

## BACKEND ENDPOINT GUIDE

Everything returns a json object, aside from the particular data you want it also returns with a integer field "status"

0: success, go check the rest of the fields for your data

-1: failure, you didn't put in valid data or put in improperly formatted data

-2: failure, you didn't provide all the needed fields

User/Class data field types:

- email: string
- id: integer
- role: boolean
  
- /users/
  - Check for "email" field in json input
    - GET
      - Check for "data" field in json input
        - Check if "data" field is:
          - "profile"
            - Returns {"email":string, "name":string, "schedule":int[5][12]}
          - "schedule"
            - Returns {"schedule":int[5][12]}
          - "name"
            - Returns {"name":string}
    - POST
      - Check for "password" field in json input
        - Nothing special, "status" is 0 on success, -1 on failure
    - PATCH
      - Check for "schedule" field in json input
        - If true, attempts to update the user's schedule, takes int [5][12] array as input
          - Adds field "schedule" to the return object, 0 if successful, -1 if not
          - Sets "status" to -1
      - Check for "name" field in json input
        - If true, attempts to update the user's name
          - Adds field "name" to the return object, 0 if successful, -1 if not
          - Sets "status" to -1
    - DELETE
      - Nothing special, "status" is 0 on success, -1 on failure
  - /classes/
    - Check for "id" field in json input

- GET
  - Check for “data” field in json input
    - Check if “data” field is:
      - “all”
        - Returns {“classid”:int, “name”:string, “schedule”:int[5][12], “officehours”:int[5][12]}
      - “name”
        - Returns {“name”:string}
      - “schedule”
        - Returns {“schedule”:int[5][12]}
      - “officehours”
        - Returns {“officehours”:int[5][12]}
- POST
  - Check for “name” field in json input
    - Nothing special, “status” is 0 on success, -1 on failure
- PATCH
  - Check for “schedule” field in json input
    - Attempts to update the field, takes int [5][12] array as input, 0 on success, -1 on failure
  - Check for “name” field in json input
    - Attempts to update the field, 0 on success, -1 on failure
  - Check for “officehours” field in json input
    - Recalculates office hours
    - Adds field “officehours” to return, contains updated set of office hours(int[5][12]) on success, -1 on failure
- DELETE
  - Nothing special, “status” is 0 on success, -1 on failure
- /users/classes/
  - Check for “email” field in json input
    - GET
      - Returns {“classes”:[][int, boolean]}
    - POST
      - Check for “id” and “role” fields in json input
        - Regular status return, 0 on success, -1 on failure
    - PATCH
      - Check for “id” and “role” fields in json input
        - Regular status return, 0 on success, -1 on failure
    - DELETE
      - Check for “id” field in json input
        - Regular status return, 0 on success, -1 on failure
  - /classes/students/
    - Check for “id” field in json input
      - GET
        - Returns {“members”:[][string, boolean]}