

Lab 8: API Integration

In this lab, you will update your application to invoke the APIs designed and implemented in [Lab 03](#). Your goal is to ensure that, by the end of the lab, you have full synchronization between the client-side interface and the server-side data. Please, note that this lab will last **2 weeks**. In the first week, we suggest that you implement points 1 and 2 below.

1. Set up CORS

Starting from the API server realized in Lab 03, update it to ensure a suitable set up of CORS. Define which domains and HTTP methods you want to enable.

2. Item List Display

Add a new “API.mjs” in your React application to store the functions needed to call the API defined on the server. Then, modify your React application so that when it loads, the available items are gathered from the server by invoking the corresponding API. The retrieved data must be **stored in the application’s state**. Make sure to display all the items associated with each exam topic, since some topics may include more than one type of item. Additionally, some pages might need one list of items, while others might require a different list. Be sure to load and display the correct items based on the current page!

3. Add a New Item

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

4. Edit an Item

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

5. Delete an Item

Make the delete operations persistent. If your exam topic requires it, when the user deletes an item, **the item is removed from the server-side database**, and the list of items is updated and displayed accordingly. To do so, invoke the proper API for deleting an item, and then retrieve the updated list of items from the server.

Notes:

1. **Trust the server!** It shall be always up-to-date, and every operation relies on it, not on the internal state of the web application.
2. Create a back-up copy of the database before testing your APIs.