

flyingcloud4156 / flyingcloud-4156-project

Code Issues 6 Pull requests 1 Discussions Actions Projects 2 Wiki Security Insights Settings

← MyASE move #59

Re-run all jobs ⋮

Summary

Jobs

Style & Static Analysis (Service)

Unit / API / Integration Tests & ...

Run details

Usage

Workflow file

Unit / API / Integration Tests & Coverage (Service)

succeeded 36 minutes ago in 3m 12s

Search logs ⋮

- Set up job 2s
- Initialize containers 31s
- Checkout repository 1s
- Setup JDK 17 2s
- Wait for MySQL to be ready 0s
- Initialize database schema and test data 1s
- Run API tests (api_all) 30s
 - Run cd API_test
 - ====
 - => 0) Resetting database and starting Spring Boot app
 - ====
 - Dropping database ledger ...
 - 19 Warning: arning] Using a password on the command line interface can be insecure.
 - 20 Creating database ledger ...
 - 21 Warning: arning] Using a password on the command line interface can be insecure.
 - 22 Loading schema file: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/ledger_flow.sql
 - 23 Warning: arning] Using a password on the command line interface can be insecure.
 - 24 Loading seed data (big file): /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/backup/ledger_big_seed.sql
 - 25 Warning: arning] Using a password on the command line interface can be insecure.
 - 26 [OK] Database is ready.
 - 27 Starting Spring Boot application in background...
 - 28 Working directory: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project
 - 29 Profile: test
 - 30 Spring Boot started in background with PID: 2957
 - 31 Waiting for app to be ready...
 - 32 Waiting... (2/120 seconds) - HTTP code: 000000
 - 33 Waiting... (4/120 seconds) - HTTP code: 000000
 - 34 Waiting... (6/120 seconds) - HTTP code: 000000
 - 35 Waiting... (8/120 seconds) - HTTP code: 000000
 - 36 Waiting... (10/120 seconds) - HTTP code: 000000
 - 37 Waiting... (12/120 seconds) - HTTP code: 000000
 - 38 Waiting... (14/120 seconds) - HTTP code: 000000
 - 39 Waiting... (16/120 seconds) - HTTP code: 000000
 - 40 Waiting... (18/120 seconds) - HTTP code: 000000
 - 41 Waiting... (20/120 seconds) - HTTP code: 000000
 - 42 Waiting... (22/120 seconds) - HTTP code: 000000
 - 43 Waiting... (24/120 seconds) - HTTP code: 000000
 - 44 Waiting... (26/120 seconds) - HTTP code: 000000
 - 45 [OK] Spring Boot app is ready!
 - 46
 - 47 =====
 - 48 => 1) AUTH + USER APIs
 - 49 =====
 - 50
 - 51 --- 1.1 Using seed users (Alice / Bob / Charlie already exist)
 - 52
 - 53 --- 1.2 Login Alice
 - 54
 - 55 >>> HTTP POST <http://127.0.0.1:8081/api/v1/auth/login>
 - 56 >>> Request Body (raw):
 - 57 {
 - 58 "email": "alice@gmail.com",
 - 59 "password": "Passw0rd!"
 - 60 }
 - 61 <<< HTTP Code: 200
 - 62 <<< Response Body:
 - 63 {
 - 64 "success": true,
 - 65 "message": "OK",
 - 66 "data": {
 - 67 "access_token": "I5fdpq9l3kslALwMYM4HMiPAfjy3lchBFh9mw00hdmw",
 - 68 "refresh_token": "n2D9uoJ--PtHsfEZkbK-Wq2Sa1VT1658qx40uQiaIP8"
 - 69 },
 - 70 "total": null
 - 71 }
 - 72 <<< END
 - 73
 - 74 --- 1.3 /users/me
 - 75
 - 76 >>> HTTP GET <http://127.0.0.1:8081/api/v1/users/me>
 - 77 >>> Header: X-Auth-Token: *****

 - 78 >>> Request Body: (empty)
 - 79 <<< HTTP Code: 200
 - 80 <<< Response Body:
 - 81 {
 - 82 "success": true

```

 94     "success": true,
 95     "message": "OK",
 96     "data": {
 97       "id": 1,
 98       "name": "Alice"
 99     }
100   <<< END
101
102   --- 1.4 /user-lookup (lookup Bob by email)
103
104  >>> HTTP GET http://127.0.0.1:8081/api/v1/user-lookup?email=bob@gmail.com
105  >>> Header: X-Auth-Token: *****(hidden)*****
106  >>> Request Body: (empty)
107  <<< HTTP Code: 200
108  <<< Response Body:
109  {
110    "success": true,
111    "message": "OK",
112    "data": {
113      "user_id": 2,
114      "name": "Bob"
115    },
116    "total": null
117  }
118 <<< END
119
120   --- 1.5 /users/{id} (Bob profile)
121
122  >>> HTTP GET http://127.0.0.1:8081/api/v1/users/2
123  >>> Header: X-Auth-Token: *****(hidden)*****
124  >>> Request Body: (empty)
125  <<< HTTP Code: 200
126  <<< Response Body:
127  {
128    "success": true,
129    "message": "OK",
130    "data": {
131      "id": 2,
132      "name": "Bob"
133    },
134    "total": null
135  }
136 <<< END
137
138   --- 1.6 refresh token
139
140  >>> HTTP POST http://127.0.0.1:8081/api/v1/auth/refresh?refreshToken=n2D9uoJ--PtHsfeZkbK-Wq2Sa1VT1658gx40u0iaIP8
141  >>> Request Body: (empty)
142  <<< HTTP Code: 200
143  <<< Response Body:
144  {
145    "success": true,
146    "message": "OK",
147    "data": {
148      "access_token": "BqR2y32mxiwPU432tZ5wMhz0UcNr3mCv_3p6Z7Sxb70",
149      "refresh_token": "04kkDcSKr-j7BtGCj85QKPk8Ych7oBAXSNuK-XMd7C4"
150    },
151    "total": null
152  }
153 <<< END
154
155 =====
156 => 2) LEDGER APIs
157 =====
158
159   --- 2.1 Create ledger
160
161  >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers
162  >>> Header: X-Auth-Token: *****(hidden)*****
163  >>> Request Body (raw):
164  {
165    "name": "Road Trip Fund - 1764900710",
166    "ledger_type": "GROUP_BALANCE",
167    "base_currency": "USD",
168    "categories": [
169      {
170        "name": "Gas",
171        "kind": "EXPENSE"
172      },
173      {
174        "name": "Food",
175        "kind": "EXPENSE"
176      }
177    ]
178  }
179 <<< HTTP Code: 201
180 <<< Response Body:
181  {
182    "success": true,
183    "message": "OK",
184    "data": {
185      "ledger_id": 3,
186      "name": "Road Trip Fund - 1764900710",
187      "ledger_type": "GROUP_BALANCE",
188      "base_currency": "USD",
189      "share_start_date": null,
190      "role": "OWNER",
191      "share_end_date": null
192    }
193  }

```

```

181     "categories": [
182     {
183         "id": 12,
184         "ledger_id": 3,
185         "name": "Food",
186         "kind": "EXPENSE",
187         "is_active": true
188     },
189     {
190         "id": 11,
191         "ledger_id": 3,
192         "name": "Gas",
193         "kind": "EXPENSE",
194         "is_active": true
195     }
196   ]
197 },
198 "total": null
199 }
200 <<< END
201
202 --- 2.2 Get my ledgers (/ledgers/mine)
203
204 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/mine
205 >>> Header: X-Auth-Token: *****()*****
206 >>> Request Body: (empty)
207 <<< HTTP Code: 200
208 <<< Response Body:
209 {
210     "success": true,
211     "message": "OK",
212     "data": [
213         "items": [
214             {
215                 "ledger_id": 1,
216                 "name": "Road Trip Demo",
217                 "ledger_type": "GROUP_BALANCE",
218                 "base_currency": "USD",
219                 "role": "OWNER"
220             },
221             {
222                 "ledger_id": 3,
223                 "name": "Road Trip Fund - 1764900710",
224                 "ledger_type": "GROUP_BALANCE",
225                 "base_currency": "USD",
226                 "role": "OWNER"
227             }
228         ]
229     },
230     "total": null
231 }
232 <<< END
233
234 --- 2.3 Get ledger details (/ledgers/{id})
235
236 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3
237 >>> Header: X-Auth-Token: *****()*****
238 >>> Request Body: (empty)
239 <<< HTTP Code: 200
240 <<< Response Body:
241 {
242     "success": true,
243     "message": "OK",
244     "data": [
245         "ledger_id": 3,
246         "name": "Road Trip Fund - 1764900710",
247         "ledger_type": "GROUP_BALANCE",
248         "base_currency": "USD",
249         "share_start_date": null,
250         "role": "OWNER",
251         "categories": [
252             {
253                 "id": 12,
254                 "ledger_id": 3,
255                 "name": "Food",
256                 "kind": "EXPENSE",
257                 "is_active": true
258             },
259             {
260                 "id": 11,
261                 "ledger_id": 3,
262                 "name": "Gas",
263                 "kind": "EXPENSE",
264                 "is_active": true
265             }
266         ]
267     },
268     "total": null
269 }
270 <<< END
271
272 --- 2.4 Add Bob + Charlie as members
273 Warning: arning] Using a password on the command line interface can be insecure.
274
275 >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers/3/members
276 >>> Header: X-Auth-Token: *****()*****
277 >>> Request Body (raw):
278 {
279     "user_id": 2,

```

```

280     "role": "EDITOR"
281   }
282 <<< HTTP Code: 201
283 <<< Response Body:
284   {
285     "success": true,
286     "message": "OK",
287     "data": {
288       "ledger_id": 3,
289       "user_id": 2,
290       "role": "EDITOR"
291     },
292     "total": null
293   }
294 <<< END
295
296 >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers/3/members
297 >>> Header: X-Auth-Token: *****()*****
298 >>> Request Body (raw):
299   {
300     "user_id": 3,
301     "role": "EDITOR"
302   }
303 <<< HTTP Code: 201
304 <<< Response Body:
305   {
306     "success": true,
307     "message": "OK",
308     "data": {
309       "ledger_id": 3,
310       "user_id": 3,
311       "role": "EDITOR"
312     },
313     "total": null
314   }
315 <<< END
316
317 --- 2.5 List members
318
319 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3/members
320 >>> Header: X-Auth-Token: *****()*****
321 >>> Request Body: (empty)
322 <<< HTTP Code: 200
323 <<< Response Body:
324   {
325     "success": true,
326     "message": "OK",
327     "data": {
328       "items": [
329         {
330           "user_id": 1,
331           "name": "Alice",
332           "role": "OWNER"
333         },
334         {
335           "user_id": 2,
336           "name": "Bob",
337           "role": "EDITOR"
338         },
339         {
340           "user_id": 3,
341           "name": "Charlie",
342           "role": "EDITOR"
343         }
344       ],
345     },
346     "total": null
347   }
348 <<< END
349
350 --- 2.6 Remove Charlie then add back
351
352 >>> HTTP DELETE http://127.0.0.1:8081/api/v1/ledgers/3/members/3
353 >>> Header: X-Auth-Token: *****()*****
354 >>> Request Body: (empty)
355 <<< HTTP Code: 204
356 <<< Response Body: (empty)
357 <<< END
358
359 >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers/3/members
360 >>> Header: X-Auth-Token: *****()*****
361 >>> Request Body (raw):
362   {
363     "user_id": 3,
364     "role": "EDITOR"
365   }
366 <<< HTTP Code: 201
367 <<< Response Body:
368   {
369     "success": true,
370     "message": "OK",
371     "data": {
372       "ledger_id": 3,
373       "user_id": 3,
374       "role": "EDITOR"
375     },
376     "total": null
377   }
378 <<< END

```

```

379
380 =====
381 => 3) TRANSACTION APIs
382 =====
383
384 --- 3.1 Create EXPENSE (EXACT split) - 2025-09
385
386 >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers/3/transactions
387 >>> Header: X-Auth-Token: *****()*****
388 >>> Request Body (raw):
389 {
390     "txn_at": "2025-09-10T08:00:00",
391     "type": "EXPENSE",
392     "payer_id": 1,
393     "amount_total": "150.00",
394     "currency": "USD",
395     "note": "Gas and tolls (test)",
396     "is_private": false,
397     "rounding_strategy": "NONE",
398     "tail_allocation": "PAYER",
399     "splits": [
400         {
401             "user_id": 1,
402             "split_method": "EXACT",
403             "share_value": 75.00,
404             "included": true
405         },
406         {
407             "user_id": 2,
408             "split_method": "EXACT",
409             "share_value": 50.00,
410             "included": true
411         },
412         {
413             "user_id": 3,
414             "split_method": "EXACT",
415             "share_value": 25.00,
416             "included": true
417         }
418     ]
419 }
420 <<< HTTP Code: 201
421 <<< Response Body:
422 {
423     "success": true,
424     "message": "OK",
425     "data": {
426         "transaction_id": 111,
427         "budget_alert": null
428     },
429     "total": null
430 }
431 <<< END
432
433 --- 3.2 Create INCOME - 2025-09
434
435 >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers/3/transactions
436 >>> Header: X-Auth-Token: *****()*****
437 >>> Request Body (raw):
438 {
439     "txn_at": "2025-09-11T08:00:00",
440     "type": "INCOME",
441     "payer_id": 1,
442     "amount_total": "100.00",
443     "currency": "USD",
444     "note": "Refund (test)",
445     "is_private": false,
446     "rounding_strategy": "NONE",
447     "tail_allocation": "PAYER",
448     "splits": [
449         {
450             "user_id": 1,
451             "split_method": "EXACT",
452             "share_value": 34.00,
453             "included": true
454         },
455         {
456             "user_id": 2,
457             "split_method": "EXACT",
458             "share_value": 33.00,
459             "included": true
460         },
461         {
462             "user_id": 3,
463             "split_method": "EXACT",
464             "share_value": 33.00,
465             "included": true
466         }
467     ]
468 }
469 <<< HTTP Code: 201
470 <<< Response Body:
471 {
472     "success": true,
473     "message": "OK",
474     "data": {
475         "transaction_id": 112,
476         "budget_alert": null
477     },

```

```

478     "total": null
479   }
480 <<< END
481
482 --- 3.3 Create another EXPENSE in October (for trend)
483
484 >>> HTTP POST http://127.0.0.1:8081/api/v1/ledgers/3/transactions
485 >>> Header: X-Auth-Token: *****(hidden)*****
486 >>> Request Body (raw):
487 {
488   "txn_at": "2025-10-05T18:00:00",
489   "type": "EXPENSE",
490   "payer_id": 2,
491   "amount_total": "240.00",
492   "currency": "USD",
493   "note": "Groceries (test)",
494   "is_private": false,
495   "rounding_strategy": "NONE",
496   "tail_allocation": "PAYER",
497   "splits": [
498     {
499       "user_id": 1,
500       "split_method": "EXACT",
501       "share_value": 80.00,
502       "included": true
503     },
504     {
505       "user_id": 2,
506       "split_method": "EXACT",
507       "share_value": 80.00,
508       "included": true
509     },
510     {
511       "user_id": 3,
512       "split_method": "EXACT",
513       "share_value": 80.00,
514       "included": true
515     }
516   ]
517 }
518 <<< HTTP Code: 201
519 <<< Response Body:
520 {
521   "success": true,
522   "message": "OK",
523   "data": {
524     "transaction_id": 113,
525     "budget_alert": null
526   },
527   "total": null
528 }
529 <<< END
530
531 --- 3.4 Get transaction detail
532
533 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3/transactions/111
534 >>> Header: X-Auth-Token: *****(hidden)*****
535 >>> Request Body: (empty)
536 <<< HTTP Code: 200
537 <<< Response Body:
538 {
539   "success": true,
540   "message": "OK",
541   "data": {
542     "transaction_id": 111,
543     "ledger_id": 3,
544     "txn_at": [
545       2025,
546       9,
547       10,
548       8,
549       0
550     ],
551     "type": "EXPENSE",
552     "currency": "USD",
553     "amount_total": 150.0000000,
554     "category_id": null,
555     "note": "Gas and tolls (test)",
556     "payer_id": 1,
557     "created_by": 1,
558     "rounding_strategy": "NONE",
559     "tail_allocation": "PAYER",
560     "splits": [
561       {
562         "user_id": 1,
563         "split_method": "EXACT",
564         "share_value": 75.0000000,
565         "included": true,
566         "computed_amount": 75.0000000
567       },
568       {
569         "user_id": 2,
570         "split_method": "EXACT",
571         "share_value": 50.0000000,
572         "included": true,
573         "computed_amount": 50.0000000
574       },
575       {
576         "user_id": 3,
577       }
578     ]
579   }
580 }

```

```

577     "split_method": "EXACT",
578     "share_value": 25.00000000,
579     "included": true,
580     "computed_amount": 25.00000000
581   }
582 ],
583 "edges_preview": [
584   {
585     "from_user_id": 1,
586     "to_user_id": 2,
587     "amount": 50.00000000,
588     "edge_currency": "USD"
589   },
590   {
591     "from_user_id": 1,
592     "to_user_id": 3,
593     "amount": 25.00000000,
594     "edge_currency": "USD"
595   }
596 ],
597 },
598 "total": null
599 }
600 <<< END
601
602 --- 3.5 List transactions (paged)
603
604 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3/transactions?page=1&size=20
605 >>> Header: X-Auth-Token: *****(@hidden)*****
606 >>> Request Body: (empty)
607 <<< HTTP Code: 200
608 <<< Response Body:
609 {
610   "success": true,
611   "message": "OK",
612   "data": {
613     "page": 1,
614     "size": 20,
615     "total": 3,
616     "items": [
617       {
618         "transaction_id": 113,
619         "txn_at": [
620           2025,
621           10,
622           5,
623           18,
624           0
625         ],
626         "type": "EXPENSE",
627         "currency": "USD",
628         "amount_total": 240.00000000,
629         "payer_id": 2,
630         "created_by": 1,
631         "note": "Groceries (test)"
632       },
633       {
634         "transaction_id": 112,
635         "txn_at": [
636           2025,
637           9,
638           11,
639           8,
640           0
641         ],
642         "type": "INCOME",
643         "currency": "USD",
644         "amount_total": 100.00000000,
645         "payer_id": 1,
646         "created_by": 1,
647         "note": "Refund (test)"
648       },
649       {
650         "transaction_id": 111,
651         "txn_at": [
652           2025,
653           9,
654           10,
655           8,
656           0
657         ],
658         "type": "EXPENSE",
659         "currency": "USD",
660         "amount_total": 150.00000000,
661         "payer_id": 1,
662         "created_by": 1,
663         "note": "Gas and tolls (test)"
664       }
665     ]
666   },
667   "total": null
668 }
669 <<< END
670
671 --- 3.6 Delete TXN2 and list again
672
673 >>> HTTP DELETE http://127.0.0.1:8081/api/v1/ledgers/3/transactions/112
674 >>> Header: X-Auth-Token: *****(@hidden)*****
675 >>> Request Body: (empty)

```

```

765 --- Request Body: (empty)
766 <-- HTTP Code: 204
767 <-- Response Body: (empty)
768 <-- END
769
770 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3/transactions?page=1&size=50
771 >>> Header: X-Auth-Token: *****(@hidden)*****
772 >>> Request Body: (empty)
773 <-- HTTP Code: 200
774 <-- Response Body:
775 {
776     "success": true,
777     "message": "OK",
778     "data": {
779         "page": 1,
780         "size": 50,
781         "total": 2,
782         "items": [
783             {
784                 "transaction_id": 113,
785                 "txns_at": [
786                     2025,
787                     10,
788                     5,
789                     18,
790                     0
791                 ],
792                 "type": "EXPENSE",
793                 "currency": "USD",
794                 "amount_total": 240.00000000,
795                 "payer_id": 2,
796                 "created_by": 1,
797                 "note": "Groceries (test)"
798             },
799             {
800                 "transaction_id": 111,
801                 "txns_at": [
802                     2025,
803                     9,
804                     10,
805                     8,
806                     0
807                 ],
808                 "type": "EXPENSE",
809                 "currency": "USD",
810                 "amount_total": 150.00000000,
811                 "payer_id": 1,
812                 "created_by": 1,
813                 "note": "Gas and tolls (test)"
814             }
815         ]
816     },
817     "total": null
818 }
819 <-- END
820
821 =====
822 => 4) SETTLEMENT PLAN API
823 =====
824
825 --- 4.1 /settlement-plan
826
827 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3/settlement-plan
828 >>> Header: X-Auth-Token: *****(@hidden)*****
829 >>> Request Body: (empty)
830 <-- HTTP Code: 200
831 <-- Response Body:
832 {
833     "success": true,
834     "message": "OK",
835     "data": {
836         "ledger_id": 3,
837         "currency": "USD",
838         "transfer_count": 2,
839         "transfers": [
840             {
841                 "from_user_id": 1,
842                 "from_user_name": "Alice",
843                 "to_user_id": 2,
844                 "to_user_name": "Bob",
845                 "amount": 5.00
846             },
847             {
848                 "from_user_id": 3,
849                 "from_user_name": "Charlie",
850                 "to_user_id": 2,
851                 "to_user_name": "Bob",
852                 "amount": 105.00
853             }
854         ]
855     },
856     "total": null
857 }
858 <-- END
859
860 =====
861 => 5) ANALYTICS API
862 =====
863
864

```

```

    //4 --- 3.1 /analytics/overview?months=3
775
776 >>> HTTP GET http://127.0.0.1:8081/api/v1/ledgers/3/analytics/overview?months=3
777 >>> Header: X-Auth-Token: *****(hidden)*****
778 >>> Request Body: (empty)
779 <<< HTTP Code: 200
780 <<< Response Body:
781 {
782     "success": true,
783     "message": "OK",
784     "data": {
785         "currency": "USD",
786         "range_start": [
787             2025,
788             10,
789             1,
790             0,
791             0
792         ],
793         "range_end": [
794             2026,
795             1,
796             1,
797             0,
798             0
799         ],
800         "total_income": 0E-8,
801         "total_expense": 80.00000000,
802         "net_balance": -80.00000000,
803         "trend": [
804             {
805                 "period": "2025-10",
806                 "income": 0E-8,
807                 "expense": 80.00000000
808             },
809             {
810                 "period": "2025-11",
811                 "income": 0,
812                 "expense": 0
813             },
814             {
815                 "period": "2025-12",
816                 "income": 0,
817                 "expense": 0
818             }
819         ],
820         "by_category": [
821             {
822                 "category_id": null,
823                 "category_name": "Uncategorized",
824                 "amount": 80.00000000,
825                 "ratio": 1.0000
826             }
827         ],
828         "arap": [
829             {
830                 "user_id": 2,
831                 "user_name": "Bob",
832                 "ar": 160.00000000,
833                 "ap": 50.00000000
834             },
835             {
836                 "user_id": 1,
837                 "user_name": "Alice",
838                 "ar": 75.00000000,
839                 "ap": 80.00000000
840             },
841             {
842                 "user_id": 3,
843                 "user_name": "Charlie",
844                 "ar": 0,
845                 "ap": 105.00000000
846             }
847         ],
848         "top_merchants": [
849             {
850                 "label": "Groceries (test)",
851                 "amount": 80.00000000
852             }
853         ],
854         "recommendations": [
855             {
856                 "code": "SPEND_TOO_HIGH",
857                 "message": "Expenses are greater than income in this period. Consider reducing spending.",
858                 "severity": "WARNING"
859             }
860         ]
861     },
862     "total": null
863 }
864 <<< END
865 Analytics check:
866     total_income = 0E-8
867     total_expense = 80.00000000
868     trend_len    = 3
869     rec_len      = 1
870
871 =====
872 => 6) LOGOUT
---
```

```

873 =====
874
875 --- 6.1 /auth/logout
876
877 >>> HTTP POST http://127.0.0.1:8081/api/v1/auth/logout?refreshToken=04kkDcSKr-j7BtGCj850kPm8Ych7oBAXSNuK-XMd7C4
878 >>> Header: X-Auth-Token: *****  
*****  

879 >>> Request Body: (empty)
880 <<< HTTP Code: 200
881 <<< Response Body:
882 {
883     "success": true,
884     "message": "OK",
885     "data": null,
886     "total": null
887 }
888 <<< END
889
890 [DONE] ALL API TESTS PASSED ✅
891 Cleaning up...
892 Stopping Spring Boot application (PID: 2957)...

```

▼ Run API tests (api_negative)

17s

```

1 ► Run cd API_test
14
15 =====
16 => 0) RESETTING DATABASE FOR NEGATIVE TESTS
17 =====
18 Dropping database ledger ...
19 Warning: arning! Using a password on the command line interface can be insecure.
20 Creating database ledger ...
21 Warning: arning! Using a password on the command line interface can be insecure.
22 Loading schema: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/ledger_flow.sql
23 Warning: arning! Using a password on the command line interface can be insecure.
24 Loading seed data: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/backup/ledger_big_seed.sql
25 Warning: arning! Using a password on the command line interface can be insecure.
26 [OK] Database initialized.
27 Starting Spring Boot application in background...
28 Working directory: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project
29 Profile: test
30 Spring Boot started with PID: 3854
31 Waiting for app to be ready...
32 Waiting... (2/120 seconds) - HTTP code: 000000
33 Waiting... (4/120 seconds) - HTTP code: 000000
34 Waiting... (6/120 seconds) - HTTP code: 000000
35 Waiting... (8/120 seconds) - HTTP code: 000000
36 Waiting... (10/120 seconds) - HTTP code: 000000
37 Waiting... (12/120 seconds) - HTTP code: 000000
38 [OK] Spring Boot app is ready!
39
40 =====
41 => 1) AUTH ENDPOINTS - EQUIVALENCE PARTITIONS AND BOUNDARIES
42 =====
43
44 --- [AUTH-LOGIN-VALID] Login with existing seed user
45
46 --- [AUTH-REG-INVALID-EMAIL] Register with invalid email (missing @)
47
48 --- [AUTH-REG-INVALID-PASS-EMPTY] Register with empty password
49
50 --- [AUTH-REG-DUPLICATE-EMAIL] Register duplicate email
51
52 --- [AUTH-LOGIN-VALID] Login with correct credentials
53
54 --- [AUTH-LOGIN-WRONG-PASSWORD] Login with wrong password
55
56 --- [AUTH-LOGIN-NONEXISTENT-USER] Login unknown email
57
58 =====
59 => 2) USERS ENDPOINTS - AUTHORIZATION AND MULTIPLE CLIENTS
60 =====
61
62 --- [USERS-ME-WITH-TOKEN] /users/me with valid token
63
64 --- [USERS-ME-NO-TOKEN] /users/me without token must fail
65
66 --- [AUTH-REG-BOB] Register Bob as second client
67
68 --- [AUTH-LOGIN-BOB] Login Bob
69
70 --- [BOB-ME] /users/me for Bob
71
72 =====
73 => 3) LEDGER ENDPOINTS - VALID/INVALID INPUTS AND MEMBERSHIP
74 =====
75
76 --- [LEDGER-CREATE-VALID] Create ledger with valid payload
77
78 --- [LEDGER-CREATE-MISSING-NAME] Create ledger with missing name
79
80 --- [LEDGER-CREATE-INVALID-TYPE] Create ledger with invalid ledger_type
81
82 --- [LEDGER-CREATE-INVALID-CURRENCY] Create ledger with unknown base_currency
83
84 --- [LEDGER-GET-AS-OWNER] /ledgers/{id} with owner token
85
86 --- [LEDGER-GET-AS-NON-MEMBER] /ledgers/{id} as non-member should fail
87
88 --- [LEDGER-ADD-MEMBER-VALID] Add Bob to ledger

```

```

89
90 --- [LEDGER-ADD-MEMBER-DUPLICATE] Add Bob again should fail
91
92 --- [LEDGER-REMOVE-NON-MEMBER] Remove non-member ID should fail
93
94 --- [LEDGER-GET-AS-MEMBER] /ledgers/{id} as added member should now succeed
95
96 =====
97 => 4) TRANSACTION ENDPOINTS - INVALID AND BOUNDARY PAYLOADS
98 =====
99
100 --- [TXN-CREATE-VALID] Create a valid EXPENSE transaction
101
102 --- [TXN-NEGATIVE-AMOUNT] amount_total < 0 should fail
103
104 --- [TXN-ZERO-AMOUNT] amount_total == 0 boundary should fail or be rejected
105
106 --- [TXN-EMPTY-SPLITS] Empty splits array should fail
107
108 --- [TXN-SUM-MISMATCH] Splits do not sum to amount_total
109
110 --- [TXN-INVALID-TYPE] Unknown transaction type must fail
111
112 --- [TXN-AS-NON-MEMBER] Non-member cannot create transaction on ledger
113
114 =====
115 => 5) SETTLEMENT PLAN AND ANALYTICS - MEMBERSHIP + BOUNDARIES
116 =====
117
118 --- [SETTLEMENT-VALID-AS-MEMBER] /settlement-plan for member should succeed
119
120 --- [SETTLEMENT-AS-NON-MEMBER] Non-member cannot view settlement plan
121
122 --- [ANALYTICS-OVERVIEW-VALID] Basic analytics overview for member
123
124 --- [ANALYTICS-MONTHS-ZERO] months=0 should clamp to default (3 months)
125
126 --- [ANALYTICS-MONTHS-ABOVE-MAX] months>24 should clamp to 24
127
128 --- [ANALYTICS-AS-NON-MEMBER] Non-member cannot access analytics
129
130 --- [ANALYTICS-UNKNOWN-LEDGER] Analytics on non-existent ledger must fail
131
132 [DONE] Basic negative API tests completed
133 Cleaning up...
134 Stopping Spring Boot application (PID: 3854)...

```

▼  Run API tests (api_budget_tests)

15s

```

1 ► Run cd API_test
14
15 =====
16 => 0) SETUP: Database and Test Data
17 =====
18 Resetting database...
19 Warning: arning] Using a password on the command line interface can be insecure.
20 Warning: arning] Using a password on the command line interface can be insecure.
21 Loading schema: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/ledger_flow.sql
22 Warning: arning] Using a password on the command line interface can be insecure.
23 Loading seed data: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/backup/ledger_big_seed.sql
24 Warning: arning] Using a password on the command line interface can be insecure.
25 [OK] Database initialized.
26 Starting Spring Boot application in background...
27 Working directory: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project
28 Profile: test
29 Spring Boot started with PID: 4692
30 Waiting for app to be ready...
31 Waiting... (2/120 seconds) - HTTP code: 000000
32 Waiting... (4/120 seconds) - HTTP code: 000000
33 Waiting... (6/120 seconds) - HTTP code: 000000
34 Waiting... (8/120 seconds) - HTTP code: 000000
35 Waiting... (10/120 seconds) - HTTP code: 000000
36 Waiting... (12/120 seconds) - HTTP code: 000000
37 [OK] Spring Boot app is ready!
38
39 --- Login Alice (OWNER)
40
41 --- Get Alice ID
42
43 --- Login Bob (will be EDITOR)
44
45 --- Get Bob ID
46
47 --- Create test ledger
48
49 --- Add Bob as EDITOR
50
51 --- Create category for budget tests
52 Warning: arning] Using a password on the command line interface can be insecure.
53 [OK] Test setup complete
54 LEDGER_ID=3
55 CATEGORY_ID=11
56 ALICE_ID=1 (OWNER)
57 BOB_ID=2 (EDITOR)
58
59 =====
60 => 1) POST /budgets - VALID EQUIVALENCE PARTITIONS
61 =====
62

```

```

63 --- [BUDGET-CREATE-VALID] Create new budget (typical valid)
64 [PASS] Budget created successfully
65
66 --- [BUDGET-UPDATE-VALID] Update existing budget (same ledger/category/year/month)
67 [PASS] Budget updated successfully
68
69 --- [BUDGET-CREATE-DIFFERENT-MONTH] Create budget for different month
70 [PASS] Budget for different month created
71
72 --- [BUDGET-CREATE-NULL-CATEGORY] Create ledger-wide budget (null categoryId)
73 [PASS] Ledger-wide budget created
74
75 =====
76 => 2) POST /budgets - INVALID EQUIVALENCE PARTITIONS & BOUNDARIES
77 =====
78
79 --- [BUDGET-NEGATIVE-AMOUNT] Negative amount (boundary: limitAmount < 0)
80 [PASS] Negative amount properly rejected
81
82 --- [BUDGET-ZERO-AMOUNT] Zero amount (boundary: limitAmount == 0)
83 [PASS] Zero amount properly rejected
84
85 --- [BUDGET-INVALID-MONTH-ZERO] Invalid month: 0 (boundary: month < 1)
86 [PASS] Month=0 properly rejected
87
88 --- [BUDGET-INVALID-MONTH-13] Invalid month: 13 (boundary: month > 12)
89 [PASS] Month=13 properly rejected
90
91 --- [BUDGET-INVALID-YEAR-LOW] Invalid year: 2019 (boundary: year < 2020)
92 [PASS] Year=2019 properly rejected
93
94 --- [BUDGET-INVALID-YEAR-HIGH] Invalid year: 2101 (boundary: year > 2100)
95 [PASS] Year=2101 properly rejected
96
97 --- [BUDGET-MISSING-AMOUNT] Missing required field: limitAmount
98 [PASS] Missing amount properly rejected
99
100 --- [BUDGET-EDITOR-ROLE] Non-OWNER/ADMIN role (Bob is EDITOR)
101 [PASS] EDITOR role properly rejected
102
103 --- [BUDGET-NO-AUTH] No authentication token
104 [PASS] No authentication properly rejected
105
106 --- [BUDGET-NONEXISTENT-LEDGER] Non-existent ledger ID
107 [PASS] Non-existent ledger properly rejected
108
109 =====
110 => 3) GET /budgets/status - VALID EQUIVALENCE PARTITIONS
111 =====
112
113 --- [BUDGET-STATUS-WITH-DATA] Get budget status with existing budgets
114 [PASS] Budget status retrieved (found 0 budgets)
115
116 --- [BUDGET-STATUS-EMPTY] Get budget status with no budgets (boundary)
117 [PASS] Empty budget list handled correctly
118
119 --- [BUDGET-STATUS-AS-MEMBER] Get budget status as member (Bob)
120 [PASS] Member can view budget status
121
122 =====
123 => 4) GET /budgets/status - INVALID EQUIVALENCE PARTITIONS
124 =====
125
126 --- [BUDGET-STATUS-INVALID-MONTH-ZERO] Invalid month: 0 (returns empty, not error)
127 [PASS] Month=0 handled (returns empty result)
128
129 --- [BUDGET-STATUS-INVALID-MONTH-13] Invalid month: 13 (returns empty, not error)
130 [PASS] Month=13 handled (returns empty result)
131
132 --- [BUDGET-STATUS-NO-AUTH] No authentication token
133 [PASS] No authentication properly rejected
134
135 --- [BUDGET-STATUS-NONEXISTENT-LEDGER] Non-existent ledger ID
136 [PASS] Non-existent ledger properly rejected
137
138 --- Create non-member user (Charlie)
139
140 --- [BUDGET-STATUS-NON-MEMBER] Non-member access
141 [PASS] Non-member access properly rejected
142
143 =====
144 => 5) TEST SUMMARY
145 =====
146
147 All Budget API tests completed successfully!
148
149 Equivalence Partitions Tested:
150 POST /budgets:
151   - Valid: 4 partitions (create, update, different month, null category)
152   - Invalid: 11 partitions (negative, zero, month boundaries, year boundaries, missing fields, auth)
153   GET /budgets/status:
154   - Valid: 3 partitions (with data, empty, as member)
155   - Invalid: 10 partitions (month boundaries, year boundaries, missing params, auth, non-member)
156
157 Stopping Spring Boot application (PID: 4692)...

```

Run API tests (api_settlement_execution_tests)

1 ► Run cd API_test

```

14
15 =====
16 => 0) SETUP: Database and Test Data
17 =====
18 Resetting database...
19 Warning: arning] Using a password on the command line interface can be insecure.
20 Warning: arning] Using a password on the command line interface can be insecure.
21 Loading schema: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/ledger_flow.sql
22 Warning: arning] Using a password on the command line interface can be insecure.
23 Loading seed data: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/backup/ledger_big_seed.sql
24 Warning: arning] Using a password on the command line interface can be insecure.
25 [OK] Database initialized.
26 Starting Spring Boot application in background...
27 Working directory: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project
28 Profile: test
29 Spring Boot started with PID: 5456
30 Waiting for app to be ready...
31 Waiting... (2/120 seconds) - HTTP code: 000000
32 Waiting... (4/120 seconds) - HTTP code: 000000
33 Waiting... (6/120 seconds) - HTTP code: 000000
34 Waiting... (8/120 seconds) - HTTP code: 000000
35 Waiting... (10/120 seconds) - HTTP code: 000000
36 Waiting... (12/120 seconds) - HTTP code: 000000
37 [OK] Spring Boot app is ready!
38
39 --- Login Alice (OWNER)
40
41 --- Get Alice ID
42
43 --- Login Bob (will be EDITOR)
44
45 --- Get Bob ID
46
47 --- Login Charlie (will be VIEWER)
48
49 --- Get Charlie ID
50
51 --- Create test ledger
52
53 --- Create category for transactions
54 Warning: arning] Using a password on the command line interface can be insecure.
55
56 --- Add Bob as EDITOR
57
58 --- Add Charlie as VIEWER
59
60 --- Create transactions to generate debts
61 [OK] Test setup complete
62 LEDGER_ID=3
63 ALICE_ID=1 (OWNER)
64 BOB_ID=2 (EDITOR)
65 CHARLIE_ID=3 (VIEWER)
66
67 =====
68 => 1) POST /settlement-plan - VALID EQUIVALENCE PARTITIONS
69 =====
70
71 --- [SETTLEMENT-EXECUTE-AS-OWNER] Execute settlement as OWNER (typical valid)
72 [PASS] Settlement executed successfully as OWNER
73
74 --- Verify settlement was executed (check transactions)
75 [PASS] Transactions retrieved after settlement
76
77 --- [SETTLEMENT-EXECUTE-AS-EDITOR] Execute settlement as EDITOR (valid)
78 [PASS] Settlement executed successfully as EDITOR
79
80 =====
81 => 2) POST /settlement-plan - VALID (VIEWER can also view plan)
82 =====
83
84 --- [SETTLEMENT-VIEW-AS-VIEWER] View settlement plan as VIEWER (valid)
85 [PASS] VIEWER can view settlement plan
86
87 =====
88 => 3) POST /settlement-plan - INVALID EQUIVALENCE PARTITIONS
89 =====
90
91 --- [SETTLEMENT-NO-AUTH] Execute settlement without authentication
92 [PASS] No authentication properly rejected
93
94 --- [SETTLEMENT-NONEXISTENT-LEDGER] Execute settlement on non-existent ledger
95 [PASS] Non-existent ledger properly rejected
96
97 --- Create non-member user
98
99 --- [SETTLEMENT-NON-MEMBER] Execute settlement as non-member
100 [PASS] Non-member properly rejected from executing settlement
101
102 =====
103 => 4) TEST SUMMARY
104 =====
105
106 All Settlement Plan Execution API tests completed successfully!
107
108 Equivalence Partitions Tested:
109   POST /settlement-plan:
110     - Valid: 3 partitions (OWNER, EDITOR, VIEWER - all members can view plan)
111     - Invalid: 3 partitions (no auth, non-existent ledger, non-member)
112

```

```

113 Stopping Spring Boot application (PID: 5456)...
14
15 =====
16 => 0. RESETTING DATABASE TO CLEAN STATE
17 =====
18 Dropping existing database...
19 Creating new database...
20 Warning: arning] Using a password on the command line interface can be insecure.
21 Loading schema...
22 Warning: arning] Using a password on the command line interface can be insecure.
23 Loading seed data...
24 Warning: arning] Using a password on the command line interface can be insecure.
25 Database reset successfully!
26 Starting Spring Boot application in background...
27 Working directory: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project
28 Profile: test
29 Spring Boot started with PID: 5997
30 Waiting for app to be ready...
31 Waiting... (2/120 seconds) - HTTP code: 000000
32 Waiting... (4/120 seconds) - HTTP code: 000000
33 Waiting... (6/120 seconds) - HTTP code: 000000
34 Waiting... (8/120 seconds) - HTTP code: 000000
35 Waiting... (10/120 seconds) - HTTP code: 000000
36 Waiting... (12/120 seconds) - HTTP code: 000000
37 [OK] Spring Boot app is ready!
38
39 =====
40 => 1. USER REGISTRATION API TESTS
41 =====
42
43 --- 1.1 Typical Valid: Complete valid registration
44 {
45     "success": true,
46     "message": "OK",
47     "data": null,
48     "total": null
49 }
50 Typical registration works
51
52 --- 1.2 Atypical Valid: Minimum valid data (short name)
53 {
54     "success": true,
55     "message": "OK",
56     "data": null,
57     "total": null
58 }
59 Edge case registration works
60
61 --- 1.3 Invalid: Missing email field
62 {
63     "success": false,
64     "message": "Invalid request: Validation failed for argument [0] in public dev.coms4156.project.groupproject.dto.Result<java.lang.Void> dev.coms4156.project.groupproject.controller.UserV1Controller.register(dev.coms4156.project.groupproject.dto.RegisterRequest): [Field error in object 'registerRequest' on field 'email': rejected value [null]; codes [NotBlank,registerRequest.email,NotBlank.email,NotBlank.java.lang.String,NotBlank]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [registerRequest.email,email]; arguments []; default message [email]]; default message [must not be blank]]",
65     "data": null,
66     "total": null
67 }
68 Invalid registration properly rejected
69
70 --- 1.4 Invalid: Invalid email format
71 {
72     "success": false,
73     "message": "Invalid request: Validation failed for argument [0] in public dev.coms4156.project.groupproject.dto.Result<java.lang.Void> dev.coms4156.project.groupproject.controller.UserV1Controller.register(dev.coms4156.project.groupproject.dto.RegisterRequest): [Field error in object 'registerRequest' on field 'email': rejected value [invalid-email-format]; codes [Email,registerRequest.email,Email.email,Email.java.lang.String,Email]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [registerRequest.email,email]; arguments []; default message [email],[Ljakarta.validation.constraints.Pattern$Flag;@2afff2a5,.*]; default message [must be a well-formed email address]]",
74     "data": null,
75     "total": null
76 }
77 Invalid email properly rejected
78 All registration tests completed!
79
80 =====
81 => 2. USER LOGIN API TESTS
82 =====
83
84 --- 2.1 Typical Valid: Login with registered user
85 {
86     "success": true,
87     "message": "OK",
88     "data": {
89         "access_token": "fUyha8qgDlu9Aq_cR5yU3CMizGBCKYSTcbB0XdfLFJE",
90         "refresh_token": "24wifcu3gL55AfmsxeYWkco7tJC7mikYZCars-CDgii"
91     },
92     "total": null
93 }
94 User1 login successful
95
96 --- 2.2 Atypical Valid: Login with different user
97

```

```

98     "success": true,
99     "message": "OK",
100    "data": {
101      "access_token": "e7UctAjRyps1qAoftN7Hl00wU1Zj9F1yvn9mPVZYek",
102      "refresh_token": "GyFvms_XiTc1fJi9H8z6j0f2EjyxQfnSn-pRWjSRURM"
103    },
104    "total": null
105  }
106  User2 login successful
107
108  --- 2.3 Invalid: Wrong password
109  {
110    "success": false,
111    "message": "Wrong credentials",
112    "data": null,
113    "total": null
114  }
115  Invalid password properly rejected
116
117  --- 2.4 Invalid: Non-existent user
118  {
119    "success": false,
120    "message": "User not found",
121    "data": null,
122    "total": null
123  }
124  Non-existent user properly rejected
125  All login tests completed!
126
127 =====
128 => 3. TOKEN REFRESH API TESTS
129 =====
130
131  --- 3.1 Typical Valid: Refresh with valid refresh token
132  {
133    "success": true,
134    "message": "OK",
135    "data": {
136      "access_token": "QqazAlcqByi0w0wG9H5hyZPXG4biQ20wi077oEq_EDQ",
137      "refresh_token": "5Ty5uJEyjpuLjWR69WuHs8Rs83_LBd535SiUNsteEYI"
138    },
139    "total": null
140  }
141  Token refresh successful
142
143  --- 3.2 Atypical Valid: Refresh multiple times
144  {
145    "success": true,
146    "message": "OK",
147    "data": {
148      "access_token": "N7zUKnsgDhi_39AUQYqudMFKF675EYeRICElz6JkH28",
149      "refresh_token": "JER-nFfcj2RnTlb35TvxZ7lKcL0Yp2KEuE_g0ENh2_s"
150    },
151    "total": null
152  }
153  Multiple refresh handled
154
155  --- 3.3 Invalid: Refresh with invalid token
156  {
157    "success": false,
158    "message": "Invalid or expired refresh token",
159    "data": null,
160    "total": null
161  }
162  Invalid refresh token properly rejected
163  All token refresh tests completed!
164
165 =====
166 => 4. USER LOGOUT API TESTS
167 =====
168
169  --- 4.1 Typical Valid: Logout with valid refresh token
170  {
171    "success": false,
172    "message": "AUTH_REQUIRED",
173    "data": null,
174    "total": null
175  }
176  Logout successful
177
178  --- 4.2 Atypical Valid: Logout with empty/null token (should be no-op)
179  {
180    "success": false,
181    "message": "AUTH_REQUIRED",
182    "data": null,
183    "total": null
184  }
185  Empty token logout handled
186
187  --- 4.3 Invalid: Logout with invalid token
188  {
189    "success": false,
190    "message": "AUTH_REQUIRED",
191    "data": null,
192    "total": null
193  }
194  Invalid logout token handled
195  All logout tests completed!
196

```

```

196
197 =====
198 => 5. USER LOOKUP API TESTS
199 =====
200
201 --- 5.1 Typical Valid: Get current user info
202 {
203   "success": true,
204   "message": "OK",
205   "data": {
206     "id": 8,
207     "name": "Alex Chen"
208   },
209   "total": null
210 }
211 User1 ID retrieved: 8
212
213 --- 5.2 Typical Valid: Lookup user by email
214 {
215   "success": true,
216   "message": "OK",
217   "data": {
218     "user_id": 10,
219     "name": "Sarah Johnson"
220   },
221   "total": null
222 }
223 User2 ID retrieved: 10
224
225 --- 5.3 Atypical Valid: Lookup with special characters in email (NEW)
226 {
227   "success": true,
228   "message": "OK",
229   "data": {
230     "user_id": 9,
231     "name": "AI"
232   },
233   "total": null
234 }
235 Special character email lookup handled
236
237 --- 5.4 Invalid: Lookup non-existent user
238 {
239   "success": false,
240   "message": "USER_NOT_FOUND",
241   "data": null,
242   "total": null
243 }
244 Non-existent user lookup properly rejected
245
246 --- 5.5 Invalid: Lookup without auth token (NEW)
247 {
248   "success": false,
249   "message": "AUTH_REQUIRED",
250   "data": null,
251   "total": null
252 }
253 Unauthorized lookup properly rejected
254 All user lookup tests completed!
255
256 =====
257 => 6. USER PROFILE API TESTS
258 =====
259
260 --- 6.1 Typical Valid: Get own profile by ID
261 {
262   "success": true,
263   "message": "OK",
264   "data": {
265     "id": 8,
266     "name": "Alex Chen"
267   },
268   "total": null
269 }
270 Own profile retrieved successfully
271
272 --- 6.2 Atypical Valid: Get another user's profile
273 {
274   "success": true,
275   "message": "OK",
276   "data": {
277     "id": 10,
278     "name": "Sarah Johnson"
279   },
280   "total": null
281 }
282 Other user profile retrieved successfully
283
284 --- 6.3 Invalid: Get profile without authentication
285 {
286   "success": false,
287   "message": "AUTH_REQUIRED",
288   "data": null,
289   "total": null
290 }
291 Unauthorized profile access properly rejected
292 All user profile tests completed!
293
294 =====

```

```

295 => 7. LEDGER CREATION API TESTS
296 =====
297
298 --- 7.1 Typical Valid: Create standard group balance ledger
299 {
300     "success": true,
301     "message": "OK",
302     "data": {
303         "ledger_id": 7,
304         "name": "Tech Team Road Trip 2025",
305         "ledger_type": "GROUP_BALANCE",
306         "base_currency": "USD",
307         "share_start_date": null,
308         "role": "OWNER",
309         "categories": [
310             {
311                 "id": 3,
312                 "ledger_id": 7,
313                 "name": "Default",
314                 "kind": "EXPENSE",
315                 "is_active": true
316             }
317         ],
318     },
319     "total": null
320 }
321 Main ledger created with ID: 7
322
323 --- 7.2 Atypical Valid: Create ledger with start date
324 {
325     "success": true,
326     "message": "OK",
327     "data": {
328         "ledger_id": 8,
329         "name": "Family Vacation Fund 2025",
330         "ledger_type": "GROUP_BALANCE",
331         "base_currency": "USD",
332         "share_start_date": [
333             2025,
334             1,
335             1
336         ],
337         "role": "OWNER",
338         "categories": [
339             {
340                 "id": 4,
341                 "ledger_id": 8,
342                 "name": "Default",
343                 "kind": "EXPENSE",
344                 "is_active": true
345             }
346         ],
347     },
348     "total": null
349 }
350 Secondary ledger created with ID: 8
351
352 --- 7.3 Invalid: Create ledger without auth token
353 {
354     "success": false,
355     "message": "AUTH_REQUIRED",
356     "data": null,
357     "total": null
358 }
359 Unauthorized ledger creation properly rejected
360
361 --- 7.4 Invalid: Create ledger with invalid type
362 {
363     "success": false,
364     "message": "\n## Error updating database. Cause: java.sql.SQLException: Data truncated for column 'ledger_type' at row 1\n## The error may exist in dev/coms4156/project/groupproject/mapper/LedgerMapper.java (best guess)\n## The error may involve dev.coms4156.project.groupproject.mapper.LedgerMapper.insert-Inline\n## The error occurred while setting parameters\n## SQL: INSERT INTO ledgers ( name, owner_id, ledger_type, base_currency ) VALUES ( ?, ?, ?, ? )\n## Cause: java.sql.SQLException: Data truncated for column 'ledger_type' at row 1",
365     "data": null,
366     "total": null
367 }
368 Invalid ledger type properly rejected
369
370 --- 7.5 Invalid: Create ledger with empty name
371 {
372     "success": false,
373     "message": "Invalid request: Validation failed for argument [0] in public dev.coms4156.project.groupproject.dto.Result<dev.coms4156.project.groupproject.dto.LedgerResponse> dev.coms4156.project.groupproject.controller.LedgerController.createLedger(dev.coms4156.project.groupproject.dto.CreateLedgerRequest): [Field error in object 'createLedgerRequest' on field 'name': rejected value []; codes [NotBlank.createLedgerRequest.name,NotBlank.name,NotBlank.java.lang.String,NotBlank]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [createLedgerRequest.name,name]; arguments []; default message [name]]; default message [must not be blank]]",
374     "data": null,
375     "total": null
376 }
377 Empty name properly rejected
378 All ledger creation tests completed!
379
380 =====
381 => 8. MY LEDGERS API TESTS
382 =====

```

```

384 --- 8.1 Typical Valid: Get ledgers for user with ledgers
385 {
386   "success": true,
387   "message": "OK",
388   "data": {
389     "items": [
390       {
391         "ledger_id": 7,
392         "name": "Tech Team Road Trip 2025",
393         "ledger_type": "GROUP_BALANCE",
394         "base_currency": "USD",
395         "role": "OWNER"
396       },
397       {
398         "ledger_id": 8,
399         "name": "Family Vacation Fund 2025",
400         "ledger_type": "GROUP_BALANCE",
401         "base_currency": "USD",
402         "role": "OWNER"
403       }
404     ]
405   },
406   "total": null
407 }
408 User1 ledgers retrieved successfully
409
410 --- 8.2 Atypical Valid: Get ledgers for user with no ledgers
411 {
412   "success": true,
413   "message": "OK",
414   "data": {
415     "items": []
416   },
417   "total": null
418 }
419 Empty ledgers list handled correctly
420
421 --- 8.3 Invalid: Get ledgers without authentication
422 {
423   "success": false,
424   "message": "AUTH_REQUIRED",
425   "data": null,
426   "total": null
427 }
428 Unauthorized access properly rejected
429 All my ledgers tests completed!
430
431 =====
432 => 9. LEDGER DETAILS API TESTS
433 =====
434
435 --- 9.1 Typical Valid: Get ledger details as owner
436 {
437   "success": true,
438   "message": "OK",
439   "data": {
440     "ledger_id": 7,
441     "name": "Tech Team Road Trip 2025",
442     "ledger_type": "GROUP_BALANCE",
443     "base_currency": "USD",
444     "share_start_date": null,
445     "role": "OWNER",
446     "categories": [
447       {
448         "id": 3,
449         "ledger_id": 7,
450         "name": "Default",
451         "kind": "EXPENSE",
452         "is_active": true
453       }
454     ]
455   },
456   "total": null
457 }
458 Ledger details retrieved successfully
459
460 --- 9.2 Invalid: Get ledger details as non-member
461 {
462   "success": false,
463   "message": "FORBIDDEN: You are not a member of this ledger",
464   "data": null,
465   "total": null
466 }
467 Non-member access properly rejected
468
469 --- 9.3 Invalid: Get non-existent ledger
470 {
471   "success": false,
472   "message": "LEDGER_NOT_FOUND",
473   "data": null,
474   "total": null
475 }
476 Non-existent ledger properly rejected
477 All ledger details tests completed!
478
479 =====
480 => 10. LEDGER MEMBER MANAGEMENT API TESTS
481 =====
482

```

```

483 --- 10.1 Typical Valid: Add member as editor
484 {
485     "success": true,
486     "message": "OK",
487     "data": {
488         "ledger_id": 7,
489         "user_id": 10,
490         "role": "EDITOR"
491     },
492     "total": null
493 }
494 User2 added as editor
495
496 --- 10.2 Atypical Valid: Add member with different role - Create USER3 first
497 {
498     "success": true,
499     "message": "OK",
500     "data": {
501         "ledger_id": 7,
502         "user_id": 11,
503         "role": "VIEWER"
504     },
505     "total": null
506 }
507 User3 added as viewer
508
509 --- 10.3 Invalid: Add non-existent user
510 {
511     "success": false,
512     "message": "n## Error updating database. Cause: java.sql.IntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails (`ledger`.`ledger_members`, CONSTRAINT `fk_members_user` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE)n## The error may exist in dev/coms4156/project/groupproject/mapper/LedgerMemberMapper.java (best guess)n## The error may involve dev.coms4156.project.groupproject.mapper.LedgerMemberMapper.insert-Inlinen## The error occurred while setting parametersn## SQL: INSERT INTO ledger_members ( ledger_id, user_id, role ) VALUES ( ?, ?, ? )n## Cause: java.sql.IntegrityConstraintViolationException: Cannot add or update a child row: a foreign key constraint fails (`ledger`.`ledger_members`, CONSTRAINT `fk_members_user` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE)n; Cannot add or update a child row: a foreign key constraint fails (`ledger`.`ledger_members`, CONSTRAINT `fk_members_user` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE)",
513     "data": null,
514     "total": null
515 }
516 Non-existent user addition properly rejected
517
518 --- 10.4 Invalid: Add member without proper permissions
519 {
520     "success": false,
521     "message": "ROLE_INSUFFICIENT: You do not have the required role",
522     "data": null,
523     "total": null
524 }
525 Unauthorized member addition properly rejected
526
527 --- 10.5 Typical Valid: List ledger members (ENHANCED)
528 {
529     "success": true,
530     "message": "OK",
531     "data": {
532         "items": [
533             {
534                 "user_id": 8,
535                 "name": "Alex Chen",
536                 "role": "OWNER"
537             },
538             {
539                 "user_id": 10,
540                 "name": "Sarah Johnson",
541                 "role": "EDITOR"
542             },
543             {
544                 "user_id": 11,
545                 "name": "Mike Wilson",
546                 "role": "VIEWER"
547             }
548         ],
549     },
550     "total": null
551 }
552 Member list retrieved successfully
553
554 --- 10.6 Invalid: List members as non-member (NEW)
555 {
556     "success": true,
557     "message": "OK",
558     "data": {
559         "items": [
560             {
561                 "user_id": 8,
562                 "name": "Alex Chen",
563                 "role": "OWNER"
564             },
565             {
566                 "user_id": 10,
567                 "name": "Sarah Johnson",
568                 "role": "EDITOR"
569             },
570             {
571                 "user_id": 11,
572                 "name": "Mike Wilson",
573                 "role": "VIEWER"
574             }

```

```

575     ]
576   },
577   "total": null
578 }
579 Non-member list access properly rejected
580 All member management tests completed!
581
582 =====
583 => 11. REMOVE MEMBER API TESTS
584 =====
585
586 --- 11.1 Typical Valid: Owner removes a member
587 Member removed successfully (USER3)
588
589 --- 11.2 Invalid: Editor tries to remove a member
590 {
591   "success": false,
592   "message": "ROLE_INSUFFICIENT: You do not have the required role",
593   "data": null,
594   "total": null
595 }
596 Unauthorized removal properly rejected
597
598 --- 11.3 Invalid: Remove non-existent member
599 Non-existent member removal properly handled
600 All remove member tests completed!
601
602 =====
603 => 12. TRANSACTION CREATION API TESTS
604 =====
605
606 --- 12.1 Typical Valid: Create simple expense transaction
607 {
608   "success": true,
609   "message": "OK",
610   "data": {
611     "transaction_id": 4,
612     "budget_alert": null
613   },
614   "total": null
615 }
616 Transaction created with ID: 4
617
618 --- 12.2 Atypical Valid: Create income transaction with percentage split
619 {
620   "success": true,
621   "message": "OK",
622   "data": {
623     "transaction_id": 5,
624     "budget_alert": null
625   },
626   "total": null
627 }
628 Income transaction created
629
630 --- 12.3 Invalid: Create transaction with negative amount
631 {
632   "success": false,
633   "message": "Invalid request: Validation failed for argument [1] in public
dev.coms4156.project.groupproject.dto.Result<dev.coms4156.project.groupproject.dto.CreateTransactionResponse>
dev.coms4156.project.groupproject.controller.TransactionController.createTransaction(java.lang.Long,dev.coms4156.project.groupproject.dto.CreateTransactionRequest): [Field error in object 'createTransactionRequest' on field 'amountTotal': rejected value [-50.00]; codes
[DecimalMin.createTransactionRequest.amountTotal,DecimalMin.amountTotal,DecimalMin.java.math.BigDecimal,DecimalMin]; arguments
[org.springframework.context.support.DefaultMessageSourceResolvable: codes [createTransactionRequest.amountTotal,amountTotal]; arguments []
; default message [amountTotal],true,0.01]; default message [Amount must be positive]] ",
634   "data": null,
635   "total": null
636 }
637 Negative amount properly rejected
638
639 --- 12.4 Invalid: Create transaction with non-member in split
640 {
641   "success": true,
642   "message": "OK",
643   "data": {
644     "transaction_id": 6,
645     "budget_alert": null
646   },
647   "total": null
648 }
649 Non-member in split properly rejected
650
651 --- 12.5 Invalid: Create transaction without auth token
652 {
653   "success": false,
654   "message": "AUTH_REQUIRED",
655   "data": null,
656   "total": null
657 }
658 Unauthorized transaction creation properly rejected
659 All transaction creation tests completed!
660
661 =====
662 => 13. TRANSACTION DETAILS API TESTS
663 =====
664
665 --- 13.1 Typical Valid: Get transaction details as member
666 {
667   "success": true.

```

```

...
668     "message": "OK",
669     "data": {
670       "transaction_id": 4,
671       "ledger_id": 7,
672       "txn_at": [
673         2025,
674         1,
675         15,
676         12,
677         0
678     ],
679     "type": "EXPENSE",
680     "currency": "USD",
681     "amount_total": 120.5000000,
682     "category_id": null,
683     "note": "Team lunch at restaurant",
684     "payer_id": 8,
685     "created_by": 8,
686     "rounding_strategy": "ROUND_HALF_UP",
687     "tail_allocation": "PAYER",
688     "splits": [
689       {
690         "user_id": 8,
691         "split_method": "EQUAL",
692         "share_value": 0E-8,
693         "included": true,
694         "computed_amount": 60.2500000
695       },
696       {
697         "user_id": 10,
698         "split_method": "EQUAL",
699         "share_value": 0E-8,
700         "included": true,
701         "computed_amount": 60.2500000
702       }
703     ],
704     "edges_preview": [
705       {
706         "from_user_id": 8,
707         "to_user_id": 10,
708         "amount": 60.2500000,
709         "edge_currency": "USD"
710       }
711     ]
712   },
713   "total": null
714 }
715 Transaction details retrieved successfully
716
717 --- 13.2 Invalid: Get transaction details as non-member
718 {
719   "success": true,
720   "message": "OK",
721   "data": {
722     "transaction_id": 4,
723     "ledger_id": 7,
724     "txn_at": [
725       2025,
726       1,
727       15,
728       12,
729       0
730     ],
731     "type": "EXPENSE",
732     "currency": "USD",
733     "amount_total": 120.5000000,
734     "category_id": null,
735     "note": "Team lunch at restaurant",
736     "payer_id": 8,
737     "created_by": 8,
738     "rounding_strategy": "ROUND_HALF_UP",
739     "tail_allocation": "PAYER",
740     "splits": [
741       {
742         "user_id": 8,
743         "split_method": "EQUAL",
744         "share_value": 0E-8,
745         "included": true,
746         "computed_amount": 60.2500000
747       },
748       {
749         "user_id": 10,
750         "split_method": "EQUAL",
751         "share_value": 0E-8,
752         "included": true,
753         "computed_amount": 60.2500000
754       }
755     ],
756     "edges_preview": [
757       {
758         "from_user_id": 8,
759         "to_user_id": 10,
760         "amount": 60.2500000,
761         "edge_currency": "USD"
762       }
763     ]
764   },
765   "total": null
766

```

```

765  Non-member access properly rejected
766
767  --- 13.3 Invalid: Get non-existent transaction
768
769  {
770    "success": false,
771    "message": "Transaction not found",
772    "data": null,
773    "total": null
774  }
775  Non-existent transaction properly rejected
776  All transaction details tests completed!
777
778 =====
779 => 14. TRANSACTION QUERY API TESTS
780 =====
781
782 --- 14.1 Typical Valid: List all transactions in ledger
783
784  {
785    "success": true,
786    "message": "OK",
787    "data": {
788      "page": 1,
789      "size": 50,
790      "total": 3,
791      "items": [
792        {
793          "transaction_id": 6,
794          "txn_at": [
795            2025,
796            1,
797            26,
798            11,
799            0
800          ],
801          "type": "EXPENSE",
802          "currency": "USD",
803          "amount_total": 75.00000000,
804          "payer_id": 8,
805          "created_by": 8,
806          "note": "Split with non-member"
807        },
808        {
809          "transaction_id": 5,
810          "txn_at": [
811            2025,
812            1,
813            20,
814            9,
815            0
816          ],
817          "type": "INCOME",
818          "currency": "USD",
819          "amount_total": 200.00000000,
820          "payer_id": 8,
821          "created_by": 8,
822          "note": "Shared project payment received"
823        },
824        {
825          "transaction_id": 4,
826          "txn_at": [
827            2025,
828            1,
829            15,
830            12,
831            0
832          ],
833          "type": "EXPENSE",
834          "currency": "USD",
835          "amount_total": 120.50000000,
836          "payer_id": 8,
837          "created_by": 8,
838          "note": "Team lunch at restaurant"
839        }
840      ],
841    },
842    "total": null
843  }
844  Transaction list retrieved
845
846 --- 14.2 Atypical Valid: List transactions with pagination
847  {
848    "success": true,
849    "message": "OK",
850    "data": {
851      "page": 1,
852      "size": 5,
853      "total": 3,
854      "items": [
855        {
856          "transaction_id": 6,
857          "txn_at": [
858            2025,
859            1,
860            26,
861            11,
862            0
863          ],
864          "type": "EXPENSE",
865          "note": "....."
866        }
867      ]
868    }
869  }
870

```

```

865     "currency": "USD",
866     "amount_total": 75.0000000,
867     "payer_id": 8,
868     "created_by": 8,
869     "note": "Split with non-member"
870   },
871   {
872     "transaction_id": 5,
873     "txn_at": [
874       2025,
875       1,
876       20,
877       9,
878       0
879     ],
880     "type": "INCOME",
881     "currency": "USD",
882     "amount_total": 200.0000000,
883     "payer_id": 8,
884     "created_by": 8,
885     "note": "Shared project payment received"
886   },
887   {
888     "transaction_id": 4,
889     "txn_at": [
890       2025,
891       1,
892       15,
893       12,
894       0
895     ],
896     "type": "EXPENSE",
897     "currency": "USD",
898     "amount_total": 120.5000000,
899     "payer_id": 8,
900     "created_by": 8,
901     "note": "Team lunch at restaurant"
902   }
903 ],
904 },
905 "total": null
906 }
907 Paginated list retrieved
908
909 --- 14.3 Atypical Valid: List transactions with date filter
910 {
911   "success": true,
912   "message": "OK",
913   "data": {
914     "page": 1,
915     "size": 50,
916     "total": 3,
917     "items": [
918       {
919         "transaction_id": 6,
920         "txn_at": [
921           2025,
922           1,
923           26,
924           11,
925           0
926         ],
927         "type": "EXPENSE",
928         "currency": "USD",
929         "amount_total": 75.0000000,
930         "payer_id": 8,
931         "created_by": 8,
932         "note": "Split with non-member"
933       },
934       {
935         "transaction_id": 5,
936         "txn_at": [
937           2025,
938           1,
939           20,
940           9,
941           0
942         ],
943         "type": "INCOME",
944         "currency": "USD",
945         "amount_total": 200.0000000,
946         "payer_id": 8,
947         "created_by": 8,
948         "note": "Shared project payment received"
949       },
950       {
951         "transaction_id": 4,
952         "txn_at": [
953           2025,
954           1,
955           15,
956           12,
957           0
958         ],
959         "type": "EXPENSE",
960         "currency": "USD",
961         "amount_total": 120.5000000,
962         "payer_id": 8,
963         "created_by": 8,

```

```

964         "note": "Team lunch at restaurant"
965     }
966   ]
967 },
968 "total": null
969 }
970 Date-filtered list retrieved
971
972 --- 14.4 Invalid: List transactions without auth token
973 {
974   "success": false,
975   "message": "AUTH_REQUIRED",
976   "data": null,
977   "total": null
978 }
979 Unauthorized access properly rejected
980
981 --- 14.5 Invalid: List transactions from non-existent ledger
982 {
983   "success": false,
984   "message": "Ledger not found",
985   "data": null,
986   "total": null
987 }
988 Non-existent ledger properly rejected
989
990 --- 14.6 Atypical Valid: List transactions with type filter
991 {
992   "success": true,
993   "message": "OK",
994   "data": {
995     "page": 1,
996     "size": 50,
997     "total": 2,
998     "items": [
999       {
1000         "transaction_id": 6,
1001         "txn_at": [
1002           2025,
1003           1,
1004           26,
1005           11,
1006           0
1007         ],
1008         "type": "EXPENSE",
1009         "currency": "USD",
1010         "amount_total": 75.0000000,
1011         "payer_id": 8,
1012         "created_by": 8,
1013         "note": "Split with non-member"
1014       },
1015       {
1016         "transaction_id": 4,
1017         "txn_at": [
1018           2025,
1019           1,
1020           15,
1021           12,
1022           0
1023         ],
1024         "type": "EXPENSE",
1025         "currency": "USD",
1026         "amount_total": 120.5000000,
1027         "payer_id": 8,
1028         "created_by": 8,
1029         "note": "Team lunch at restaurant"
1030       }
1031     ]
1032   },
1033   "total": null
1034 }
1035 Type-filtered list retrieved
1036 All transaction query tests completed!
1037
1038 =====
1039 => 15. DELETE TRANSACTION API TESTS
1040 =====
1041
1042 --- 15.1 Typical Valid: Delete own transaction
1043 Transaction deleted successfully
1044
1045 --- 15.2 Invalid: Delete transaction as non-owner/non-admin
1046 Unauthorized deletion properly rejected
1047
1048 --- 15.3 Invalid: Delete non-existent transaction
1049 {
1050   "success": false,
1051   "message": "Transaction not found",
1052   "data": null,
1053   "total": null
1054 }
1055 Non-existent transaction deletion properly handled
1056 All delete transaction tests completed!
1057
1058 =====
1059 => 16. FINAL VERIFICATION AND SUMMARY
1060 =====
1061
1062 --- 16.1 Verify user tokens are still valid

```

```

1063 User1 token still valid
1064 User2 token still valid
1065
1066 --- 16.2 Verify ledger ownership and membership
1067 {
1068   "success": true,
1069   "message": "OK",
1070   "data": {
1071     "items": [
1072       {
1073         "user_id": 8,
1074         "name": "Alex Chen",
1075         "role": "OWNER"
1076       },
1077       {
1078         "user_id": 10,
1079         "name": "Sarah Johnson",
1080         "role": "EDITOR"
1081       },
1082       {
1083         "user_id": 11,
1084         "name": "Mike Wilson",
1085         "role": "VIEWER"
1086       }
1087     ]
1088   },
1089   "total": null
1090 }
1091 Ledger membership verification completed
1092
1093 --- 16.3 Database final state check
1094 Users in database:
1095 Warning: [arning] Using a password on the command line interface can be insecure.
1096 id      email      name
1097 12      emma.davis.qa2025@gmail.com  Emma Davis
1098 11      mike.wilson.dev2025@gmail.com  Mike Wilson
1099 10      sarah.johnson.biz2025@gmail.com  Sarah Johnson
1100 9       short.minimal2025@gmail.com  AI
1101 8       alex.chen.eng2025@gmail.com  Alex Chen
1102 7       my@gmail.com  my
1103 6       bob.baker@example.org  user_b3ef93e0403f
1104 5       charlie.clark@example.org  Charlie Clark
1105 4       alice.anderson@example.org  Alice Anderson
1106 3       aaa@gmail.com  aaa
1107
1108 Ledgers created:
1109 Warning: [arning] Using a password on the command line interface can be insecure.
1110 id      name      owner_id      ledger_type
1111 8      Family Vacation Fund 2025  8      GROUP_BALANCE
1112 7      Tech Team Road Trip 2025  8      GROUP_BALANCE
1113 6      Road Trip 2025  4      GROUP_BALANCE
1114 5      Road Trip 2025  4      GROUP_BALANCE
1115 4      Road Trip 2025  4      GROUP_BALANCE
1116
1117 Transactions in main ledger:
1118 id      type      amount_total      note
1119 6      EXPENSE  75.00000000  Split with non-member
1120 5      INCOME   200.00000000  Shared project payment received
1121 4      EXPENSE  120.50000000  Team lunch at restaurant
1122
1123 =====
1124 => API TESTS COMPLETED!
1125 =====
1126 All core API functionality has been comprehensively verified!
1127 Every endpoint includes typical valid, atypical valid, and invalid input tests as required.
1128
1129 Stopping Spring Boot application (PID: 5997)...

```

✓ Run API tests (test_ledger_filtering)

```

1  ► Run cd API_test
14
15 =====
16 => LEDGER FILTERING SQL QUERY TEST
17 =====
18 This test verifies that the SQL query fix properly filters transactions by ledger_id
19 and prevents cross-ledger transaction leakage.
20
21 --- 1) SETTING UP TEST DATABASE
22 Dropping database ledger ...
23 Warning: [arning] Using a password on the command line interface can be insecure.
24 Creating database ledger ...
25 Warning: [arning] Using a password on the command line interface can be insecure.
26 Loading schema: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/ledger_flow.sql
27 Warning: [arning] Using a password on the command line interface can be insecure.
28 Loading seed data: /home/runner/work/flyingcloud-4156-project/flyingcloud-4156-project/ops/sql/backup/ledger_big_seed.sql
29 Warning: [arning] Using a password on the command line interface can be insecure.
30
31 --- 2) VERIFYING DATABASE STATE
32 Available ledgers:
33 Warning: [arning] Using a password on the command line interface can be insecure.
34   Ledger 1: Road Trip Demo
35 Warning: [arning] Using a password on the command line interface can be insecure.
36   Transactions: 10
37   Ledger 2: Apartment Demo
38 Warning: [arning] Using a password on the command line interface can be insecure.
39   Transactions: 10
40
41 =====

```

```
42 => 3) TESTING SQL QUERY LEDGER FILTERING
43 =====
44 Testing the fixed SQL query logic by simulating different scenarios...
45
46 --- 3.1) TESTING LEDGER 1 (ROAD TRIP DEMO) QUERY
47 Simulating query for ledger 1 with user alice (id=1)...
```