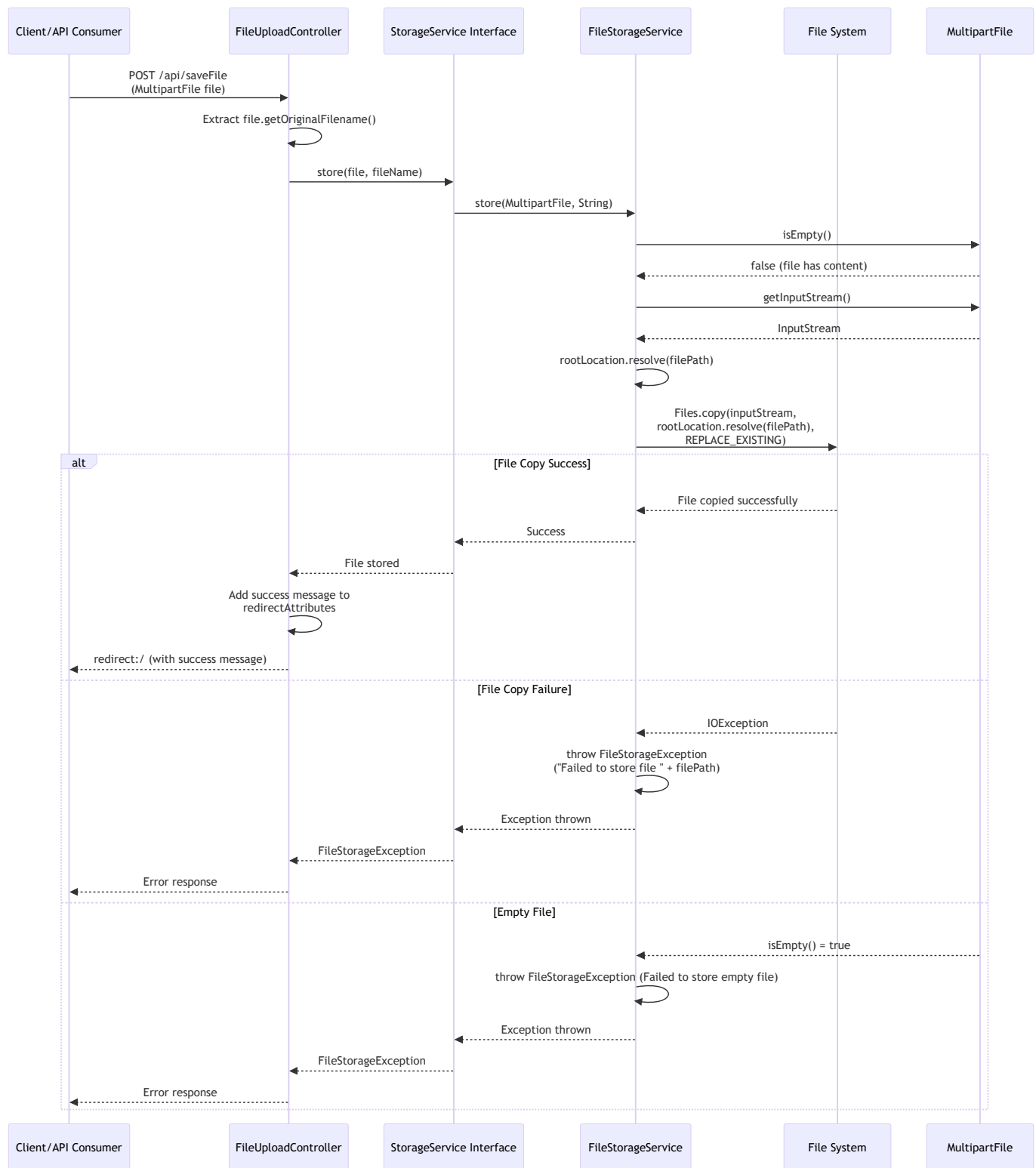
1. Module Initialization Flow

This diagram shows how the Storage Module is initialized when the application starts.



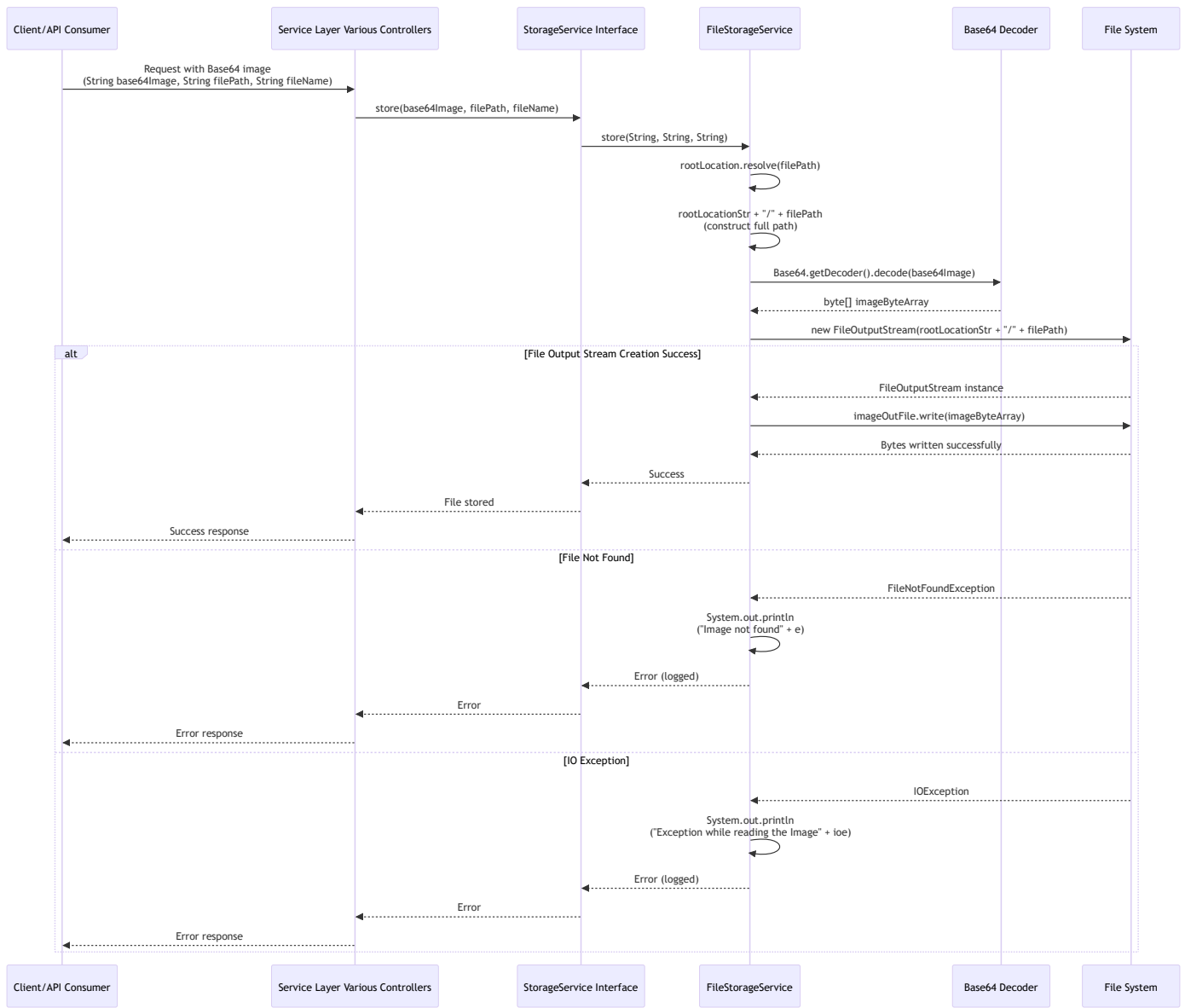
2. File Upload Flow (MultipartFile)

This diagram shows the complete flow when uploading a file via MultipartFile.



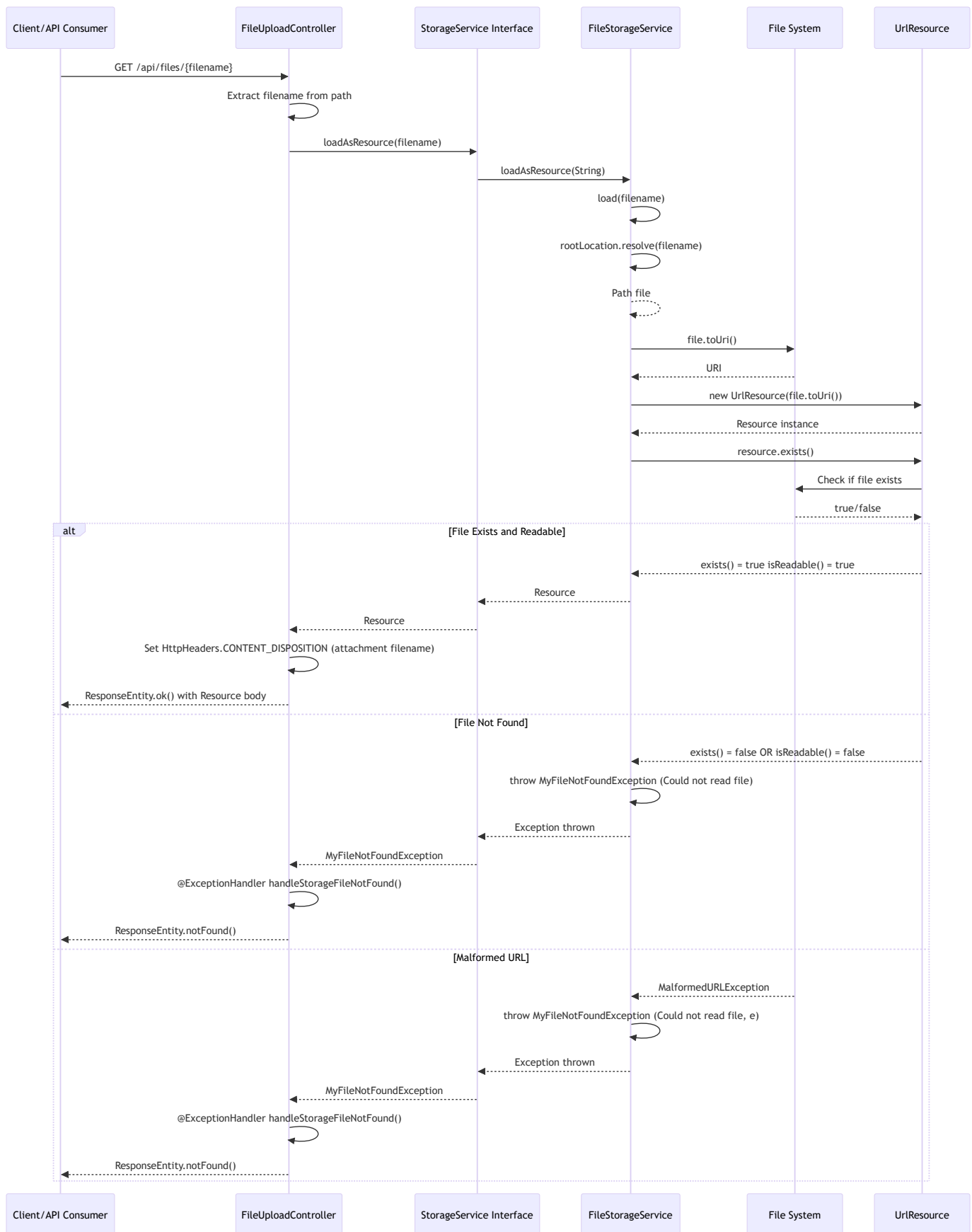
3. Base64 Image Upload Flow

This diagram shows the flow when uploading a Base64 encoded image.



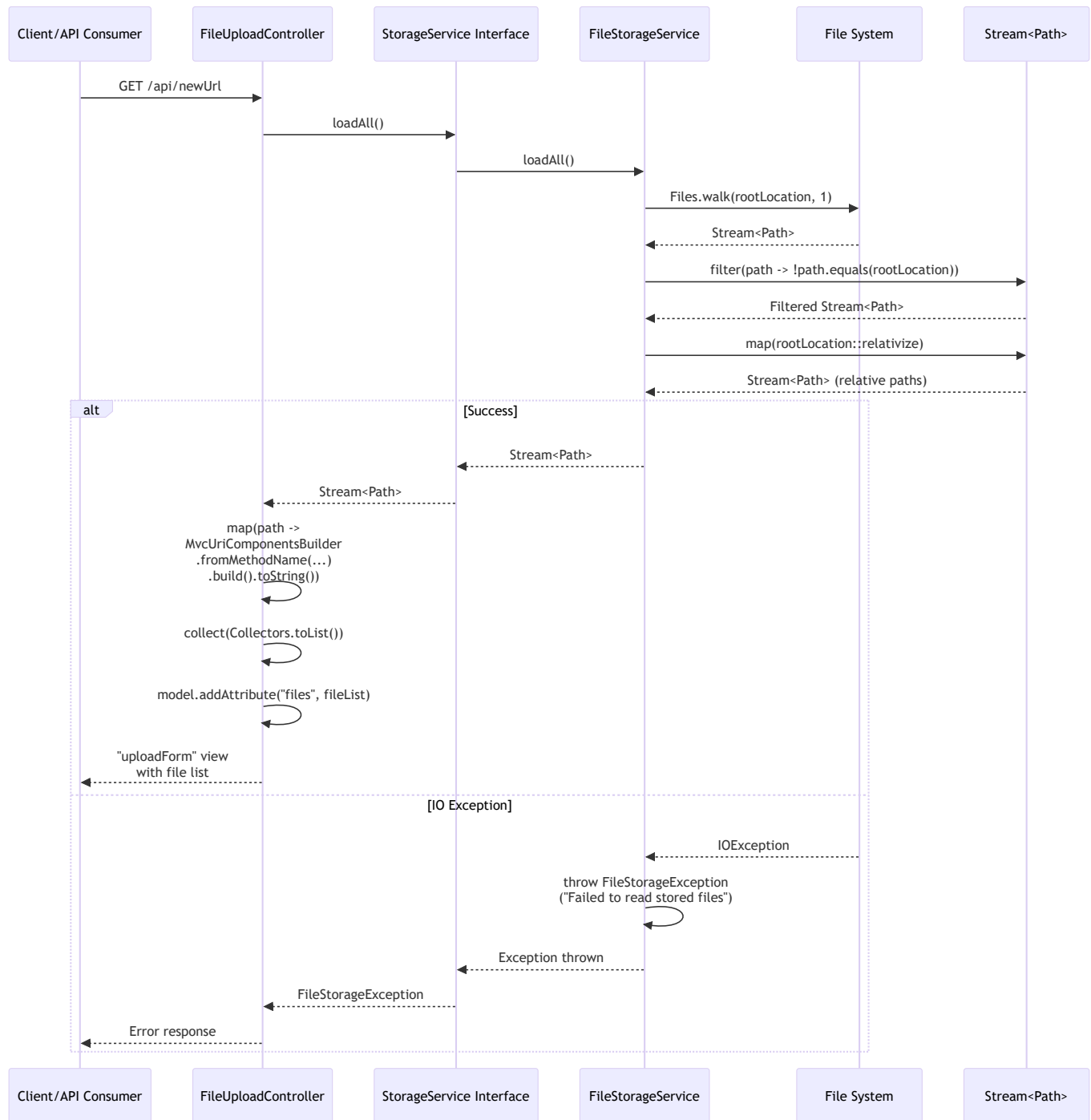
4. File Retrieval Flow

This diagram shows how files are retrieved and served to clients.



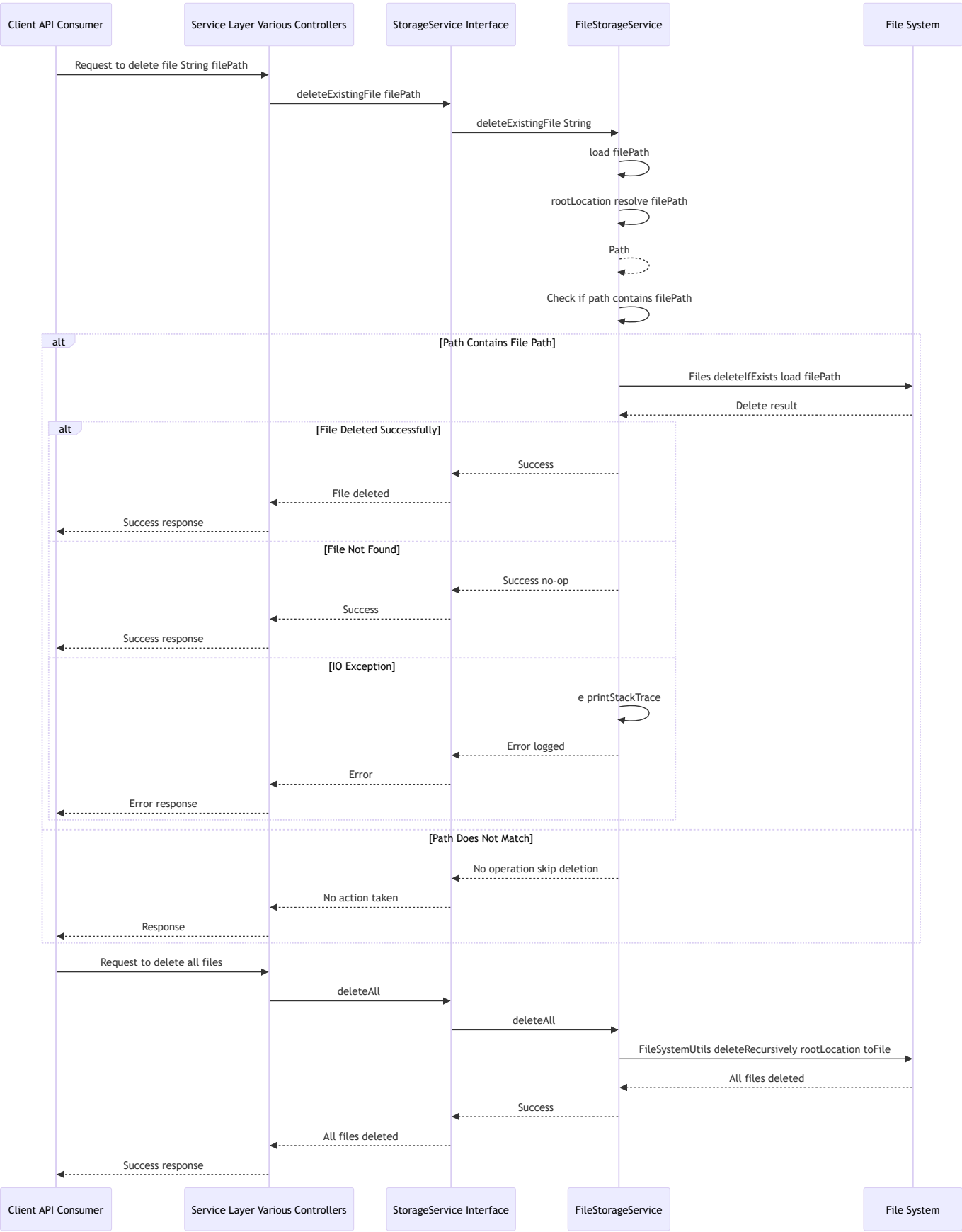
5. File Listing Flow

This diagram shows how all stored files are listed.



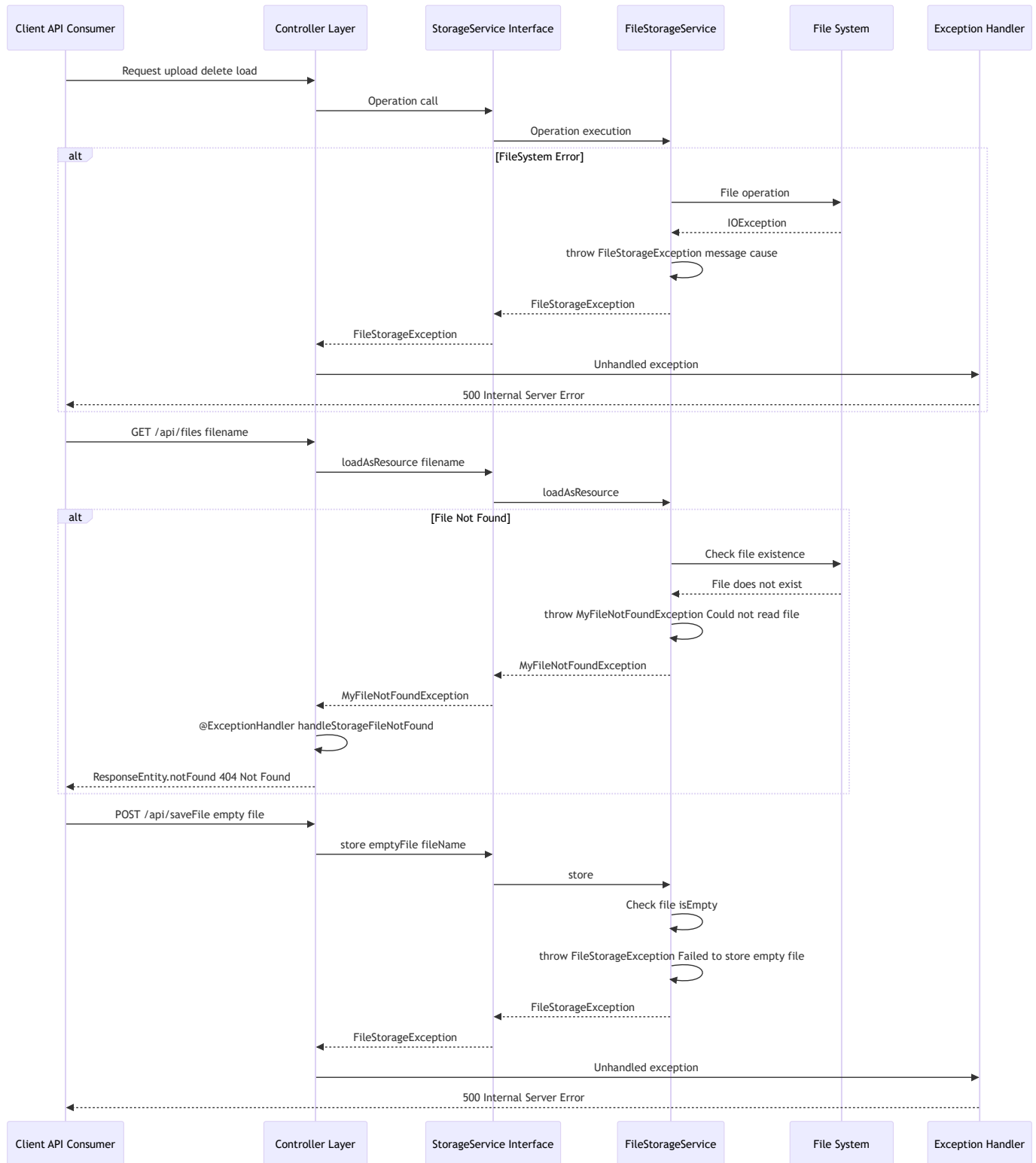
6. File Deletion Flow

This diagram shows how files are deleted from storage.



7. Error Handling Flow

This diagram shows how errors are handled throughout the storage module.



Module Components Overview

Classes and Interfaces

1. **StorageService** (Interface)

- Defines the contract for storage operations
- Methods: `init()` , `store()` , `loadAll()` , `load()` , `loadAsResource()` , `deleteAll()` , `deleteExistingFile()`

2. **FileStorageService** (Implementation)

- Implements `StorageService` interface
- Handles actual file system operations
- Uses `StorageProperties` for configuration

3. **StorageProperties**

- Configuration properties class
- Annotated with `@ConfigurationProperties("storage")`
- Default location: "upload-dir"

4. **FileStorageException**

- Runtime exception for storage-related errors
- Used for general storage failures

5. **MyFileNotFoundException**

- Specialized exception for file not found scenarios
- Annotated with `@ResponseStatus(HttpStatus.NOT_FOUND)`
- Returns 404 status code

Key Features

- **File Upload:** Supports both `MultipartFile` and Base64 image uploads
- **File Retrieval:** Load files as Spring Resource for serving
- **File Listing:** List all stored files with relative paths
- **File Deletion:** Delete single files or all files
- **Error Handling:** Comprehensive exception handling with appropriate HTTP status codes
- **Initialization:** Automatic directory creation on application startup

Integration Points

The Storage Module is used by various controllers across the application:

- `FileUploadController` - Direct file upload/download operations

- Various service implementations for module-specific file operations (Warranty, Service, Sales, CRM, etc.)