POST, GET, PUT, DELETE - Detailed Code Explanation

1. POST API - Create Booking

Purpose: The POST API is responsible for creating a new booking in the system. It builds a dynamic payload, sends a POST request, stores the booking ID, and returns the response body.

```
async postCreateBooking(overrides = {}, statuscode) {

    const dynamicPayload = await this.api.buildPayload(
        payloads.postbooking.booking,
        overrides
    );

    const response = await this.api.request({
        method: "POST",
        url: endpoints.url,
        endpoint: endpoints.getbookingid,
        body: dynamicPayload,
        expectedStatus: statuscode
    });

    this.world.bookingId = response.body.bookingid;

    return response.body;
}
```

2. GET API - Retrieve Booking

Purpose: The GET API retrieves booking information. It can fetch either all booking IDs or a specific booking based on bookingId stored in the World object.

```
async getBookingDetails(enterBookingID = null, statuscode = 200) {

    const bookingId = enterBookingID || this.world?.bookingId;

    const response = await this.api.request({
        method: "GET",
        url: endpoints.url,
        endpoint: endpoints.getbookingdetils + bookingId,
        expectedStatus: statuscode
    });

    return response.body;
}
```

3. PUT API - Update Booking

Purpose: The PUT API updates an existing booking. It first generates an authentication token, builds a dynamic payload, and sends a PUT request.

```
async updateBookingDetailsByID(overrides = {}, statuscode = 200) {

    const bookingId = this.world?.bookingId;

    const token = await this.api.tokengenerator();

    const dynamicPayload = await this.api.buildPayload(
        payloads.putbooking.updatebooking,
        overrides
    );

    const response = await this.api.request({
        method: "PUT",
        url: endpoints.url,
        endpoint: endpoints.getbookingdetils + bookingId,
        body: dynamicPayload,
        token: token,
        expectedStatus: statuscode
    });

    return response.body;
}
```

4. DELETE API - Remove Booking

Purpose: The DELETE API removes a booking from the system. It generates an authentication token, sends DELETE request, and clears bookingId from World object after successful deletion.

```
async deleteBookingDetailsByID(statuscode = 201) {

    const bookingId = this.world?.bookingId;

    const token = await this.api.tokengenerator();

    const response = await this.api.request({
        method: "DELETE",
        url: endpoints.url,
        endpoint: endpoints.getbookingdetils + bookingId,
        token: token,
        expectedStatus: statuscode
    });

    this.world.bookingId = null;

    return response.body;
}
```

Complete Flow Summary

POST → Creates resource and stores ID GET → Retrieves resource using stored ID PUT → Updates resource using token authentication DELETE → Deletes resource and clears state All APIs use the centralized ApiMethods request engine for sending HTTP requests and performing validations.