

# Probando con un proveedor de OPEN ID Connect - Auth0

Para una lista de implementaciones certificadas de OpenID Connect, ver el link <https://openid.net/developers/certified/>. Usaremos Auth0, <https://auth0.com>, para nuestras pruebas con un proveedor de OpenID. Para poder usar Auth0 en lugar de nuestro servidor de autorización, iremos, a través de estos pasos:

- Crear una cuenta en auth0
- Aplicar los cambios requeridos para usar Auth0 como un proveedor OpenID y ejecutar el test script para verificar que está trabajando
- Adquirir tokens de acceso usando lo siguiente:
  - Password grant flow
  - Implicit grant flow
  - Authorization code grant flow
- Invocar nuestras APIs usando los tokens de acceso
- Usar el endpoint **user info** para obtener más información del usuario

## Configurar una cuenta en Auth0

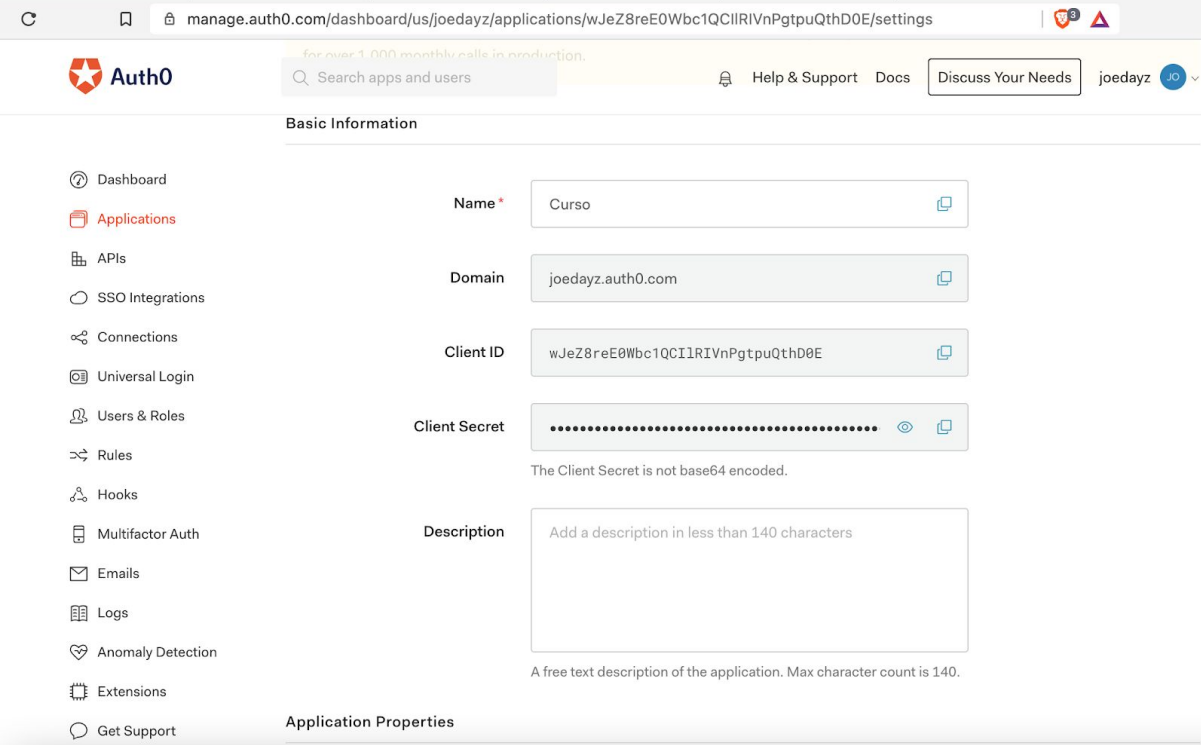
Vamos a ejecutar los siguientes pasos para abrir una cuenta gratuita en Auth0, configurar el cliente OAuth 2.0 y el API product-composite, y finalmente registrar un usuario:

1. Abre el URL, <https://auth0.com>, en tu navegador.
2. Clic en el botón SIGN UP:
  - a. Crea tu cuenta con la opción de tu elección (red social o una combinación de email y password que tu desees)

- b. Después de crear tu cuenta, se te pedirá el nombre del tenant. En mi caso le puse **joedayz.auth0.com**, tu colócale el que gustes.
  - c. Completa toda la información que sea requerida y listo.
3. Luego de sacar tu cuenta, se te redireccionará al dashboard. Selecciona el tab de Applications (al lado izquierdo) para ver el cliente por defecto que fue creado en el momento que abriste la cuenta.

NOTA: Yo borre el Default App y cree una aplicación desde cero, por eso no me sale en la lista.

4. Clic en Default App (ó si deseas creas una App desde cero):
  - a. Copia el Client ID y Client Secret; se necesitarán ambos después.



The screenshot shows the Auth0 dashboard interface. The browser address bar displays the URL: `manage.auth0.com/dashboard/us/joedayz/applications/wJeZ8reE0Wbc1QCIIRIVnPgtpuQthD0E/settings`. The Auth0 logo is in the top left, and a search bar is in the top center. The left sidebar contains a navigation menu with items: Dashboard, Applications (highlighted), APIs, SSO Integrations, Connections, Universal Login, Users & Roles, Rules, Hooks, Multifactor Auth, Emails, Logs, Anomaly Detection, Extensions, and Get Support. The main content area is titled 'Basic Information' and contains the following fields:

- Name \***: A text input field containing 'Curso'.
- Domain**: A text input field containing 'joedayz.auth0.com'.
- Client ID**: A text input field containing 'wJeZ8reE0Wbc1QCIIRIVnPgtpuQthD0E'.
- Client Secret**: A text input field filled with dots, with a copy icon and an eye icon. Below it, a message states: 'The Client Secret is not base64 encoded.'
- Description**: A text area with a placeholder 'Add a description in less than 140 characters'. Below it, a message states: 'A free text description of the application. Max character count is 140.'

At the bottom of the main content area, the text 'Application Properties' is visible.

- b. Cómo tipo de aplicación, selecciona Machine to Machine.

manage.auth0.com/dashboard/us/joedayz/applications/wJeZ8reE0Wbc1QCIIrIVnPgtpuQthD0E/settings

Auth0

Search apps and users

Help & Support Docs Discuss Your Needs joedayz

### Application Properties

- Dashboard
- Applications**
- APIs
- SSO Integrations
- Connections
- Universal Login
- Users & Roles
- Rules
- Hooks
- Multifactor Auth
- Emails
- Logs
- Anomaly Detection
- Extensions
- Get Support

**Application Logo**

https://path.to/my\_logo.png

The URL of the logo to display for the application, if none is set the default badge for this type of application will be shown. Recommended size is 150x150 pixels.

**Application Type**

Machine to Machine

The type of application will determine which settings you can configure from the dashboard.

**Token Endpoint Authentication Method**

Post

Defines the requested authentication method for the token endpoint. Possible values are 'None' (public application without a client secret), 'Post' (application uses HTTP POST parameters) or 'Basic' (application uses HTTP Basic).

**Use Auth0 instead of the IdP to do Single Sign On**

☐

If this setting is enabled, Auth0 will handle Single Sign On instead of the Identity Provider (e.g.: No redirect to Facebook to log the user in if they have already logged in before).

c. Cómo Token Endpoint Authentication Method, selecciona POST.

manage.auth0.com/dashboard/us/joedayz/applications/wJeZ8reE0Wbc1QCIIrIVnPgtpuQthD0E/settings

Auth0

Search apps and users

Help & Support Docs Discuss Your Needs joedayz

**Application Type**

Machine to Machine

The type of application will determine which settings you can configure from the dashboard.

**Token Endpoint Authentication Method**

Post

Defines the requested authentication method for the token endpoint. Possible values are 'None' (public application without a client secret), 'Post' (application uses HTTP POST parameters) or 'Basic' (application uses HTTP Basic).

**Use Auth0 instead of the IdP to do Single Sign On**

☐

If this setting is enabled, Auth0 will handle Single Sign On instead of the Identity Provider (e.g.: No redirect to Facebook to log the user in if they have already logged in before).

**Application URIs**

d. Ingresa <http://my.redirect.uri> como el callback URL.

manage.auth0.com/dashboard/us/joedayz/applications/wJeZ8reE0Wbc1QCIIrIVnPgtpuQthD0E/settings

Auth0

Search apps and users

Help & Support Docs Discuss Your Needs joedayz

### Application URIs

Dashboard

**Applications**

APIs

SSO Integrations

Connections

Universal Login

Users & Roles

Rules

Hooks

Multifactor Auth

Emails

Logs

Anomaly Detection

Extensions

Get Support

**Application Login URI**

https://my.redirect.uri

In some scenarios, Auth0 will need to redirect to your application's login page. This URI needs to point to a route in your application that should redirect to your tenant's `/authorize` endpoint. [Learn more](#)

**Allowed Callback URLs**

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol ( `https://` ) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://`.

**Allowed Logout URLs**

A set of URLs that are valid to redirect to after logout from Auth0. After a user logs out from Auth0 you can redirect them with the `returnTo`

- e. Clic en Show Advanced Settings, ve al tab Grant Types, deselecciona Client Credentials, y selecciona el Password box.

manage.auth0.com/dashboard/us/joedayz/applications/wJeZ8reE0Wbc1QCIIrIVnPgtpuQthD0E/settings

Auth0

Search apps and users

Help & Support Docs Discuss Your Needs joedayz

Dashboard

**Applications**

APIs

SSO Integrations

Connections

Universal Login

Users & Roles

Rules

Hooks

Multifactor Auth

Emails

Logs

Anomaly Detection

Extensions

Get Support

**Refresh Token Lifetime (Absolute)**

2592000

The absolute lifetime a `refresh_token` can be used to retrieve an access token.

**Refresh Token Reuse Interval**

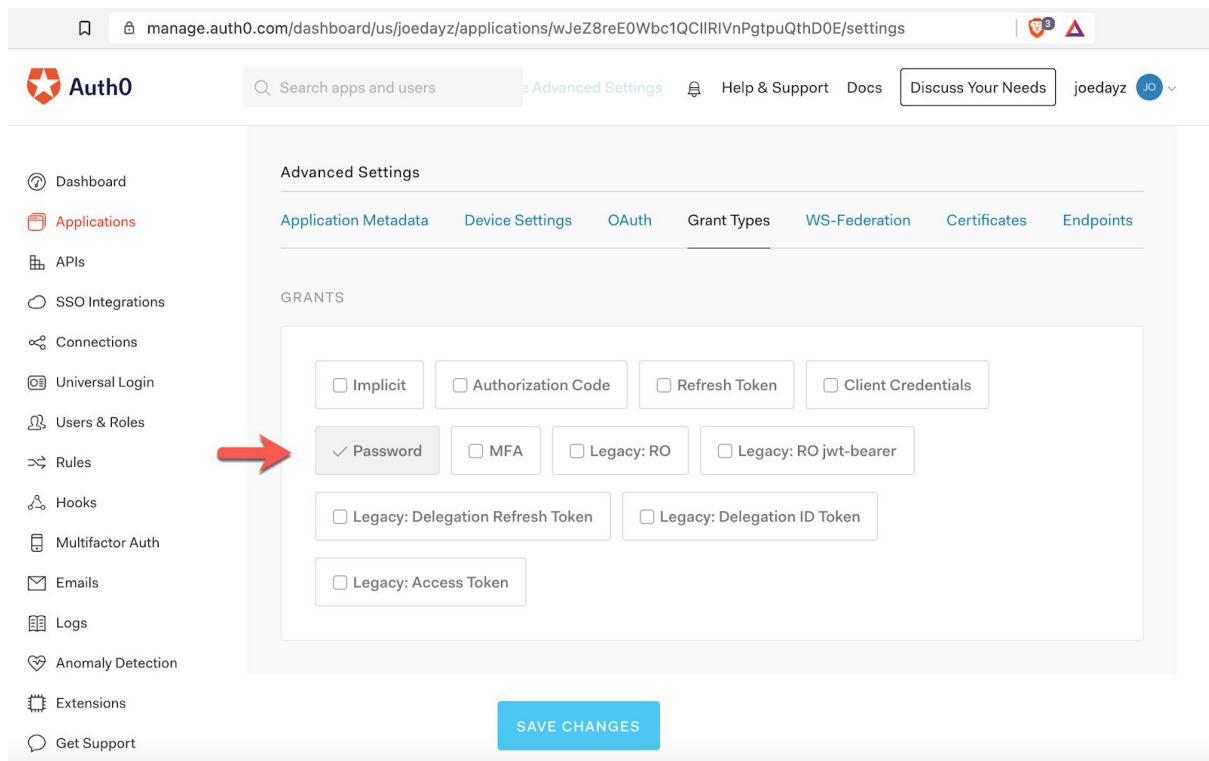
0

When refresh tokens are configured for rotation, the allowable time (in seconds) that the same refresh token can be used to request an `access_token` without triggering automatic reuse detection. This setting represents the `leeway` property.

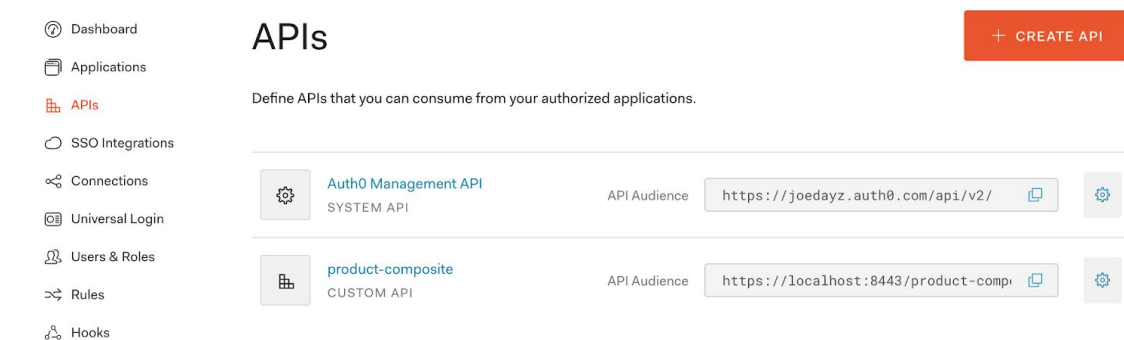
[Show Advanced Settings](#)

**SAVE CHANGES**


**Danger Zone**




- f. Clic en SAVE CHANGES.
5. Ahora define las autorizaciones para nuestras APIs:
  - a. Clic en el tab APIs (al lado izquierdo) y clic en el botón + CREATE API.



- b. Coloca el nombre del API como product-composite, y dale el identificador <https://localhost:8443/product-composite>, y clic en el botón CREATE.

 **Auth0**

Search apps and users

Help & Support Docs Discuss Your Needs joedayz 

Dashboard

Applications

APIs

SSO Integrations

Connections

Universal Login

Users & Roles

Rules

Hooks

Multifactor Auth


Emails

Logs

Anomaly Detection

Extensions

Get Support



# product-composite

CUSTOM API Identifier `https://localhost:8443/product-composite`

Quick Start

Settings

Permissions

Machine to Machine Applications

Test

## General Settings

Id

5ebcbfd535b812000e0ae6f0

The API id on our system. Useful if you prefer to work directly with Auth0's Management API instead.

Name \*

product-composite

A friendly name for the API. The following characters are not allowed < >


Identifier

https://localhost:8443/product-composite


Unique identifier for the API. This value will be used as the audience parameter in authorization calls.

c. Clic en el tab Permissions y crea dos permisos (esto es, scopes de OAuth): **product:read** y **product:write**.

manage.auth0.com/dashboard/us/joedayz/apis/5ebcbfd535b812000e0ae6f0/permissions

 **Auth0**

Search apps and users

Help & Support Docs Discuss Your Needs joedayz 

Dashboard

Applications

APIs

SSO Integrations

Connections

Universal Login

Users & Roles

Rules

Hooks

Multifactor Auth


Emails

Logs

Anomaly Detection

Extensions

Get Support



# product-composite

CUSTOM API Identifier `https://localhost:8443/product-composite`

Quick Start

Settings

Permissions

Machine to Machine Applications

Test

## Add a Permission (Scope)

Define the permissions (scopes) that this API uses.

Permission (Scope) \*

read:appointments

Description \*

Read your appointments

+ ADD

### List of Permissions (Scopes)

These are all the permissions (scopes) that this API uses.

Permission	Description
product:read	product read
product:write	product write

6. Luego, crea un usuario:

- a. Clic en Users & Roles y en el tab Users (al lado izquierdo) y luego en el botón CREATE YOUR FIRST USER.

manage.auth0.com/dashboard/us/joedayz/users

Auth0 Search apps and users Help & Support Docs Discuss Your Needs joedayz

Dashboard Applications APIs SSO Integrations Connections Universal Login **Users & Roles** Rules Hooks

**Users** + CREATE USER

An easy to use UI to help administrators manage user identities including password resets, creating and provisioning, blocking and deleting users. [Learn more](#)

Search for users Search by User RESET

Name	Connection	Logins	Latest Login
JA jamdiazdiaz@gmail.com jamdiazdiaz@gmail.com	Username-Password-Authen...	5	4 months ago
JD Jose Diaz Diaz (empty)	facebook	18	a year ago

- b. Ingresa un email y password de tu preferencia y clic en el botón SAVE.
  - c. Busca el email de verificación que se te va a enviar desde Auth0 y confirma.
7. Finalmente, valida tu configuración de Default Directory, usado por el password grant flow:
- a. Clic en el profile de tu tenant en la esquina superior derecha y selecciona Settings.

Discuss Your Needs joedayz

joedayz US - Free

Settings

Invite an admin

Create tenant

Switch tenant

Account usage

Logout

jose.diaz@joeday... View profile



- b. En tab llamado General, baja hacia el campo denominado Default Directory y verifica que esta contenga el valor Username-Password-Authentication. Si no es así, actualiza el campo y guarda el cambio.

The screenshot shows the Auth0 dashboard interface. The left sidebar contains navigation links: Dashboard, Applications, APIs, SSO Integrations, Connections, Universal Login, Users & Roles (with sub-links for Users and Roles), Rules, Hooks, Multifactor Auth, Emails, Logs, Anomaly Detection, Extensions, and Get Support. The main content area is titled 'API Authorization Settings'. It features two input fields: 'Default Audience' with the value 'https://your-default-endpoint/' and 'Default Directory' with the value 'Username-Password-Authentication'. A red arrow points to the 'Default Directory' field. Below each field is a 'SAVE' button. The 'Default Directory' field has a descriptive text below it: 'Name of the connection to be used for Password Grant exchanges. The default\_directory value should be the exact name of an existing connections of one of the following strategies: ad, auth0, email, sms, waad, adfs'.

8. Eso es todo. Usaremos el client ID y secret para el default app; este es el cliente OAuth.

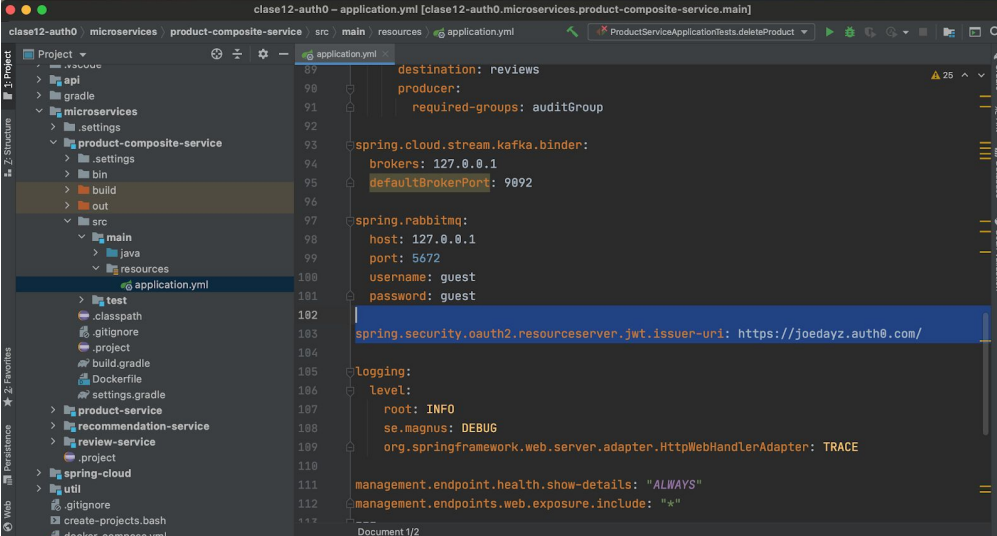
Con la cuenta Auth0 creada y configurada, ya nos podemos mover y aplicar la configuración necesaria al sistema de completo.

## Aplicar los cambios necesarios para usar Auth0 como un proveedor OpenID

Vamos a ver qué cambios son necesarios para reemplazar el servidor de autorización por Auth0. Solo se necesita cambiar la configuración para los dos servicios que actúan como servidores de recursos OAuth, el *product-composite*, y el *gateway*. También se necesita

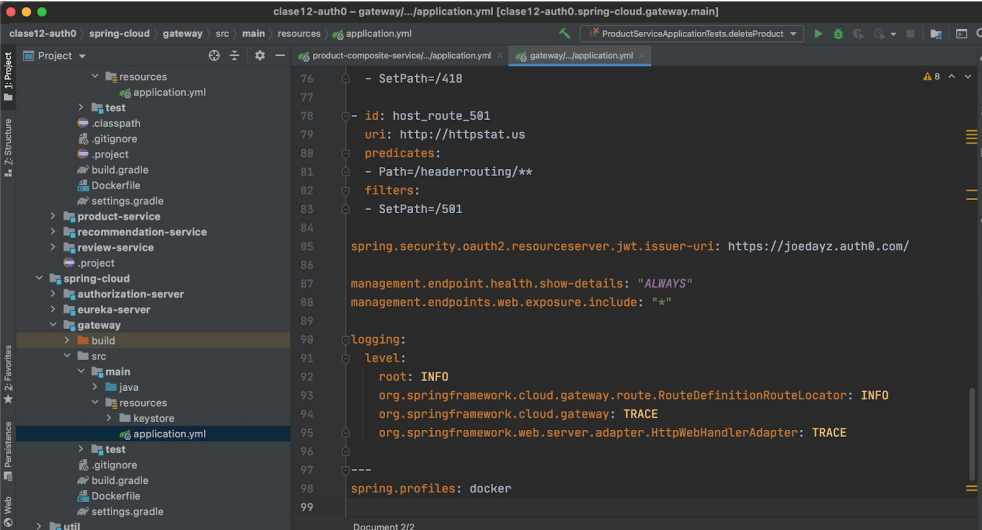
cambiar el test script, para que pueda acceder a los tokens de Auth0 en lugar de nuestro servidor de autorización local. Vamos a iniciar con los servidores de recurso OAuth, esto es, el product-composite y el gateway.

## 1. Colocar en product-composite o gateway



```
87 destination: reviews
88 producer:
89   required-groups: auditGroup
90
91
92
93 spring.cloud.stream.kafka.binder:
94   brokers: 127.0.0.1
95   defaultBrokerPort: 9092
96
97
98 spring.rabbitmq:
99   host: 127.0.0.1
100   port: 5672
101   username: guest
102   password: guest
103
104 spring.security.oauth2.resourceserver.jwt.issuer-uri: https://joedayz.auth0.com/
105
106 logging:
107   level:
108     root: INFO
109     se.magnus: DEBUG
110     org.springframework.web.server.adapter.HttpWebHandlerAdapter: TRACE
111
112 management.endpoint.health.show-details: "ALWAYS"
113 management.endpoints.web.exposure.include: "*"
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Application.yml en Product-composite



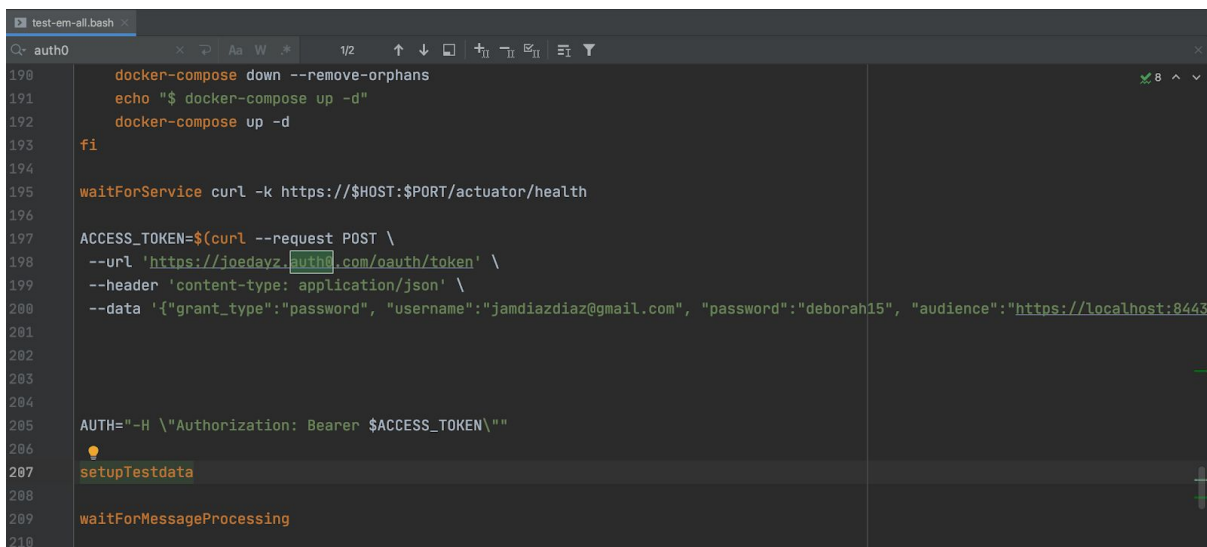
```
76 - SetPath=/418
77
78 - id: host_route_501
79 uri: http://httpstat.us
80 predicates:
81   - Path=/headerrouting/**
82 filters:
83   - SetPath=/501
84
85 spring.security.oauth2.resourceserver.jwt.issuer-uri: https://joedayz.auth0.com/
86
87 management.endpoint.health.show-details: "ALWAYS"
88 management.endpoints.web.exposure.include: "*"
89
90 logging:
91   level:
92     root: INFO
93     org.springframework.cloud.gateway.route.RouteDefinitionRouteLocator: INFO
94     org.springframework.cloud.gateway: TRACE
95     org.springframework.web.server.adapter.HttpWebHandlerAdapter: TRACE
96
97 ---
98 spring.profiles: docker
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Application.yml en Gateway

## 2. Luego de hacer el cambio:

- `./gradlew clean build`
- `docker-compose build`
- Cambiar el test script. Te muestro como hice en mi caso usando mi tenat: `joedayz.auth0.com`

```
ACCESS_TOKEN=$(curl --request POST \
--url 'https://joedayz.auth0.com/oauth/token' \
--header 'content-type: application/json' \
--data '{"grant_type":"password", "username":"jamdiazdiaz@gmail.com",
"password":"deborah15", "audience":"https://localhost:8443/product-composite",
"scope":"openid email product:read product:write", "client_id":
"wJeZ8reE0Wbc1QCIIrIVnPgtpuQthD0E", "client_secret":
"1BXVGV0X_IsCZlB71IN4FDRti4P9mT0PzG7rc0lsxxxxxxx" }' -s | jq -r .access_token)
```



```
test-em-all.bash
auth0
190 docker-compose down --remove-orphans
191 echo "$ docker-compose up -d"
192 docker-compose up -d
193 fi
194
195 waitForService curl -k https://$HOST:$PORT/actuator/health
196
197 ACCESS_TOKEN=$(curl --request POST \
198 --url 'https://joedayz.auth0.com/oauth/token' \
199 --header 'content-type: application/json' \
200 --data '{"grant_type":"password", "username":"jamdiazdiaz@gmail.com", "password":"deborah15", "audience":"https://localhost:8443
201
202
203
204
205 AUTH="-H \"Authorization: Bearer $ACCESS_TOKEN\""
206
207 setupTestData
208
209 waitForMessageProcessing
210
```

# Verify that the reader - client with only read scope can call the read API but not delete API.

```
READER_ACCESS_TOKEN=$(curl --request POST \
--url 'https://joedayz.auth0.com/oauth/token' \
--header 'content-type: application/json' \
--data '{"grant_type":"password", "username":"jamdiazdiaz@gmail.com",
"password":"deborah15", "audience":"https://localhost:8443/product-composite",
```

```
"scope":"openid email product:read", "client_id":  
"wJeZ8reE0Wbc1QCIIrIVnPgtpuQthD0E", "client_secret":  
"1BXVGV0X_IsCZlb71IN4FDRti4P9mT0PzG7rc0lsUxxxxxxxxx" }' -s | jq -r .access_token)
```

```
test-em-all.bash x  
Q- auth0 x Aa W .* 2/2 ↑ ↓ ↻ + - = < > |  
239  
240 # Verify that a request without access token fails on 401, Unauthorized  
241 assertCurl 401 "curl -k https://$HOST:$PORT/product-composite/$PROD_ID_REVS_RECS -s"  
242  
243 # Verify that the reader - client with only read scope can call the read API but not delete API.  
244  
245 READER_ACCESS_TOKEN=$(curl --request POST \  
246 --url 'https://joedayz.auth0.com/oauth/token' \  
247 --header 'content-type: application/json' \  
248 --data '{"grant_type":"password", "username":"jamdiazdiaz@gmail.com", "password":"deborah15", "audience":"https://localhost:  
249  
250  
251  
252  
253 READER_AUTH="-H \"Authorization: Bearer $READER_ACCESS_TOKEN\""  
254  
255 assertCurl 200 "curl -k https://$HOST:$PORT/product-composite/$PROD_ID_REVS_RECS $READER_AUTH -s"  
256 assertCurl 403 "curl -k https://$HOST:$PORT/product-composite/$PROD_ID_REVS_RECS $READER_AUTH -X DELETE -s"  
257  
258 echo "End, all tests OK:" `date`  
259  
260 if [[ $@ == *"stop"* ]]  
261 then  
262     echo "Stopping the test environment..."
```

Eso es todo lo que hay que cambiar a nuestro aplicativo.

## Ejecutar el test script con Auth0 como el proveedor de OpenID

`./test-em-all.bash start` (nota: asegurate que no haya ningún contenedor de algún ejemplo previo corriendo).

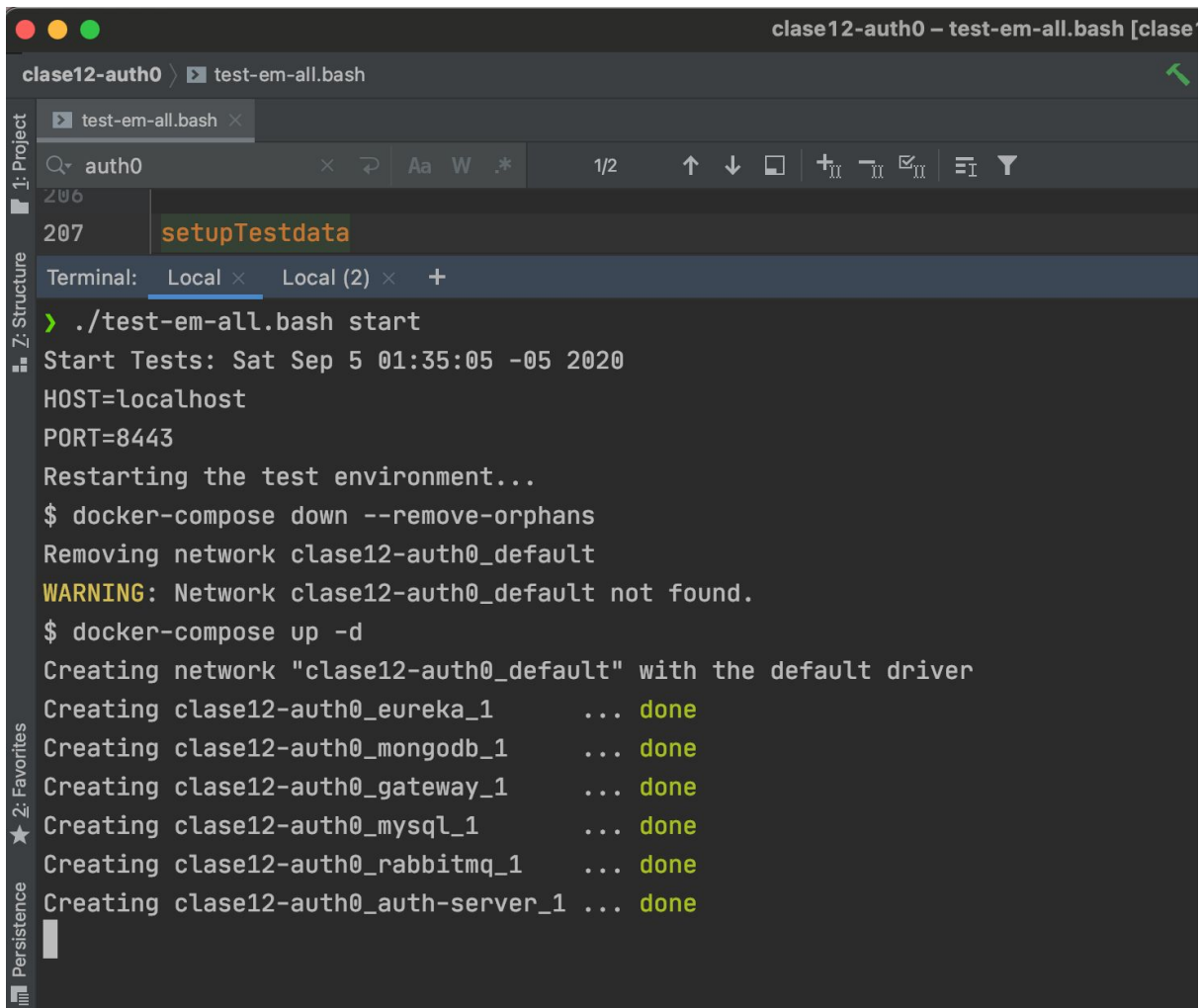
En los logs (usando `docker-compose logs -f product-composite`), veremos la información de autorización desde el access token provisto por Auth0:

Llamadas usando los scopes: `product:read`, `product:write`.

```
2. @700dd0df9ffe:/ (bash)  
product-composite_1 ...  
Authorization info: Subject: auth0l5ca0b73c97f31e11bc85a5e6,  
scopes: openid email product:read product:write,  
expires Mon Apr 08 10:46:35 UTC 2019: issuer: https://dev-ml.eu.auth0.com/,  
audience: [https://localhost:8443/product-composite, https://dev-ml.eu.auth0.com/userinfo]
```

```
product-composite_1 ...
Authorization info: Subject: auth015ca0b73c97f31e11bc85a5e6,
scopes: openid email product:read,
expires Mon Apr 08 10:46:41 UTC 2019: issuer: https://dev-ml.eu.auth0.com/,
audience: [https://localhost:8443/product-composite, https://dev-ml.eu.auth0.com/userinfo]
```

Los contenedores van a iniciar:



```
clase12-auth0 - test-em-all.bash [clase12-auth0]
test-em-all.bash
auth0
207 setupTestdata
Terminal: Local x Local (2) x +
> ./test-em-all.bash start
Start Tests: Sat Sep 5 01:35:05 -05 2020
HOST=localhost
PORT=8443
Restarting the test environment...
$ docker-compose down --remove-orphans
Removing network clase12-auth0_default
WARNING: Network clase12-auth0_default not found.
$ docker-compose up -d
Creating network "clase12-auth0_default" with the default driver
Creating clase12-auth0_eureka_1 ... done
Creating clase12-auth0_mongodb_1 ... done
Creating clase12-auth0_gateway_1 ... done
Creating clase12-auth0_mysql_1 ... done
Creating clase12-auth0_rabbitmq_1 ... done
Creating clase12-auth0_auth-server_1 ... done
```

El resultado final debe ser:

The screenshot shows an IDE terminal window titled "clase12-auth0 - test-em-all.bash [clase12-auth0]". The terminal output displays the results of a test suite. It begins with a "Test OK" message for an HTTP 404 status, followed by several "Test OK" messages for various HTTP status codes (200, 208, 400, 422, 401, 200, 403) and their corresponding actual values. The tests cover scenarios like "No product found for productId: 14", "Invalid productId: -1", and "Type mismatch". The terminal concludes with "End, all tests OK: Sat Sep 5 01:43:37 -05 2020". The IDE interface includes a sidebar with "Project", "Structure", "Favorites", "Persistence", and "Web" views, and a bottom status bar showing "207:14 LF UTF-8 2 spaces".

```
clase12-auth0 - test-em-all.bash [clase12-auth0]
test-em-all.bash
207
Test OK (HTTP Code: 404, {"timestamp":"2020-09-05T06:43:31.707939Z","path":"/product-composite/14","message":"No product found for productId: 14","status":404,"error":"Not Found"})
Test OK (HTTP Code: 200)
Test OK (actual value: 114)
Test OK (actual value: 0)
Test OK (actual value: 3)
Test OK (HTTP Code: 200)
Test OK (actual value: 214)
Test OK (actual value: 3)
Test OK (actual value: 0)
Test OK (HTTP Code: 422, {"timestamp":"2020-09-05T06:43:33.644853Z","path":"/product-composite/-1","message":"Invalid productId: -1","status":422,"error":"Unprocessable Entity"})
Test OK (actual value: "Invalid productId: -1")
Test OK (HTTP Code: 400, {"timestamp":"2020-09-05T06:43:34.328+00:00","path":"/product-composite/invalidProductId","status":400,"error":"Bad Request","message":"Type mismatch.","requestId":"41b86db8-19"})
Test OK (actual value: "Type mismatch.")
Test OK (HTTP Code: 401, )
Test OK (HTTP Code: 200)
Test OK (HTTP Code: 403, )
End, all tests OK: Sat Sep 5 01:43:37 -05 2020
~/Downloads/clase12-auth0
```

Enjoy!

Joe