

| Which example step of num "CRUD" | Example on a website | On the client | On the server | |
|---|---|---|--|--------------------------------------|
| | | React | Express | call(s) to MongoDB |
| 1 Create | user submits new apartment for rent | Display form fields Handle button or submit event POST /apartments using fetch Get response If successful: Present created review else: Present error | app.post("/apartments", myRouteHandler) get details from submitted req.body validate try to insert into mongodb res.json(newlyCreatedApartment) //this includes an or return an error (json + status code) | insertOne(document, callback) |
| 2 Read | search for apartments to rent by location using a search box | Display search box Handle input change events and button click event GET /apartments?location=camden using fetch Get response, turn it into object with json() store results list in state variable Display results with apartments.map(a => <Apartment apartment={a}... /> | app.get("/apartments", myRouteHandler) get location param from req.query.location get all matches from mongoDB (or first 20) res.json(matchingApartments) | find(filter, callback) |
| 3 Read | click to read the details of one apartment for rent | handle button click or route GET /apartments/1337 using fetch Get response, turn string into object with json() Store result in a state variable Display the object (JSX) | app.get("/apartments/:id", myRouteHandler) req.params.id (a route param) res.json(foundApartment) | findOne(filter, callback) |
| | | | | |
| | | | | |

| | | | On the client | On the server | |
|---------------------------------------|--|---|---|--|---|
| Which exar step of num"CRUD" | | Example on a website | React | Express | call(s) to MongoDB |
| 4 Update | | submit a form which changes details of an apartment | Display existing record, with form fields for editing Handle submit button event Validate fields PUT /apartments/1337 using fetch Get response Show user the outcome, including any error | app.put("/apartments/:id", myRouteHandler) get document from req.body (use express.json()) perhaps lookup original record validate try to update in mongodb --> res.json(updatedApartment) or an error | findOneAndUpdate(filter, updateObj, callback) |
| | | | | | |
| 5 Delete | | Delete an apartment | DELETE to /apartments/1337 using fetch Get response Show user the outcome, including any error | app.delete("/apartments/:id", myRouteHandler) get id to delete from the route param req.params.id try to delete in mongodb res.sendStatus(200) or an error | deleteOne(filter, callback) |