

Lab 13: Complete APIs Integration (Add, Edit, and Delete)

You will continue to **update the client part** of the web application to **invoke all the APIs** designed and implemented in the previous labs. Remember to minimize the “*n-clients problem*” shown during the lecture, by reloading the application state (i.e., the list of films) from the server every time you modify it.

Note: it is not required to implement operations using the optimistic update approach.

1. Add a New Film

When the user adds a new film, it must be **saved on the server-side database**, and the displayed list of films must be updated accordingly. To do so, invoke the proper API endpoint for adding films, and then retrieve the updated list of films from the server. In this way you will be sure that client and server side are always aligned.

2. Edit a Film

Make the update operations persistent: when the user updates a film through its dedicated edit form, the **film is modified on the server-side database**, and the list of films is updated and displayed accordingly. To do so, invoke the proper API for updating a film, and then retrieve the updated list of films from the server.

3. Delete a Film

Make the delete persistent: when the user deletes a film, the **film is removed from the server-side database**, and the list of films is updated and displayed accordingly. To do so, invoke the proper API for deleting a film, and then retrieve the updated list of films from the server.

4. Optional: Edit in-line

- Users should have the possibility of changing some values “in-line”:
 - the *favorite* property through the checkbox displayed near each film in the list.
 - the rating property through the displayed stars. E.g., if the film is rated two out of five and the user clicks on the fourth star, the new rating becomes four out of five.

Note: if the user clicks on the current assigned rate, you should not trigger an update towards the server.

Hints:

1. **Trust the server!** It shall be always up-to-date, and the final state of every operation should rely on it, not on the internal state of the web application.
2. Make sure you have a back-up copy of the database file before testing your APIs, since it will be needed to restore the films after being deleted or edited. Note that you might need to restart the server to re-read the content of the overwritten database file.