Doodle QA Backend Challenge

1. User API (UserController)

Create User

- Create user with valid details
- Create user with missing required field (name/email).
- Create user with duplicate email
- Create user with maximum allowed calendars.
- Create user exceeding calendar limit CalendarLimitExceededException.
- Create user with Empty calendar list
- Create user with invalid email format.
- Create user with UNICODE characters.

Get User by ID

- Retrieve existing user by valid ID.
- Retrieve user with non-existent ID.
- Retrieve user with invalid ID format.

List Users

- List users – custom pagination & sorting (by createdAt, by lastName).
- List users – default pagination (page=0,size=20).

Update User

- Update existing user with valid details.
- Update user with non-existent ID.
- Update user with outdated version.
- Update user to have duplicate email.

Delete User

- Delete existing user by valid ID
- Delete non-existent user
- Delete user with active calendars (if business rule restricts).

User–Calendar association rules

- Create user calendar when user exists
- Add calendar – duplicate calendar ID
- Create user calendar for non-existent user
- Add calendar – over limit
- Remove calendar from user
- Remove calendar – calendar not present on user
- Add/remove multiple calendars atomically – all succeed or none (transactionality).

2. Meeting API (MeetingController)

Create Meeting

- Create meeting with valid details
- Create meeting with missing required fields (title, start/end time)
- Create meeting in non-existent calendar
- Create overlapping meeting in same calendar
- Create meeting with invalid time range (end < start)

Get Meeting by ID

- Retrieve existing meeting by valid ID
- Retrieve meeting with non-existent ID
- Retrieve meeting with invalid ID format

Update Meeting

- Update existing meeting with valid details.
- Update meeting with non-existent ID.
- Update meeting with invalid time range .
- Update meeting to overlap with another meeting.
- Update meeting after creating in different TimeZones.

Delete Meeting

- Delete existing meeting by valid ID.
- Delete non-existent meeting.

List Meetings

- Retrieve all meetings for a calendar
- Retrieve meetings with pagination parameters.
- Retrieve meetings for non-existent calendar.

Authentication & Authorization

(If the service is secured — not present in current code, but important to verify if implemented later)

- Access API without authentication token.
- Access API with expired token.
- Access API with valid token but insufficient permissions
- Access API with valid token and correct permissions.

Input Validation & Injection Prevention

- SQL Injection attempt in `name` field
- SQL Injection attempt in `email` field.
- XSS attempt in `name` field.
- Path parameter injection (`/users/{id}` with `../../../etc/passwd`)
- Oversized payload in request body.

- **Invalid JSON structure in request.**

Data Integrity & Privacy

- **Verify sensitive data (email, IDs) not logged in plaintext.**
- **Verify error responses do not expose internal stack traces in production mode**
- **Verify deleted user data cannot be retrieved through API or Kafka messages.**
- **Attempt to update another user’s data without permission.**