Client - Server interactions

Homepage Search query:
POST to server, send JSON object:

- Location (City specific): LatLng
- Arrival Date: dd:mm:yyyy formatted string
- Departure Date: dd:mm:yyyy formatted string

Reponse:

- Load search results page with map centered on search location (client-side)
- Send back JSON Rooms object from server. Client populates search results and map markers from this

Search results search query:
POST to server using AJAX, send JSON object:

- Location (City specific): LatLng
- Arrival Date: dd:mm:yyyy formatted string
- Departure Date: dd:mm:yyyy formatted string
- “More filters” - All options available for refining search

Response:

- Update map to new center if applicable (client-side)
- Send back JSON Rooms object from server. Rooms object will only contain rooms that fit the “more filters” - this processing is done on the server. Client populates search results and map markers.

Load room details (clicking a search result):
POST to server with AJAX, send JSON object:

- Room object clicked on

Response:

- Hotel object. Client then populates divs with information from the Hotel and Room objects, including images

Confirm booking:
POST to server with AJAX, send JSON object:

- User object (for keeping track of who booked the room)
- Arrival/Departure dates
- Payment info as object, if not already stored in user object via adding to the users profile.

Response:

- Success code, something to tell the client the booking was successful. Client updates page accordingly.

Go to “my profile” page:
GET to server with AJAX:

- GET user profile object

Response:

- Send user profile object, client populates new divs with this information

Update user personal details:
POST to server with AJAX, send JSON object:

- Updated details from forms within user profile, for updating the user profile object on the server

Response:

- Success code, something to tell the client the changes were made successfully. Client updates page accordingly.

‘Favourite’ a room:
POST to server with AJAX, send JSON object:

- Room object favourited, to be added to the User profile object.

Response:

- Success code

Log in/sign up:
POST to server with AJAX, send JSON object:

- Information from forms or use APIs for logging in (unsure of these details yet)

Response:

- User profile object that matches the login, or a failure code/message if no matches. Create user session

Logout:
POST to server with AJAX, send JSON object:

- Some string or text indicating the user wants to log out

Response:

- End user session

Data to be stored on server

Hotel object:

- Hotel name: (String)
- Hotel ID ? : (int)
- Location: LatLng
- Address: (string)
- Facilities: (Facilities object)

Facilities object:

- General information: (string - for room Details page)
- Internet access: (bool)
- Room service: (bool)
- Laundry services: (bool)
- Pool: (bool)
- Gym: (bool)
- Breakfast: (bool)

Room object:

- Number of beds: (int)
- Average star rating: (int, 1-10 for out-of-5 with half marks)
- Cost per night: (int/float)
- Reviews: Array (Review objects)
- Images: Array of images, [0] is hero image, rest are for gallery
- Details: (string - for room Details page - can be expanded to its own object)
- isBooked: (bool)
- bookedUser: (User object)
- bookedDates: (2 strings, dd:mm:yyyy)

User profile object:

- Booked hotels: array (hotel objects)
- Favourite hotels: array (hotel objects)
- Name: (string)
- Age: (int)
- Password: (encrypted string)
- Email: (string)
- Credit Card details: (ALL DUMMY PLACE HOLDERS/NO FORM)

Manager profile object:

- Listed properties: Array (Hotel objects)
- Listed rooms: Array (Room objects)
- Current bookings: Array (Room objects)
- Password: (Encrypted string)
- Email: (string)
- Credit Card details: (ALL DUMMY PLACE HOLDERS/NO FORM)

Review object:

- Reviewer name: (string)
- Star rating given: (int, 1-10)
- Review text: (string)