# Test Plan

Positive tests have a blue ID, negative tests have a red ID.

Manual tests are underlined

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Importance | Description | Expected outcome | Actual Outcome | Status |
| 1 | API Status | High | Check that API URL is accessible online | API returns the requested page without error. | Page was returned | ***Success*** |
| 2 | Movie Data Structure | Medium | Inspect that movie data obeys the mandated structure | Movie data has no extra or missing fields. | Structure was fine | ***Success*** |
| 3 | Get All Movies | High | Check that API returns a list of all movies | API returns a list of movies. | Successfully populated list without error | ***Success*** |
| 4 | Get All Movies Individually | High | Check that each movie returned in test 3 is accessible individually | Each movie from the get all method is traversable. | Successfully downloaded every individual movie | ***Success*** |
| 5 | Get Non-existent ID Value | Low | Try to load a movie with a non-existent ID value | The API realizes that the ID is missing. |  | ***Pending*** |
| 6 | Get Invalid ID Data Type | Low | Try to load a movie with a string ID | The API catches this invalid case. |  | ***Pending*** |
| 7 | Update Movie | High | Check that a movie can be updated | Movies can be updated. | Status was OK | ***Success*** |
| 8 | Update Movie Details | High | Read a movie and store its data.  Create an updated version of this data, ensuring that it is different.  Update the movie with the new data.  Get the movie from the API and ensure that it has been updated accordingly. | Movie is updated properly. | Movie was properly updated | ***Success*** |
| 9 | Invalid Movie Update (Data Type) | High | Update movie using a string rating | Movie rating is not updated. |  | ***Pending*** |
| 10 | Invalid Movie Update (ID) | Medium | Try to update a non-existent ID | The API realizes that the ID is missing. *PTO*  Movie is not inserted or updated. |  | ***Pending*** |
| 11 | Update Movie ID | High | Try to change a movie’s ID using an existent ID | ID does not change. |  | ***Pending*** |
| 12 | Delete Movie | High | Delete a random movie.  Ensure it has been deleted. | Movie is deleted. | Movie was gone | ***Success*** |
| 13 | Delete Invalid Movie | Low | Delete a non-existent movie | API realizes that the movie doesn’t exist. |  | ***Pending*** |
| 14 | Add Movie | High | Add a new movie and check that it exists properly | Movie exists with correct data. | Movie was added properly | ***Success*** |
| 15 | Add Invalid Movie (ID) | Medium | Add a movie with a negative ID | Movie does not exist under given ID: ID should be autogenerated. |  | ***Pending*** |
| 16 | Add Invalid Movie (Data Structure 1) | Low | Add a movie with missing title | Movie does not exist: title is mandatory. |  | ***Pending*** |
| 17 | Add Invalid Movie (Data Structure 2) | Low | Add a movie with excess fields | API tackles excess fields. |  | ***Pending*** |
| 18 | Add Invalid Movie (Data Types) | Medium | Add a movie with a string ID and rating | API handles incorrect data types. |  | ***Pending*** |