บริษัทบ้านบ้านจำกัดมหาชน จ้างให้ผู้เรียนสร้าง Spring Boot Application สำหรับการจัดการระบบพนักงานโดยที่ระบบต้องสามารถเก็บข้อมูลได้ดังนี้

1. ชื่อจริง (first_name)
2. นามสกุล (last_name)
3. ชื่อเล่น (nickname)
4. เงินเดือน (salary)
5. สถานะ (status) - current, deleted
6. ที่อยู่ (address)
7. ตำแหน่ง (position)

โดยให้รหัสพนักงานคือ Primary key

และมี Endpoint ตามนี้

- (3 คะแนน) GET /api/employees สำหรับดูพนักงานทั้งหมดในระบบ
    - โดยต้องคืนข้อมูลดังนี้
        - รหัสพนักงาน (id) ⟵ Primary key
        - ชื่อจริง (first_name)
        - นามสกุล (last_name)
        - ชื่อเล่น (nickname)
        - เงินเดือน (salary)
        - สถานะ (status)
        - ที่อยู่ (address)
        - ตำแหน่ง (position)
    - และสถานะที่คืนออกมาเป็น 200 OK
- (3 คะแนน) GET /api/employees/{id} สำหรับดูพนักงาน 1 คน
    - ถ้าหาเจอให้คืนสถานะออกมาเป็น 200 OK
    - ถ้าหาไม่เจอให้คืนสถานะออกมาเป็น 404 Not found พร้อมข้อความว่า “Not found employee ID: {id}” โดย {id} คือรหัสพนักงาน
- (5 คะแนน) POST /api/employees สำหรับเพิ่มพนักงาน 1 คนในระบบ
    - โดยต้องระบุข้อมูลดังนี้
        - ชื่อจริง (first_name)
        - นามสกุล (last_name)
        - ชื่อเล่น (nickname)
        - เงินเดือน (salary)
        - ที่อยู่ (address)
        - ตำแหน่ง (position)
    - สำหรับพนักงานที่ถูกเพิ่มเข้ามาใหม่ ให้กำหนด สถานะ (status) เป็น current
    - เมื่อเพิ่มพนักงานสำเสร็จให้คืนข้อมูลพนักงานมาพร้อมรหัสพนักงาน (Primary key)
    - โดยคืนสถานะเป็น 200 OK
- (3 คะแนน) DELETE /api/employees/{id} สำหรับลบพนักงาน 1 คน (แบบ Soft delete)
    - โดย id คือรหัสพนักงาน
    - เมื่อมีการเรียก Endpoint นี้พนักงานจะไม่ถูกลบจริง ๆ แต่ถูกแก้ไขสถานะเป็น deleted
    - เมื่อ Delete สำเสร็จให้คืนค่าออกมาเป็นสถานะ 204 (No Content)

    - ถ้าพนักงานที่ลบไม่มีอยู่จริงให้คืนค่าออกมาเป็นสถานะ 404 (Not found) พร้อมข้อความว่า “Not found employee ID: {id}” โดย {id} คือรหัสพนักงาน
- (8 คะแนน) PUT /api/employees สำหรับแก้ไขข้อมูลของพนักงาน 1 คนในระบบ
    - โดยสามารถแก้ไขข้อมูลได้แค่ 4 อย่างดังนี้
        - ชื่อจริง (first_name)
        - นามสกุล (last_name)
        - ชื่อเล่น (nickname)
        - ที่อยู่ (address)
    - เมื่อแก้ไขสำเร็จให้คืนค่าออกมาเป็นสถานะ 200 OK และส่งค่าออกมาดังนี้
        - รหัสพนักงาน (id) ⟵ Primary key
        - ชื่อจริง (first_name)
        - นามสกุล (last_name)
        - ชื่อเล่น (nickname)
        - เงินเดือน (salary)
        - ที่อยู่ (address)
        - ตำแหน่ง (position)
    - หลังจากการแก้ไข ตัวข้อมูลอื่น ๆ ที่แก้ไขไม่ได้ เช่น เงินเดือน ตำแหน่ง สถานะ ยังคงต้องมีค่าเป็นค่าเดิม (ค่าเหมือนก่อนการแก้ไข)
    -
- (4 คะแนน) PUT /api/employees/salary/{id} สำหรับแก้ไขเงินเดือนของพนักงาน 1 คนในระบบ
    - โดย id คือรหัสพนักงานและต้องระบุข้อมูลดังนี้
        - เปอร์เซ็นต์ของการแก้ไขเงินเดือน (0% - 100% เท่านั้น)
        - ตัวอย่าง ถ้านาย A มีเงินเดือน 40,000 แล้ว Admin มีการใช้ Endpoint นี้ในการแก้ไขเงินเดือน โดยระบุเปอร์เซ็นต์เป็น 20% เงินเดือนของนาย A จะกลายเป็น 48,000
- (4 คะแนน) PUT /api/employees/position/{id} สำหรับแก้ไขเงินเดือนของพนักงาน 1 คนในระบบ
    - โดย id คือรหัสพนักงานและต้องระบุข้อมูลดังนี้
        - เก่าที่ต้องการแก้ไข
        - ตำแหน่งใหม่
    - ถ้าใส่ตำแหน่งเก่าไม่ถูกต้องจะไม่ให้แก้ไข และคืนค่าออกมาเป็นสถานะ BAD_REQUEST 400 และข้อความว่า “Current position is incorrect”
- (4 คะแนน) GET /api/employees/name สำหรับดูพนักงานทั้งหมดที่มีชื่อตรงกับแนบมา
    - รับ query parameter ที่ชื่อว่า q โดยใช้ชื่อนี้ในการค้นหา
    - ตัวอย่าง GET /api/employees/name?q=om จะคืนพนักงานที่มี “om” อยู่ในชื่อ เช่น Somchai, Somying, Aommie เป็นต้น
    - โดยต้องคืนข้อมูลดังนี้
        - รหัสพนักงาน (id) ⟵ Primary key
        - ชื่อจริง (first_name)
        - นามสกุล (last_name)
        - ชื่อเล่น (nickname)
        - เงินเดือน (salary)
        - สถานะ (status)
        - ที่อยู่ (address)
        - ตำแหน่ง (position)
    - และสถานะที่คืนออกมาเป็น 200 OK

- ถ้าไม่เจอเลยก็ให้คืนค่าเป็นอาเรย์ว่าง

- (6 คะแนน) DELETE /api/employees สำหรับพนักงานหลายคน
  - ให้อิสระในการเลือกว่าจะรับ List ของรหัสพนักงานผ่านทางไหน
  - เช่น Query parameter, POJOs
  - ตัวอย่าง (เป็นแบบ query parameters) เมื่อมีการเรียก DELETE /api/employees?ids=1,2,3,5,7
  - ให้ลบ (Soft delete) ของพนักงานทั้ง 5 คน (ID: 1,2,3,5,7)
  - เมื่อลบสำเร็จให้คืนค่าออกมาเป็น 204 No content
  - แต่ในกรณีที่ในบางรหัสพนักงานของรหัสพนักงานที่ส่งมาไม่มีในระบบ เช่น รหัสพนักงาน 3 และ 5 ไม่มีในระบบ ให้คืนค่าออกมาเป็นสถานะ 207 Multi status และให้ส่งรหัสพนักงานที่ไม่มีอยู่ในระบบออกมาด้วย

    HTTP status code: 207

    ```
    {
        not_found_ids: [3, 5]
    }
    ```