Technical decisions

For the authentication scheme, I chose the most basic HTTP authentication method of sending over the username and password on every request. However, constantly sending over the username and password over the network is not very secure, so if this were a real project I would've gone with something more advanced, such as token-based authentication.

For the database design, I chose to store each bond entry in the bond table along with the user id of the user who uploaded the bond. The downside of this is that if multiple users upload the same bond the same bond may appear multiple times in the table which is not storage efficient. However, the upside is that querying the bond table is extremely fast since no joins are required.

Alternatively I could have ensured that each bond (each ISIN) only appears once in the bond table, and then used a separate table to keep track of the ISINs that each user has uploaded. This would have the benefit of being more storage efficient since each bond is saved only once, but the downside is that querying is a lot slower because the two tables would have to be joined.

I also tested the API with the Postman API client. In addition to the typical tests one would do I also went through some edge cases such as attempting to access the API without user credentials, and testing for what happens if I try to add the same bond twice with the same user (I choose to return an error message in this case and not add anything to the DB). If I had more time and this were a real project I would have implemented automated unit tests which would allow the tests to be run regularly every time a new feature is implemented.