A FULLSTACK REACT *MASTERCLASS*

TINYHOUSE

Quiz Questions | Answers

Module 7 | Building the Listing Page

1. Within a Listing GraphQL object in our TinyHouse server, we've used the JavaScript `JSON.stringify()` function to resolve the `bookingsIndex` field to a string before sending it to the server. Why?

A: The `bookingsIndex` field is a string on the server so we want to ensure it remains a string.
B: It's good practice to do so whenever possible.
C: The `bookingsIndex` object within a listing document is an unstructured data set where we won't know what the values are going to be so we aren't able to define a suitable GraphQL type. GraphQL doesn't have a default scalar type for JSON so we use the `JSON.stringify()` to send the object as a JSON string.
D: To ensure the `bookingsIndex` field of a Listing is to represent a GraphQL scalar type of JSON.

Answer: C - The `bookingsIndex` object within a listing document is an unstructured data set where we won't know what the values are going to be so we aren't able to define a suitable GraphQL type. GraphQL doesn't have a default scalar type for JSON so we use the `JSON.stringify()` to send the object as a JSON string.

2. What does the following `moment.js` function do?

```
moment(checkOutDate).isBefore(checkInDate, "days");
```

A: It returns a `boolean` value based on the condition that `checkOutDate` is before `checkInDate` by a period of days.
B: It returns a number value based on the number of days between the `checkInDate` and `checkOutDate` properties.
C: It decrements the number of days from `checkOutDate` by the difference between `checkInDate` and `checkOutDate`.
D: `moment.isBefore()` is not a valid moment function.

Answer: A - It returns a `boolean` value based on the condition that `checkOutDate` is before `checkInDate` by a period of days.

3. For TinyHouse, we've only allowed a user to view the bookings that have been made for a listing only when the user views one of their own listings.

A: True
B: False

Answer: A - True.

4. Which of the following best describes where the value of the `id` variable in the code below comes from?

```
import React from "react";
import { RouteComponentProps } from "react-router-dom";
import { useQuery } from "@apollo/react-hooks";
import { LISTING } from "../../lib/graphql/queries";
import {
  Listing as ListingData,
  ListingVariables
} from "../../lib/graphql/queries/Listing/__generated__/Listing";

interface MatchParams {
  id: string;
}

export const Component = ({ match }: RouteComponentProps<MatchParams>) => {
  // ...

  const { loading, data, error } = useQuery<ListingData, ListingVariables>(LISTING, {
    variables: {
      id: match.params.id,
      // ...
    }
  });
```

A: The `match.params.id` value refers to the `id` value created and kept as part of component state.
B: The `match.params.id` value refers to the `id` query parameter in the route (e.g. `/listings?id=2`).
C: The `match.params.id` value refers to the `id` URL parameter in the route (e.g. `/listings/:id`).
D: The `match.params.id` value refers to the `id` value within a `match` object that was created by and passed down from a parent component.

Answer: C - The `match.params.id` value refers to the `id` URL parameter in the route (e.g. `/listings/:id`).

5. The `<Link />` component from React Router is used to help display a pre-styled anchor tag in the UI.

Answer: B - False. The `<Link />` component from React Router helps provide links that allow for navigation around an application.