Stage 1;

```
Last login: Sun Aug 31 14:33:37 on ttys006
neha@Neharikas-MacBook-Air task-backend % BASE="http://localhost:8080"
CT="Content-Type: application/json"
neha@Neharikas-MacBook-Air task-backend % TOKEN=$(
  curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
    -d '{"email":"alice@example.com","password":"secret123","name":"Alice"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" \
    -d '{"email":"alice@example.com","password":"secret123"}' | jq -r .token
)
AUTH="Authorization: Bearer $TOKEN"
echo "Token length: ${#TOKEN}"
curl -i "$BASE/api/auth/me" | sed -n '1,8p'
curl -s "$BASE/api/auth/me" -H "$AUTH" | jq .
Token length: 172
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
HTTP/1.1 401
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
{
  "userId": 1,
  "email": "alice@example.com",
  "name": "Alice",
  "role": "USER",
  "token": null
}
neha@Neharikas-MacBook-Air task-backend % BOARD_ID=$(curl -s "$BASE/api/boards" -H "$AUTH" -H "$CT" -d
'{"name":"Demo Board"}' | jq -r .id)
echo "BOARD_ID=$BOARD_ID"
curl -s "$BASE/api/boards" -H "$AUTH" | jq .
curl -s "$BASE/api/boards/$BOARD_ID" -H "$AUTH" | jq .
curl -s -X PUT "$BASE/api/boards/$BOARD_ID" -H "$AUTH" -H "$CT" \
  -d '{"name":"Demo Board (renamed)"}' | jq .
BOARD_ID=1
[
  {
    "id": 1,
    "name": "Demo Board",
    "lists": []
  }
]
{
  "id": 1,
  "name": "Demo Board",
  "ownerId": 1,
  "lists": []
}
{
  "id": 1,
  "name": "Demo Board (renamed)",
  "lists": []
}
neha@Neharikas-MacBook-Air task-backend % TODO_ID=$(curl -s "$BASE/api/boards/$BOARD_ID/lists" -H "$AUTH"
-H "$CT" \
  -d '{"name":"To Do","position":1}' | jq -r .id)
DOING_ID=$(curl -s "$BASE/api/boards/$BOARD_ID/lists" -H "$AUTH" -H "$CT" \
  -d '{"name":"In Progress","position":2}' | jq -r .id)
echo "TODO_ID=$TODO_ID  DOING_ID=$DOING_ID"
curl -s -X PUT "$BASE/api/boards/$BOARD_ID/lists/$DOING_ID" -H "$AUTH" -H "$CT" \
  -d '{"name":"Doing","position":2}' | jq .
curl -s "$BASE/api/boards/$BOARD_ID" -H "$AUTH" | jq .
TODO_ID=1  DOING_ID=2
{
  "id": 2,
  "name": "Doing",
  "position": 2
}
{
  "id": 1,
  "name": "Demo Board (renamed)",
  "ownerId": 1,
  "lists": [
    {
      "id": 1,
      "name": "To Do",
      "position": 1,
      "tasks": []
    },
    {
      "id": 2,
      "name": "Doing",
      "position": 2,
      "tasks": []
    }
  ]
}
neha@Neharikas-MacBook-Air task-backend % TASK_ID=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH" -H
"$CT" \
  -d '{"title":"Write tests","description":"Stage 1 check","dueDate":"2030-01-01"}' | jq -r .id)
echo "TASK_ID=$TASK_ID"
curl -s -X PUT "$BASE/api/lists/$TODO_ID/tasks/$TASK_ID" -H "$AUTH" -H "$CT" \
  -d '{"title":"Write tests (upd)","description":"updated","dueDate":"2030-01-02","status":"IN_PROGRESS"}' | jq .
curl -s -X PATCH "$BASE/api/tasks/$TASK_ID" -H "$AUTH" -H "$CT" \
  -d '{"status":"DONE"}' | jq .
curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH" | jq .
ALICE_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH" | jq -r .userId)
curl -s -X PUT "$BASE/api/tasks/$TASK_ID/assignees" -H "$AUTH" -H "$CT" \
  -d "{\"userIds\":[$ALICE_ID]}" | jq .
curl -i -s -X DELETE "$BASE/api/lists/$TODO_ID/tasks/$TASK_ID" -H "$AUTH" | sed -n '1,4p'
TASK_ID=1
{
  "id": 1,
  "title": "Write tests (upd)",
  "description": "updated",
  "dueDate": "2030-01-02",
  "status": "IN_PROGRESS",
  "createdAt": "2025-08-31T18:35:20.753076Z",
  "completedAt": null
}
{
  "id": 1,
  "title": "Write tests (upd)",
  "description": "updated",
  "dueDate": "2030-01-02",
  "status": "DONE",
  "createdAt": "2025-08-31T18:35:20.753076Z",
  "completedAt": "2025-08-31T18:35:20.808149Z"
}
[
  {
    "id": 1,
    "title": "Write tests (upd)",
    "description": "updated",
    "dueDate": "2030-01-02",
    "status": "DONE",
    "createdAt": "2025-08-31T18:35:20.753076Z",
    "completedAt": "2025-08-31T18:35:20.808149Z"
  }
]
{
  "id": 1,
  "title": "Write tests (upd)",
  "description": "updated",
  "dueDate": "2030-01-02",
  "status": "DONE",
  "createdAt": "2025-08-31T18:35:20.753076Z",
  "completedAt": "2025-08-31T18:35:20.808149Z"
}
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
neha@Neharikas-MacBook-Air task-backend % TOKEN_B=$(
  curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
    -d '{"email":"bob@example.com","password":"secret123","name":"Bob"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" \
    -d '{"email":"bob@example.com","password":"secret123"}' | jq -r .token
)
AUTH_B="Authorization: Bearer $TOKEN_B"
BOB_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_B" | jq -r .userId)
curl -i -s "$BASE/api/boards/$BOARD_ID" -H "$AUTH_B" | sed -n '1,8p'
HTTP/1.1 403
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
neha@Neharikas-MacBook-Air task-backend %
```

Stage 2;

```
Last login: Sun Aug 31 14:34:26 on ttys001
neha@Neharikas-MacBook-Air task-backend % BASE="http://localhost:8080"
CT="Content-Type: application/json"
neha@Neharikas-MacBook-Air task-backend % TOKEN_A=$(
  curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
    -d '{"email":"alice@example.com","password":"secret123","name":"Alice"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" \
    -d '{"email":"alice@example.com","password":"secret123"}' | jq -r .token
)
AUTH_A="Authorization: Bearer $TOKEN_A"
ALICE_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_A" | jq -r .userId)
TOKEN_B=$(
  curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
    -d '{"email":"bob@example.com","password":"secret123","name":"Bob"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" \
    -d '{"email":"bob@example.com","password":"secret123"}' | jq -r .token
)
AUTH_B="Authorization: Bearer $TOKEN_B"
BOB_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_B" | jq -r .userId)
TOKEN_C=$(
  curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
    -d '{"email":"carol@example.com","password":"secret123","name":"Carol"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" \
    -d '{"email":"carol@example.com","password":"secret123"}' | jq -r .token
)
AUTH_C="Authorization: Bearer $TOKEN_C"
CAROL_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_C" | jq -r .userId)
echo "ALICE=$ALICE_ID  BOB=$BOB_ID  CAROL=$CAROL_ID"
ALICE=1  BOB=2  CAROL=3
neha@Neharikas-MacBook-Air task-backend % BOARD_ID=$(curl -s "$BASE/api/boards" -H "$AUTH_A" -H "$CT" -d
'{"name":"Stage2 Board"}' | jq -r .id)
echo "BOARD_ID=$BOARD_ID"
BOARD_ID=1
neha@Neharikas-MacBook-Air task-backend % curl -i -s "$BASE/api/boards/$BOARD_ID" -H "$AUTH_B" | sed -n
'1,8p'
HTTP/1.1 403
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
neha@Neharikas-MacBook-Air task-backend % curl -s -X POST
"$BASE/api/boards/$BOARD_ID/members?userId=$BOB_ID&role=MEMBER" -H "$AUTH_A" | jq .
curl -s -X POST "$BASE/api/boards/$BOARD_ID/members?userId=$CAROL_ID&role=VIEWER" -H "$AUTH_A" | jq .
curl -s "$BASE/api/boards/$BOARD_ID/members" -H "$AUTH_A" | jq .
{
  "id": 2,
  "role": "MEMBER"
}
{
  "id": 3,
  "role": "VIEWER"
}
[
  {
    "id": 1,
    "role": "OWNER"
  },
  {
    "id": 2,
    "role": "MEMBER"
  },
  {
    "id": 3,
    "role": "VIEWER"
  }
]
neha@Neharikas-MacBook-Air task-backend % curl -i -s -X POST "$BASE/api/boards/$BOARD_ID/lists" -H
"$AUTH_C" -H "$CT" \
  -d '{"name":"Viewer List","position":1}' | sed -n '1,10p'
HTTP/1.1 403
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
```

```
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
neha@Neharikas-MacBook-Air task-backend % TODO_ID=$(curl -s -X POST "$BASE/api/boards/$BOARD_ID/lists" -H
"$AUTH_B" -H "$CT" \
  -d '{"name":"To Do","position":1}' | jq -r .id)
DOING_ID=$(curl -s -X POST "$BASE/api/boards/$BOARD_ID/lists" -H "$AUTH_B" -H "$CT" \
  -d '{"name":"In Progress","position":2}' | jq -r .id)
echo "TODO=$TODO_ID DOING=$DOING_ID"
TODO=1 DOING=2
neha@Neharikas-MacBook-Air task-backend % curl -i -s -X PUT "$BASE/api/boards/$BOARD_ID" -H "$AUTH_B" -H
"$CT" \
  -d '{"name":"Hacked by Bob"}' | sed -n '1,10p'
HTTP/1.1 500
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
neha@Neharikas-MacBook-Air task-backend % curl -s -X PUT "$BASE/api/boards/$BOARD_ID" -H "$AUTH_A" -H
"$CT" \
  -d '{"name":"Stage2 Board (official)"}' | jq .
{
  "id": 1,
  "name": "Stage2 Board (official)",
  "lists": []
}
neha@Neharikas-MacBook-Air task-backend % curl -i -s -X DELETE
"$BASE/api/boards/$BOARD_ID/members/$CAROL_ID" -H "$AUTH_A" | sed -n '1,6p'
curl -i -s "$BASE/api/boards/$BOARD_ID" -H "$AUTH_C" | sed -n '1,8p'
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
HTTP/1.1 403
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
neha@Neharikas-MacBook-Air task-backend % T1=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_B" -H
"$CT" \
  -d '{"title":"Spec review","description":"read docs","dueDate":"2030-01-01"}' | jq -r .id)
T2=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_B" -H "$CT" \
  -d '{"title":"Write tests","description":"stage2 tests","dueDate":"2030-01-02"}' | jq -r .id)
curl -s -X PATCH "$BASE/api/tasks/$T2" -H "$AUTH_B" -H "$CT" -d '{"status":"DONE"}' | jq .
curl -i -s -X DELETE "$BASE/api/lists/$TODO_ID/tasks/$T1" -H "$AUTH_A" | sed -n '1,6p'
curl -s "$BASE/api/boards/$BOARD_ID/activity?page=0&size=20" -H "$AUTH_A" | jq .
{
  "id": 2,
  "title": "Write tests",
  "description": "stage2 tests",
  "dueDate": "2030-01-02",
  "status": "DONE",
  "createdAt": "2025-08-31T18:37:55.177575Z",
  "completedAt": "2025-08-31T18:37:55.203774Z"
}
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
{
  "totalPages": 1,
  "totalElements": 8,
  "first": true,
  "last": true,
  "size": 20,
  "content": [
    {
      "id": 8,
      "type": "TASK_DELETED",
      "detail": "Task deleted",
      "createdAt": "2025-08-31T18:37:55.225716Z"
    },
    {
      "id": 7,
      "type": "TASK_UPDATED",
      "detail": "Task \"Write tests\" updated",
      "createdAt": "2025-08-31T18:37:55.204965Z"
    },
    {
      "id": 6,
      "type": "TASK_CREATED",
      "detail": "Task \"Write tests\" created",
      "createdAt": "2025-08-31T18:37:55.179134Z"
    },
    {
      "id": 5,
      "type": "TASK_CREATED",
      "detail": "Task \"Spec review\" created",
      "createdAt": "2025-08-31T18:37:55.152298Z"
    },
    {
      "id": 4,
      "type": "BOARD_MEMBER_REMOVED",
      "detail": "User carol@example.com removed from board",
      "createdAt": "2025-08-31T18:37:44.628804Z"
    },
    {
      "id": 3,
      "type": "BOARD_UPDATED",
      "detail": "Board renamed to \"Stage2 Board (official)\"",
      "createdAt": "2025-08-31T18:37:38.600647Z"
    },
    {
      "id": 2,
      "type": "BOARD_MEMBER_ADDED",
      "detail": "User carol@example.com added as VIEWER",
      "createdAt": "2025-08-31T18:37:13.750917Z"
    },
    {
      "id": 1,
      "type": "BOARD_MEMBER_ADDED",
      "detail": "User bob@example.com added as MEMBER",
      "createdAt": "2025-08-31T18:37:13.722748Z"
    }
  ],
  "number": 0,
  "pageable": {
    "pageNumber": 0,
    "pageSize": 20,
    "sort": {
      "empty": true,
      "unsorted": true,
      "sorted": false
    },
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": true,
    "unsorted": true,
    "sorted": false
  },
  "numberOfElements": 8,
  "empty": false
}
neha@Neharikas-MacBook-Air task-backend % A=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_B" -H "$CT"
\
  -d '{"title":"Write docs","description":"user manual","dueDate":"2030-02-01"}' | jq -r .id)
B=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_B" -H "$CT" \
  -d '{"title":"Fix bug 123","description":"null pointer","dueDate":"2030-01-15"}' | jq -r .id)
C=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_B" -H "$CT" \
  -d '{"title":"Prepare demo","description":"slides + script","dueDate":"2030-01-20"}' | jq -r .id)
ALICE_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_A" | jq -r .userId)
curl -s -X PUT "$BASE/api/tasks/$B/assignees" -H "$AUTH_A" -H "$CT" \
  -d "{\"userIds\":[$ALICE_ID]}" | jq . > /dev/null
curl -s -X PATCH "$BASE/api/tasks/$C" -H "$AUTH_B" -H "$CT" -d '{"status":"IN_PROGRESS"}' | jq . > /dev/null
curl -s "$BASE/api/tasks/search?boardId=$BOARD_ID&q=write&page=0&size=10" -H "$AUTH_A" | jq .
curl -s "$BASE/api/tasks/search?boardId=$BOARD_ID&status=DONE&page=0&size=10" -H "$AUTH_A" | jq .
curl -s "$BASE/api/tasks/search?boardId=$BOARD_ID&assigneeId=$ALICE_ID&page=0&size=10" -H "$AUTH_A" | jq .
curl -s "$BASE/api/tasks/search?boardId=$BOARD_ID&from=2030-01-01&to=2030-01-31&page=0&size=10" -H
"$AUTH_A" | jq .
curl -s "$BASE/api/tasks/search?boardId=$BOARD_ID&page=0&size=1" -H "$AUTH_A" | jq .
curl -s "$BASE/api/tasks/search?boardId=$BOARD_ID&page=1&size=1" -H "$AUTH_A" | jq .
{
  "totalPages": 1,
  "totalElements": 2,
  "first": true,
  "last": true,
  "size": 10,
  "content": [
    {
      "id": 3,
      "listId": 1,
      "boardId": 1,
      "title": "Write docs",
      "description": "user manual",
      "dueDate": "2030-02-01",
      "status": "TODO",
      "createdAt": "2025-08-31T18:38:01.482283Z",
      "completedAt": null
    },
    {
      "id": 2,
      "listId": 1,
      "boardId": 1,
      "title": "Write tests",
      "description": "stage2 tests",
      "dueDate": "2030-01-02",
      "status": "DONE",
      "createdAt": "2025-08-31T18:37:55.177575Z",
      "completedAt": "2025-08-31T18:37:55.203774Z"
    }
  ],
  "number": 0,
  "pageable": {
    "pageNumber": 0,
    "pageSize": 10,
    "sort": {
      "empty": false,
      "unsorted": false,
      "sorted": true
    },
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": false,
    "unsorted": false,
    "sorted": true
  },
  "numberOfElements": 2,
  "empty": false
}
{
  "totalPages": 1,
  "totalElements": 1,
  "first": true,
  "last": true,
  "size": 10,
  "content": [
    {
      "id": 2,
      "listId": 1,
      "boardId": 1,
      "title": "Write tests",
      "description": "stage2 tests",
      "dueDate": "2030-01-02",
      "status": "DONE",
      "createdAt": "2025-08-31T18:37:55.177575Z",
      "completedAt": "2025-08-31T18:37:55.203774Z"
    }
  ],
  "number": 0,
  "pageable": {
    "pageNumber": 0,
    "pageSize": 10,
    "sort": {
      "empty": false,
      "unsorted": false,
      "sorted": true
    },
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": false,
    "unsorted": false,
    "sorted": true
  },
  "numberOfElements": 1,
  "empty": false
}
{
  "totalPages": 1,
  "totalElements": 1,
  "first": true,
  "last": true,
  "size": 10,
  "content": [
    {
      "id": 4,
      "listId": 1,
      "boardId": 1,
      "title": "Fix bug 123",
      "description": "null pointer",
      "dueDate": "2030-01-15",
      "status": "TODO",
      "createdAt": "2025-08-31T18:38:01.503266Z",
      "completedAt": null
    }
  ],
  "number": 0,
  "pageable": {
    "pageNumber": 0,
    "pageSize": 10,
    "sort": {
      "empty": false,
      "unsorted": false,
      "sorted": true
    },
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": false,
    "unsorted": false,
    "sorted": true
  },
  "numberOfElements": 1,
  "empty": false
}
{
  "totalPages": 1,
  "totalElements": 3,
  "first": true,
  "last": true,
  "size": 10,
  "content": [
    {
      "id": 5,
      "listId": 1,
      "boardId": 1,
      "title": "Prepare demo",
      "description": "slides + script",
      "dueDate": "2030-01-20",
      "status": "IN_PROGRESS",
      "createdAt": "2025-08-31T18:38:01.520411Z",
      "completedAt": null
    },
    {
      "id": 4,
      "listId": 1,
      "boardId": 1,
      "title": "Fix bug 123",
      "description": "null pointer",
      "dueDate": "2030-01-15",
      "status": "TODO",
      "createdAt": "2025-08-31T18:38:01.503266Z",
      "completedAt": null
    },
    {
      "id": 2,
      "listId": 1,
      "boardId": 1,
      "title": "Write tests",
      "description": "stage2 tests",
      "dueDate": "2030-01-02",
      "status": "DONE",
      "createdAt": "2025-08-31T18:37:55.177575Z",
      "completedAt": "2025-08-31T18:37:55.203774Z"
    }
  ],
  "number": 0,
  "pageable": {
    "pageNumber": 0,
    "pageSize": 10,
    "sort": {
      "empty": false,
      "unsorted": false,
      "sorted": true
    },
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": false,
    "unsorted": false,
    "sorted": true
  },
  "numberOfElements": 3,
  "empty": false
}
{
  "totalPages": 4,
  "totalElements": 4,
  "first": true,
  "last": false,
  "size": 1,
  "content": [
    {
      "id": 5,
      "listId": 1,
      "boardId": 1,
      "title": "Prepare demo",
      "description": "slides + script",
      "dueDate": "2030-01-20",
      "status": "IN_PROGRESS",
      "createdAt": "2025-08-31T18:38:01.520411Z",
      "completedAt": null
    }
  ],
  "number": 0,
  "pageable": {
    "pageNumber": 0,
    "pageSize": 1,
    "sort": {
      "empty": false,
      "unsorted": false,
      "sorted": true
    },
    "offset": 0,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": false,
    "unsorted": false,
    "sorted": true
  },
  "numberOfElements": 1,
  "empty": false
}
{
  "totalPages": 4,
  "totalElements": 4,
  "first": false,
  "last": false,
  "size": 1,
  "content": [
    {
      "id": 4,
      "listId": 1,
      "boardId": 1,
      "title": "Fix bug 123",
      "description": "null pointer",
      "dueDate": "2030-01-15",
      "status": "TODO",
      "createdAt": "2025-08-31T18:38:01.503266Z",
      "completedAt": null
    }
  ],
  "number": 1,
  "pageable": {
    "pageNumber": 1,
    "pageSize": 1,
    "sort": {
      "empty": false,
      "unsorted": false,
      "sorted": true
    },
    "offset": 1,
    "unpaged": false,
    "paged": true
  },
  "sort": {
    "empty": false,
    "unsorted": false,
    "sorted": true
  },
  "numberOfElements": 1,
  "empty": false
}
neha@Neharikas-MacBook-Air task-backend % Validate that non-owners cannot escalate roles or create resources
they shouldn't.
curl -i -s -X POST "$BASE/api/boards/$BOARD_ID/members?userId=$ALICE_ID&role=VIEWER" -H "$AUTH_B" | sed
-n '1,8p'
curl -s -X POST "$BASE/api/boards/$BOARD_ID/members?userId=$CAROL_ID&role=VIEWER" -H "$AUTH_A" | jq .
curl -i -s -X POST "$BASE/api/boards/$BOARD_ID/lists" -H "$AUTH_C" -H "$CT" -d '{"name":"Nope","position":3}' | sed
-n '1,8p'
curl -s "$BASE/api/boards/$BOARD_ID" -H "$AUTH_C" | jq .
zsh: command not found: Validate
HTTP/1.1 403
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
{
  "id": 4,
  "role": "VIEWER"
}
HTTP/1.1 403
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
{
  "id": 1,
  "name": "Stage2 Board (official)",
  "ownerId": 1,
  "lists": [
    {
      "id": 1,
      "name": "To Do",
      "position": 1,
      "tasks": [
        {
          "id": 2,
          "title": "Write tests",
          "status": "DONE"
        },
        {
          "id": 3,
          "title": "Write docs",
          "status": "TODO"
        },
        {
          "id": 4,
          "title": "Fix bug 123",
          "status": "TODO"
        },
        {
          "id": 5,
          "title": "Prepare demo",
          "status": "IN_PROGRESS"
        }
      ]
    },
    {
      "id": 2,
      "name": "In Progress",
      "position": 2,
      "tasks": []
    }
  ]
}
neha@Neharikas-MacBook-Air task-backend %
```

Stage 3;

```
Last login: Sun Aug 31 14:35:31 on ttys002
neha@Neharikas-MacBook-Air task-backend % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
neha@Neharikas-MacBook-Air task-backend % docker run -d --name task_db -e POSTGRES_PASSWORD=secret -p
5432:5432 postgres:16 || true
docker rm -f mailhog 2>/dev/null || true
docker run -d --name mailhog -p 1025:1025 -p 8025:8025 mailhog/mailhog
docker rm -f redis 2>/dev/null || true
docker run -d --name redis -p 6379:6379 redis:7
docker: Error response from daemon: Conflict. The container name "/task_db" is already in use by container
"e1209e70531d604ebdfc283863fd086ae92fc6fc7fcd0cc33680ef402c32747e". You have to remove (or rename) that
container to be able to reuse that name.
Run 'docker run --help' for more information
mailhog
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8)
and no specific platform was requested
c5c09d7c5edcff3a1186ef982ed523d6c4e317861b5a100e6151907db619b417
redis
45859eedcaed86d59f55abc41fc8f9abdd4a058988b07d38b31f13f8925c4dc0
neha@Neharikas-MacBook-Air task-backend % docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED         STATUS         PORTS
NAMES
45859eedcaed   redis:7           "docker-entrypoint.s…"   5 seconds ago   Up 5 seconds   0.0.0.0:6379->6379/tcp,
[::]:6379->6379/tcp                                        redis
c5c09d7c5edc   mailhog/mailhog   "MailHog"                6 seconds ago   Up 5 seconds   0.0.0.0:1025->1025/tcp,
[::]:1025->1025/tcp, 0.0.0.0:8025->8025/tcp, [::]:8025->8025/tcp   mailhog
neha@Neharikas-MacBook-Air task-backend % open http://localhost:8025 || xdg-open http://localhost:8025 || true
neha@Neharikas-MacBook-Air task-backend % >....
TOKEN_B=$(curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
  -d '{"email":"bob@example.com","password":"secret123","name":"Bob"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" -d '{"email":"bob@example.com","password":"secret123"}' | jq -r .token)
AUTH_B="Authorization: Bearer $TOKEN_B"
TOKEN_C=$(curl -sS -f "$BASE/api/auth/signup" -H "$CT" \
  -d '{"email":"carol@example.com","password":"secret123","name":"Carol"}' | jq -r .token 2>/dev/null \
  || curl -sS -f "$BASE/api/auth/login" -H "$CT" -d '{"email":"carol@example.com","password":"secret123"}' | jq -r .token)
AUTH_C="Authorization: Bearer $TOKEN_C"
ALICE_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_A" | jq -r .userId)
BOB_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_B" | jq -r .userId)
CAROL_ID=$(curl -s "$BASE/api/auth/me" -H "$AUTH_C" | jq -r .userId)
echo "ALICE=$ALICE_ID  BOB=$BOB_ID  CAROL=$CAROL_ID"
title "Create board and memberships"
BOARD_ID=$(curl -s "$BASE/api/boards" -H "$AUTH_A" -H "$CT" -d '{"name":"Stage3 Board"}' | jq -r .id)
echo "BOARD_ID=$BOARD_ID"
```

```
curl -s -X POST "$BASE/api/boards/$BOARD_ID/members?userId=$BOB_ID&role=MEMBER" -H "$AUTH_A" > /dev/null
curl -s -X POST "$BASE/api/boards/$BOARD_ID/members?userId=$CAROL_ID&role=VIEWER" -H "$AUTH_A" > /dev/null
echo "Added Bob as MEMBER, Carol as VIEWER"
title "Create lists and a task"
TODO_ID=$(curl -s "$BASE/api/boards/$BOARD_ID/lists" -H "$AUTH_A" -H "$CT" -d '{"name":"To Do","position":1}' | jq -r .id)
DOING_ID=$(curl -s "$BASE/api/boards/$BOARD_ID/lists" -H "$AUTH_A" -H "$CT" -d '{"name":"In Progress","position":2}' | jq -r .id)
echo "TODO_ID=$TODO_ID  DOING_ID=$DOING_ID"
TASK_ID=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_A" -H "$CT" -d '{"title":"Stage3 seed","description":"for RT/email tests","dueDate":"2030-01-01"}' | jq -r .id)
echo "TASK_ID=$TASK_ID"
title "Assignment email test (MailHog should receive 1 email to bob@example.com)"
BEFORE=$(mh_count || echo 0)
echo "MailHog messages before: $BEFORE"
curl -s -X PUT "$BASE/api/tasks/$TASK_ID/assignees" -H "$AUTH_A" -H "$CT" -d "{\"userIds\":[$BOB_ID]}" > /dev/null
sleep 2
AFTER=$(mh_count || echo 0)
echo "MailHog messages after:  $AFTER"
echo "Last messages:"; mh_last_n 3
if [ "$AFTER" -le "$BEFORE" ]; then
  echo "WARNING: MailHog count did not increase. Check spring.mail.* config and MailHog on localhost:1025" >&2
fi
title "Due-soon reminder test (create task due tomorrow and assign Bob)"
TOMORROW=$(tomorrow_ymd)
echo "Tomorrow (UTC) = $TOMORROW"
REM_TASK_ID=$(curl -s "$BASE/api/lists/$TODO_ID/tasks" -H "$AUTH_A" -H "$CT" \
  -d "{\"title\":\"Due soon\",\"description\":\"reminder test\",\"dueDate\":\"$TOMORROW\"}" | jq -r .id)
curl -s -X PUT "$BASE/api/tasks/$REM_TASK_ID/assignees" -H "$AUTH_A" -H "$CT" -d "{\"userIds\":[$BOB_ID]}" > /dev/null
echo "Waiting 12s for scheduler (adjust if your @Scheduled frequency is longer)..."
sleep 12
AFTER2=$(mh_count || echo 0)
echo "MailHog messages now:   $AFTER2"
echo "Last messages:"; mh_last_n 5
echo
echo "DONE. Next: open a realtime stream (SSE or WS) in another terminal and then mutate tasks here to see events."
SH
chmod +x stage3-test.sh
./stage3-test.sh
=== Signing up / logging in users ===
ALICE=1  BOB=2  CAROL=3
=== Create board and memberships ===
BOARD_ID=1
Added Bob as MEMBER, Carol as VIEWER
=== Create lists and a task ===
TODO_ID=1  DOING_ID=2
TASK_ID=1
=== Assignment email test (MailHog should receive 1 email to bob@example.com) ===
MailHog messages before: 0
MailHog messages after:  1
Last messages:
[
  {
    "subj": "[Task Assigned] Stage3 seed",
    "to": "bob@example.com",
    "from": "",
    "date": "2025-08-31T18:39:31.999489Z"
  }
]
```

```
=== Due-soon reminder test (create task due tomorrow and assign Bob) ===
Tomorrow (UTC) = 2025-09-01
Waiting 12s for scheduler (adjust if your @Scheduled frequency is longer)...
MailHog messages now:   4
Last messages:
[
  {
    "subj": "[Task Due Soon] Due soon",
    "to": "bob@example.com",
    "from": "",
    "date": "2025-08-31T18:39:41.741547Z"
  },
  {
    "subj": "[Task Due Soon] Due soon",
    "to": "bob@example.com",
    "from": "",
    "date": "2025-08-31T18:39:36.711544Z"
  },
  {
    "subj": "[Task Assigned] Due soon",
    "to": "bob@example.com",
    "from": "",
    "date": "2025-08-31T18:39:34.182004Z"
  },
  {
    "subj": "[Task Assigned] Stage3 seed",
    "to": "bob@example.com",
    "from": "",
    "date": "2025-08-31T18:39:31.999489Z"
  }
]
DONE. Next: open a realtime stream (SSE or WS) in another terminal and then mutate tasks here to see events.
neha@Neharikas-MacBook-Air task-backend % curl -s "$BASE/api/lists/1/tasks" -H "$AUTH_A" -H "$CT" \
  -d '{"title":"RT check","description":"sse","dueDate":"2030-03-01"}' | jq .
TASK_ID=2
curl -s -X PATCH "$BASE/api/tasks/$TASK_ID" -H "$AUTH_A" -H "$CT" -d '{"status":"IN_PROGRESS"}' | jq .
curl -s -X PUT "$BASE/api/boards/$BOARD_ID" -H "$AUTH_A" -H "$CT" \
  -d '{"name":"Stage3 Board (rt)"}' | jq .
neha@Neharikas-MacBook-Air task-backend %
neha@Neharikas-MacBook-Air task-backend %
```

```
Last login: Sun Aug 31 14:38:47 on ttys003
neha@Neharikas-MacBook-Air task-backend % BASE="http://localhost:8080"
CT="Content-Type: application/json"
TOKEN_A=$(curl -sS -f "$BASE/api/auth/login" -H "$CT" -d '{"email":"alice@example.com","password":"secret123"}' | jq -r .token)
AUTH_A="Authorization: Bearer $TOKEN_A"
BOARD_ID=$(curl -s "$BASE/api/boards" -H "$AUTH_A" | jq -r '.[0].id')
echo "BOARD_ID=$BOARD_ID"
curl -N -H "$AUTH_A" "$BASE/api/boards/$BOARD_ID/stream"
BOARD_ID=1
event:connected
data:{"ts":"2025-08-31T18:39:53.253168Z","boardId":1}
id:1756665593253
retry:2000
^C
neha@Neharikas-MacBook-Air task-backend %
```

Stage 4;

```
Last login: Sun Aug 31 14:44:43 on ttys002
neha@Neharikas-MacBook-Air task-backend % export BASE="http://localhost:8080"
```

```
export CT="Content-Type: application/json"
neha@Neharikas-MacBook-Air task-backend %
curl -s -f "$BASE/api/auth/signup" -H "$CT" -d '{"email":"alice@example.com","password":"secret123","name":"Alice"}'
>/dev/null || true
TOKEN_A=$(curl -s -f "$BASE/api/auth/login" -H "$CT" -d '{"email":"alice@example.com","password":"secret123"}' | jq
-r .token)
export AUTH_A="Authorization: Bearer $TOKEN_A"
neha@Neharikas-MacBook-Air task-backend % BOARD_ID=$(curl -s -X POST "$BASE/api/boards" -H "$AUTH_A" -H
"$CT" -d '{"name":"Stage4 Board"}' | jq -r .id)
export BOARD_ID
neha@Neharikas-MacBook-Air task-backend % LIST_ID=$(curl -s -X POST "$BASE/api/boards/$BOARD_ID/lists" -H
"$AUTH_A" -H "$CT" -d '{"name":"To Do","position":1}' | jq -r .id)
export LIST_ID
neha@Neharikas-MacBook-Air task-backend %
TASK_ID=$(curl -s -X POST "$BASE/api/lists/$LIST_ID/tasks" -H "$AUTH_A" -H "$CT" -d '{"title":"Baseline
task","description":"ok","dueDate":"2030-01-01"}' | jq -r .id)
export TASK_ID
cat > cache-test.sh <<'SH'
set -euo pipefail
: "${BASE:=http://localhost:8080}"
: "${AUTH_A:?missing AUTH_A}"
: "${BOARD_ID:?missing BOARD_ID}"
curl -s -o /dev/null -w "cold: %{time_total}\n"  -H "$AUTH_A" "$BASE/api/boards/$BOARD_ID"
curl -s -o /dev/null -w "warm1: %{time_total}\n" -H "$AUTH_A" "$BASE/api/boards/$BOARD_ID"
curl -s -o /dev/null -w "warm2: %{time_total}\n" -H "$AUTH_A" "$BASE/api/boards/$BOARD_ID"
SH
chmod +x cache-test.sh
./cache-test.sh
curl -s -X PUT "$BASE/api/boards/$BOARD_ID" -H "$AUTH_A" -H "$CT" -d '{"name":"Stage4 Board cache test"}'
>/dev/null
./cache-test.sh
curl -i -s -X PATCH "$BASE/api/tasks/$TASK_ID" -H "$AUTH_A" -H "$CT" -d '{"status":"NOT_A_STATUS"}' | sed -n
'1,25p'
BAD_ID=$(curl -s "$BASE/api/lists/$LIST_ID/tasks" -H "$AUTH_A" -H "$CT" -d
'{"title":"<script>alert(1)</script>","description":"xss","dueDate":"2030-01-01"}' | jq -r .id)
curl -s -H "$AUTH_A" "$BASE/api/lists/$LIST_ID/tasks" | jq .
python3 - <<'PY' | curl -i -s "$BASE/api/lists/$LIST_ID/tasks" -H "$AUTH_A" -H "$CT" -d @- | sed -n '1,25p'
print('{"title":"' + "A"*10000 + '","description":"too long","dueDate":"2030-01-01"}')
PY
cat > rate-burst.sh <<'SH'
set -euo pipefail
: "${BASE:=http://localhost:8080}"
: "${AUTH_A:?missing AUTH_A}"
: "${BOARD_ID:?missing BOARD_ID}"
seq 1 150 | xargs -n1 -P16 -I{} curl -s -o /dev/null -w "%{http_code}\n" -H "$AUTH_A" "$BASE/api/boards/$BOARD_ID" |
sort | uniq -c
SH
chmod +x rate-burst.sh
./rate-burst.sh
cold: 0.032583
warm1: 0.010299
warm2: 0.008619
cold: 0.009676
warm1: 0.008324
warm2: 0.007910
HTTP/1.1 400
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
```

```
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json
Transfer-Encoding: chunked
Date: Sun, 31 Aug 2025 18:45:51 GMT
Connection: close
{"path":"/api/tasks/1","status":400,"error":"Bad Request","message":"JSON parse error: Cannot deserialize value of type `com.example.taskmanager.task.TaskStatus` from String \"NOT_A_STATUS\": not one of the values accepted for Enum class: [DONE, TODO, IN_PROGRESS]"}[
  {
    "id": 1,
    "title": "Baseline task",
    "description": "ok",
    "dueDate": "2030-01-01",
    "status": "TODO",
    "createdAt": "2025-08-31T18:45:51.798287Z",
    "completedAt": null
  },
  {
    "id": 2,
    "title": "",
    "description": "xss",
    "dueDate": "2030-01-01",
    "status": "TODO",
    "createdAt": "2025-08-31T18:45:52.025040Z",
    "completedAt": null
  }
]
HTTP/1.1 400
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json
Transfer-Encoding: chunked
Date: Sun, 31 Aug 2025 18:45:51 GMT
Connection: close
{"path":"/api/lists/1/tasks","status":400,"error":"Bad Request","message":"title: size must be between 0 and 255"}  84 200
  66 429
neha@Neharikas-MacBook-Air task-backend % TOKEN_ADMIN=$(curl -s -f "$BASE/api/auth/login" -H "$CT" -d '{"email":"alice@example.com","password":"secret123"}' | jq -r .token)
export AUTH_ADMIN="Authorization: Bearer $TOKEN_ADMIN"
curl -s -H "$AUTH_ADMIN" "$BASE/api/admin/analytics/board-task-counts" | jq .
curl -s -H "$AUTH_ADMIN" "$BASE/api/admin/analytics/avg-completion-per-board" | jq .
curl -s -H "$AUTH_ADMIN" "$BASE/api/admin/analytics/most-active-users?limit=5" | jq .
[
  {
    "boardId": 1,
    "boardName": "Stage4 Board cache test",
    "todo": 2,
    "inProgress": 0,
    "done": 0
  }
]
[]
[
  {
    "userId": 1,
    "email": "alice@example.com",
    "activityCount": 3
  }
]
neha@Neharikas-MacBook-Air task-backend % curl -s -X POST "$BASE/api/lists/$LIST_ID/tasks" -H "$AUTH_A" -H
"$CT" -d '{"title":"Finish docs","description":"x","dueDate":"2030-01-02"}' >/dev/null
curl -s -X POST "$BASE/api/lists/$LIST_ID/tasks" -H "$AUTH_A" -H "$CT" -d '{"title":"Ship
v1","description":"y","dueDate":"2030-01-03"}' >/dev/null
curl -s -X PATCH "$BASE/api/tasks/$TASK_ID" -H "$AUTH_A" -H "$CT" -d '{"status":"DONE"}' >/dev/null
curl -s -H "$AUTH_ADMIN" "$BASE/api/admin/analytics/board-task-counts" | jq .
curl -s -H "$AUTH_ADMIN" "$BASE/api/admin/analytics/avg-completion-per-board" | jq .
[
  {
    "boardId": 1,
    "boardName": "Stage4 Board cache test",
    "todo": 3,
    "inProgress": 0,
    "done": 1
  }
]
[
  {
    "boardId": 1,
    "boardName": "Stage4 Board cache test",
    "avgHours": 0.010833333333333334
  }
]
neha@Neharikas-MacBook-Air task-backend %
neha@Neharikas-MacBook-Air task-backend %
```