# Projects Upload API

## Current state

Currently API is not in a state to add dataload implementations to it. Authentication is hardcoded using Developer’s connection details, none of the tests are working, no logging, or dataload management in place, and there is lot of unused code that can be removed. It needs to incorporate the code review comments that were initially given.

1. Need for an authentication & authorization layer that seamlessly integrates with existing PS user-base, which will make it easier to scale in the future. This will involve designing the security interface in such a way that it can talk to both fourth salesforce SSO service and People System (TRGAuthentication database) interchangeably.

2. Currently we only have two integration tests and both of them are breaking. None of them are going to test the dataload process. Need to rewrite these integration tests that are breaking and add more tests to test upload activities which will be relevant to the development.

3. URL, context classes, entity classes should be renamed to correctly reflect the inherent DB objects that it represents, method naming to be more subtle and consistent e.g. avoid PSLivePortalDBContextFactory\_ASYNC

4. Strip off all the irrelevant HTTP methods e.g. get, delete, put and some that are not required for a dataload process. This will otherwise quickly generate tech debt. It may have been copied over by some sample API code and is not really required for this project .

5. Design the DB changes to accommodate dataload management that allows dataload drill down

6. Incorporate the code comments from the earlier revision 1 (bullets from CR rev 1)

* GetDefaultHolidayAllowanceData() is defined as private making it inaccessible from anywhere.
* Use of Nbuilder where necessary.
* Code cleanup is needed for the projects and libraries that are not in use. There are references to .js libraries and they are not in use.
* Create Singleton Configuration manager and replace all direct references.
* Data validation should be done separate from the controller methods
* Deletion (running without authorization), can be run number of times with the same IDs and always returns success. e.g. <http://localhost:54541/organisations/426/defaultholidayallowance/1> (delete) can be run multiple times with same outcome
* Code refactor to improve readability e.g. **use ternary operators** to reduce code clutter and every public method should be testable.
* Try to use **inversion of control** for Logger by passing it through constructor so that if we ever decide to change the logger in the future we will not need to change the code inside the class.
* Versioning of API, check if needed. Starting from V1 for future development.
* Project Fourth.PSLiveDataLoads.Repositories.Tests isn’t doing anything right now, this can be removed.

## Minimum Viable Product

### Dataload Management

Need the ability to list all the dataloads that were run on a portal, of a type. This can then help track the status of its run such as success/failure/ or minor issues along with the details of what went wrong along with their respective reason.

### Security

Projects upload api needs to also talk to Layer 7 for authorization for each request.

### Dataload

All dataloads will go by their respective names under GroupID and accept data in Jason/XML format. This data will primarily be fed into through a Fourth Connect API (using Informatica), which will convert excelsheet into raw json data.

## Priority Data loads

Following is the order in which dataloads implementation will be attempted.

