## Reviewer-One call Input API

**Description**  
This API is used for **PDF proofing** of all required users (reviewers) under the same manuscript. Each reviewer will have a **unique URL** along with a **token number** provided by the ePDF system through the *Reviewer One Status API*.

**Steps to call this API:**

1. Collect all required documents (.docx – manuscripts uploaded by reviewers). All files will share a **common ManuscriptId**, which comes from the JSON uploaded via the Reviewer One application.
2. Gather all files into a single folder named with the corresponding **ManuscriptId**.
3. Convert the folder (including the JSON) into a **single ZIP file** named **ManuscriptId.zip**.
4. Insert an entry into the database-Table with the following fields:
   * Id, ReferenceNo, ManuscriptId, UserId, Status, CreatedOn, ModifiedOn
   * **First-time entry values:**
     + *Status*: **Initiated**
     + *ReferenceNo*: **NULL**
     + *ModifiedOn*: **NULL**
5. Once the API responds with a **Reference Number**, update the database entry as follows:
   * *ReferenceNo*: (value from API response)
   * *Status*: **In Process**
   * *ModifiedBy*: **Trigger**

Below is the path for the files and zip file,

Folder structure

…/UserId/Input\ManuscriptId.zip\ManuscriptId\ManiuscriptId



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## ReviewerOne-Status Api called by UCC

Provide the author information in **JSON format** using the same **referenceNumber**. Set the **Status** field to AuthorJSON-Received-InProcess.

The JSON file should be named:

AuthorInfo\_<ManuscriptId>.json

and stored in the following directory structure:

<UserId>/Input/<ManuscriptId>/AuthorInfo\_<ManuscriptId>.json

A screenshot of a computer

AI-generated content may be incorrect.

The author information from the JSON will be **bound** to the metadata, as shown in the screenshot below.A screenshot of a computer

AI-generated content may be incorrect.

## ReviewerOne-Status Api called by ePdf

## Description (Revised)

## Provide the number of links that contain the same URL, each associated with a unique token for every reviewer, along with the reference number.

## The ReviewerOne system should update the main table with the new Status: Links-Received.

## Simultaneously, a separate table should receive a new entry with the following columns:

|  |  |
| --- | --- |
| Column | Description |
| Id | Primary key of the entry |
| UserId | Reviewer’s unique identifier |
| Links | URLs assigned to the reviewer |
| TokenNumber | Unique authentication token |
| ManuscriptId | Foreign Key referencing the main table |
| Status | Either Fresh or Read |
| ReferenceNumber | The reference number tied to the API flow |
| CreatedOn | Timestamp of when the link was assigned |
| AssignedBy | Indicates the assigner (system) |

## It is necessary to resolve the UserId based on the given ManuscriptId.

## When any reviewer logs in, they should see the link(s) associated with their UserId, as stored in this secondary table.

## If the publisher assigns a reviewer, they generate a new entry in that table using the same UserId.

## ReviewOne Stauts api called by ePdf

The API will return the referenceNumber and token for each link’s output. The system will then store the resulting output file in the designated folder structure.

…/UserId/Token/Output/ManuscriptId/OutputManuscriptId.Zip

The status in the secondary table will be set to **Completed** once all links associated with the same manuscript and reference number have been processed. At that point, the status in the primary table should also be updated to **Completed**.