Java Advanced Programming

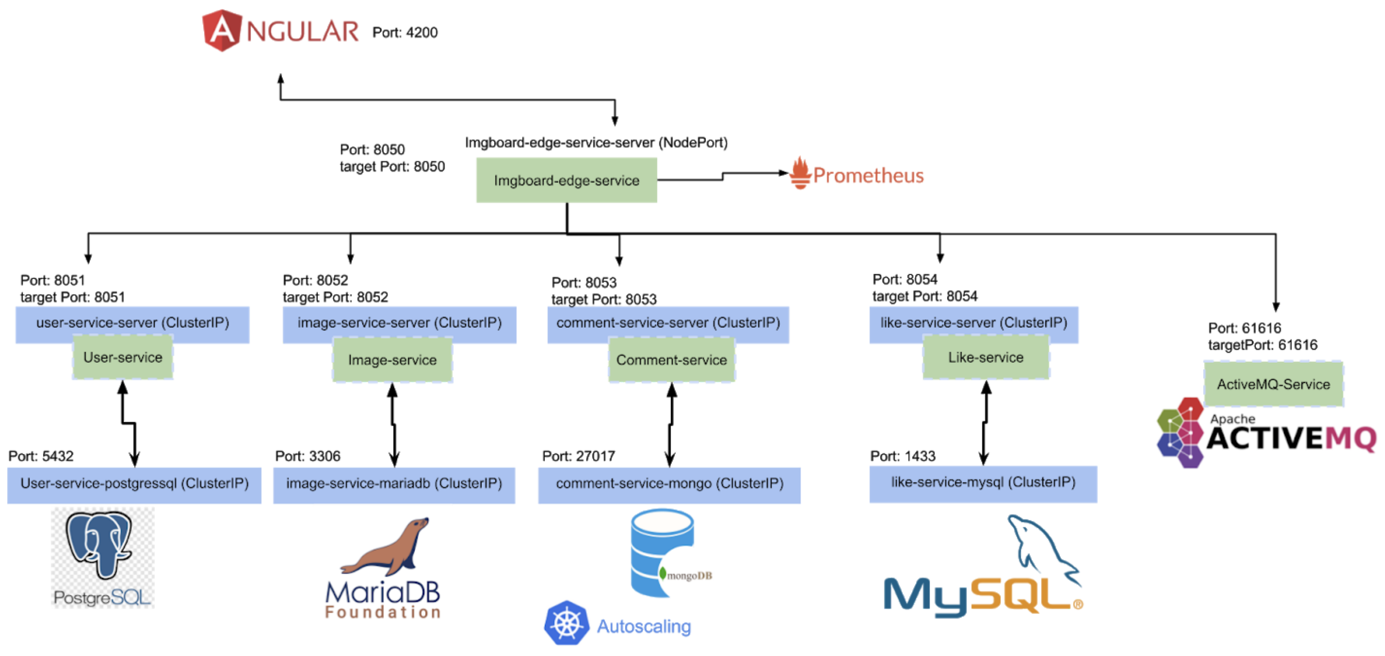
2020-2021

Youri Van Laer
Gust van der Sanden
Robin Vranckx
Robin Thoelen

Thema

Met deze services willen we een foto board (IMG board) maken. Hierbij kan de gebruiker registreren/inloggen waarbij hij zelf een foto kan plaatsen. Ook kan hij andere foto’s liken en commentaar geven.

Structuur


ANGULAR
Port: 4200
Imgboard-edge-service-server (NodePort)
Port: 8050
target Port: 8050
Imgboard-edge-service
Prometheus
Port: 8051
target Port: 8051
user-service-server (ClusterIP)
User-service
Port: 8052
target Port: 8052
image-service-server (ClusterIP)
Image-service
Port: 8053
target Port: 8053
comment-service-server (ClusterIP)
Comment-service
Port: 8054
target Port: 8054
like-service-server (ClusterIP)
Like-service
Port: 61616
targetPort: 61616
ActiveMQ-Service
Apache ACTIVEMQ
Port: 5432
User-service-postgressql (ClusterIP)
Port: 3306
image-service-mariadb (ClusterIP)
Port: 27017
comment-service-mongo (ClusterIP)
Port: 1433
like-service-mysql (ClusterIP)
PostgreSQL
MariaDB Foundation
mongoDB
Autoscaling
MySQL

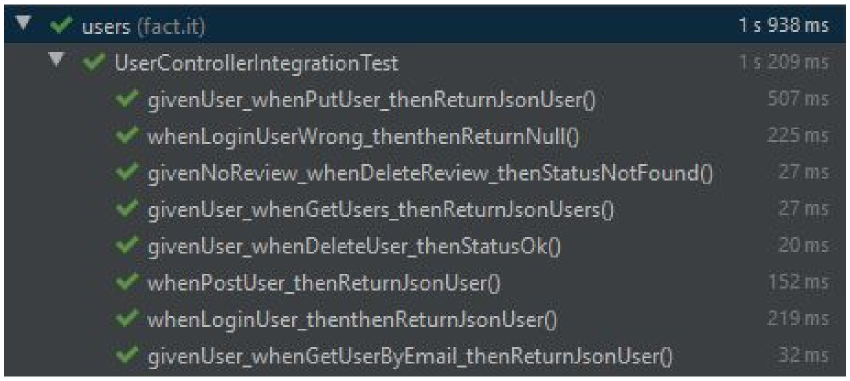
MicroServices

User

Robin Vranckx

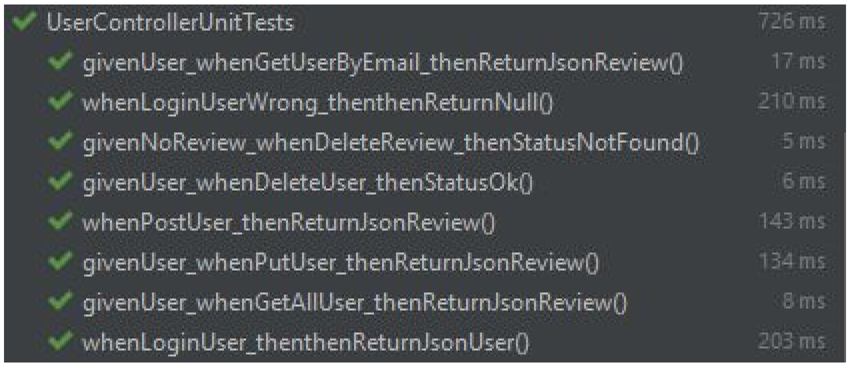
Github: https://github.com/vranckxrobin/users-service
Docker: docker push r0703028/users-ex

Integration Test



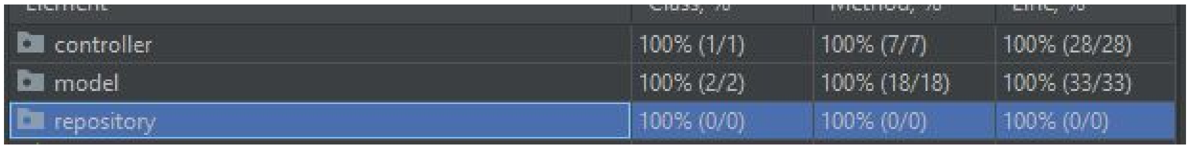
```
▼ ✓ users (fact.it)                                              1 s 938 ms
   ▼ ✓ UserControllerIntegrationTest                             1 s 209 ms
        ✓ givenUser_whenPutUser_thenReturnJsonUser()                 507 ms
        ✓ whenLoginUserWrong_thenthenReturnNull()                    225 ms
        ✓ givenNoReview_whenDeleteReview_thenStatusNotFound()         27 ms
        ✓ givenUser_whenGetUsers_thenReturnJsonUsers()                27 ms
        ✓ givenUser_whenDeleteUser_thenStatusOk()                     20 ms
        ✓ whenPostUser_thenReturnJsonUser()                          152 ms
        ✓ whenLoginUser_thenthenReturnJsonUser()                     219 ms
        ✓ givenUser_whenGetUserByEmail_thenReturnJsonUser()           32 ms
```

Unit Test

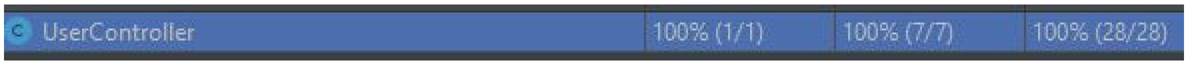


```
✓ UserControllerUnitTests                                          726 ms
   ✓ givenUser_whenGetUserByEmail_thenReturnJsonReview()            17 ms
   ✓ whenLoginUserWrong_thenthenReturnNull()                       210 ms
   ✓ givenNoReview_whenDeleteReview_thenStatusNotFound()             5 ms
   ✓ givenUser_whenDeleteUser_thenStatusOk()                         6 ms
   ✓ whenPostUser_thenReturnJsonReview()                           143 ms
   ✓ givenUser_whenPutUser_thenReturnJsonReview()                  134 ms
   ✓ givenUser_whenGetAllUser_thenReturnJsonReview()                 8 ms
   ✓ whenLoginUser_thenthenReturnJsonUser()                        203 ms
```

Test Coverage



controller	100% (1/1)	100% (7/7)	100% (28/28)
model	100% (2/2)	100% (18/18)	100% (33/33)
repository	100% (0/0)	100% (0/0)	100% (0/0)



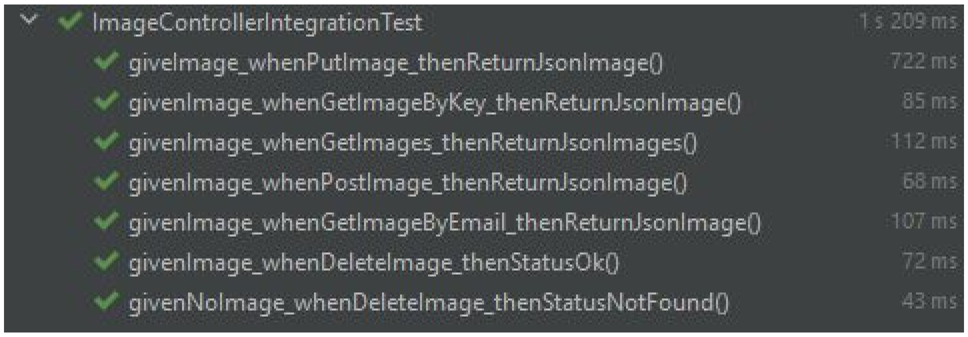
UserController	100% (1/1)	100% (7/7)	100% (28/28)

Images

Gust van der Sanden

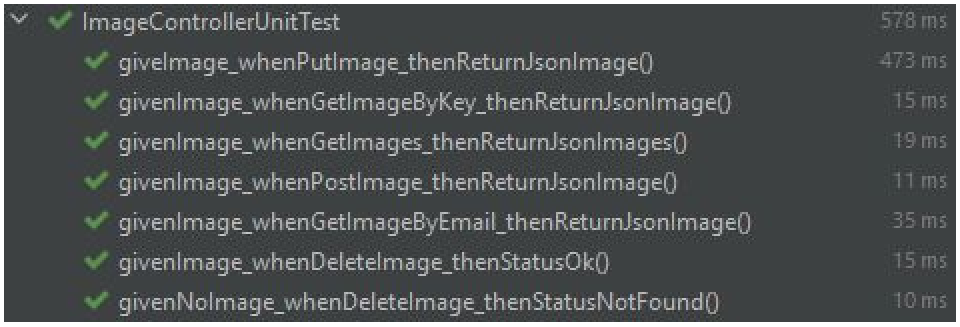
Github: https://github.com/gustvdsanden/microserviceImagesJava
Docker: docker push gustvdsanden/image-service-ex

Integration Test



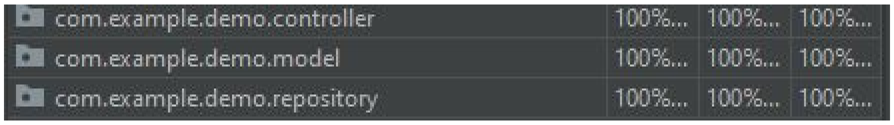
```
ImageControllerIntegrationTest                              1 s 209 ms
    giveImage_whenPutImage_thenReturnJsonImage()              722 ms
    givenImage_whenGetImageByKey_thenReturnJsonImage()         85 ms
    givenImage_whenGetImages_thenReturnJsonImages()           112 ms
    givenImage_whenPostImage_thenReturnJsonImage()             68 ms
    givenImage_whenGetImageByEmail_thenReturnJsonImage()      107 ms
    givenImage_whenDeleteImage_thenStatusOk()                  72 ms
    givenNoImage_whenDeleteImage_thenStatusNotFound()          43 ms
```

Unit Test



```
ImageControllerUnitTest                                       578 ms
    giveImage_whenPutImage_thenReturnJsonImage()              473 ms
    givenImage_whenGetImageByKey_thenReturnJsonImage()         15 ms
    givenImage_whenGetImages_thenReturnJsonImages()            19 ms
    givenImage_whenPostImage_thenReturnJsonImage()             11 ms
    givenImage_whenGetImageByEmail_thenReturnJsonImage()       35 ms
    givenImage_whenDeleteImage_thenStatusOk()                  15 ms
    givenNoImage_whenDeleteImage_thenStatusNotFound()          10 ms
```

Test Coverage



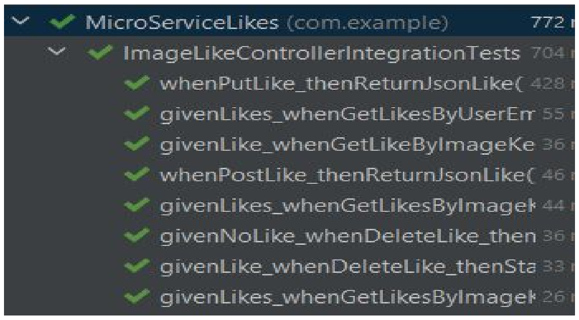
```
com.example.demo.controller     100%...  100%...  100%...
com.example.demo.model          100%...  100%...  100%...
com.example.demo.repository     100%...  100%...  100%...
```

Like

Robin Thoelen

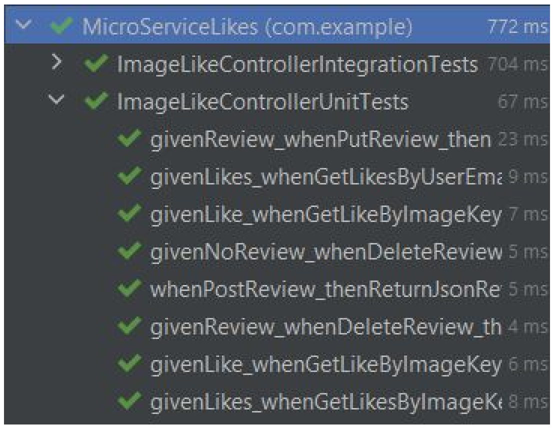
Github: https://github.com/RoboticRobin20/MicroServiceLikes
Docker: docker push robinthoelen/like-service-ex

Integration Test



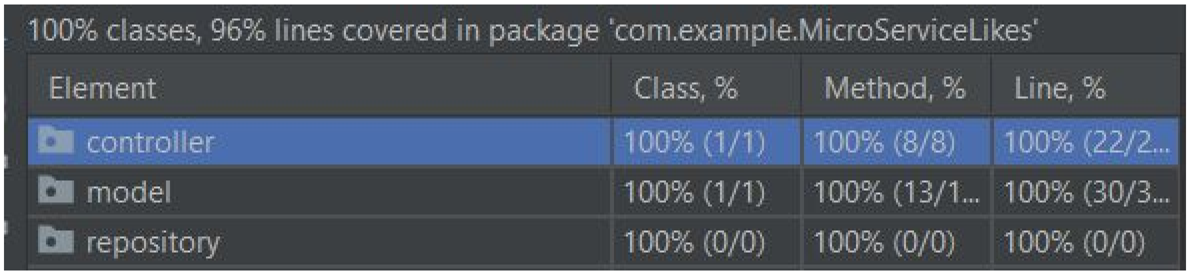
```
MicroServiceLikes (com.example)                 772 r
  ImageLikeControllerIntegrationTests           704 r
    whenPutLike_thenReturnJsonLike(             428 r
    givenLikes_whenGetLikesByUserEm             55 r
    givenLike_whenGetLikeByImageKe              36 r
    whenPostLike_thenReturnJsonLike(            46 r
    givenLikes_whenGetLikesByImageK             44 r
    givenNoLike_whenDeleteLike_then             36 r
    givenLike_whenDeleteLike_thenSta            33 r
    givenLikes_whenGetLikesByImageK             26 r
```

Unit Test



```
MicroServiceLikes (com.example)                 772 ms
  ImageLikeControllerIntegrationTests           704 ms
  ImageLikeControllerUnitTests                  67 ms
    givenReview_whenPutReview_then              23 ms
    givenLikes_whenGetLikesByUserEm             9 ms
    givenLike_whenGetLikeByImageKey             7 ms
    givenNoReview_whenDeleteReview              5 ms
    whenPostReview_thenReturnJsonRe             5 ms
    givenReview_whenDeleteReview_th             4 ms
    givenLike_whenGetLikeByImageKey             6 ms
    givenLikes_whenGetLikesByImageK             8 ms
```

Test Coverage



100% classes, 96% lines covered in package 'com.example.MicroServiceLikes'

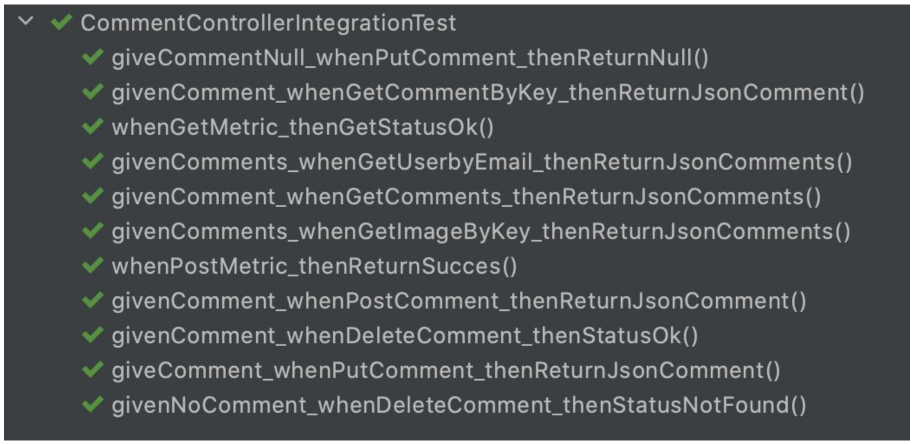
Element	Class, %	Method, %	Line, %
controller	100% (1/1)	100% (8/8)	100% (22/2...
model	100% (1/1)	100% (13/1...	100% (30/3...
repository	100% (0/0)	100% (0/0)	100% (0/0)

Comment

Youri Van Laer

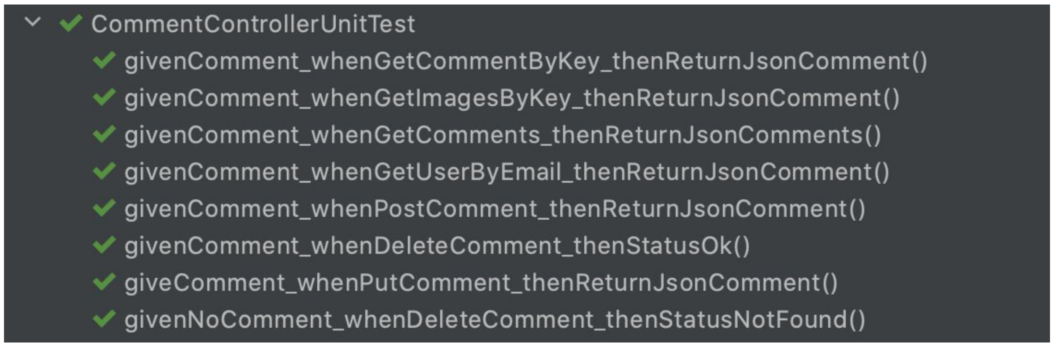
Github: https://github.com/YouriVLTM/microserviceCommentsJava
Docker: docker push yourivanlaer/comment-service-ex

Integration Test



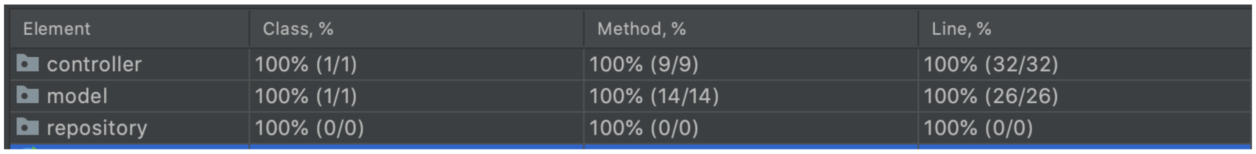
```
CommentControllerIntegrationTest
    giveCommentNull_whenPutComment_thenReturnNull()
    givenComment_whenGetCommentByKey_thenReturnJsonComment()
    whenGetMetric_thenGetStatusOk()
    givenComments_whenGetUserbyEmail_thenReturnJsonComments()
    givenComment_whenGetComments_thenReturnJsonComments()
    givenComments_whenGetImageByKey_thenReturnJsonComments()
    whenPostMetric_thenReturnSucces()
    givenComment_whenPostComment_thenReturnJsonComment()
    givenComment_whenDeleteComment_thenStatusOk()
    giveComment_whenPutComment_thenReturnJsonComment()
    givenNoComment_whenDeleteComment_thenStatusNotFound()
```

Unit Test



```
CommentControllerUnitTest
    givenComment_whenGetCommentByKey_thenReturnJsonComment()
    givenComment_whenGetImagesByKey_thenReturnJsonComment()
    givenComment_whenGetComments_thenReturnJsonComments()
    givenComment_whenGetUserByEmail_thenReturnJsonComment()
    givenComment_whenPostComment_thenReturnJsonComment()
    givenComment_whenDeleteComment_thenStatusOk()
    giveComment_whenPutComment_thenReturnJsonComment()
    givenNoComment_whenDeleteComment_thenStatusNotFound()
```

Test Coverage



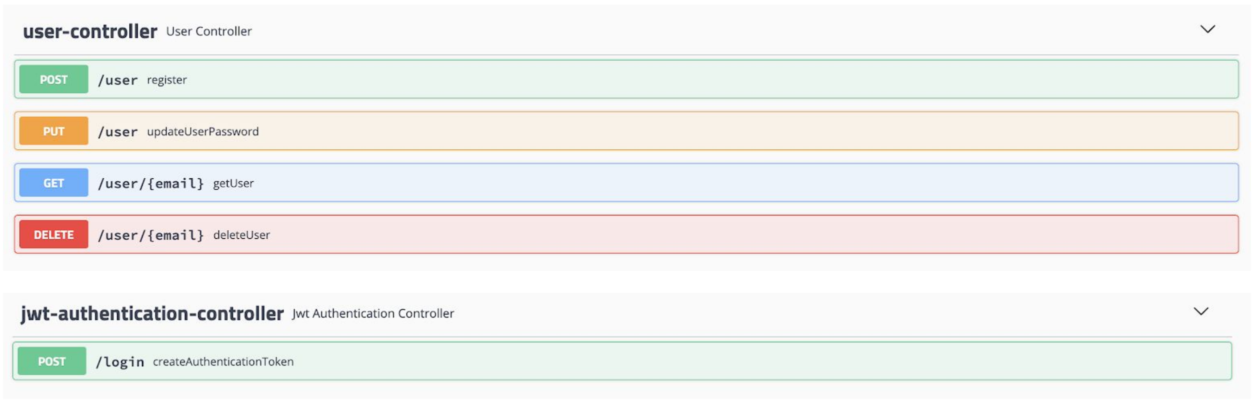
Element	Class, %	Method, %	Line, %
controller	100% (1/1)	100% (9/9)	100% (32/32)
model	100% (1/1)	100% (14/14)	100% (26/26)
repository	100% (0/0)	100% (0/0)	100% (0/0)

EdgeService

User

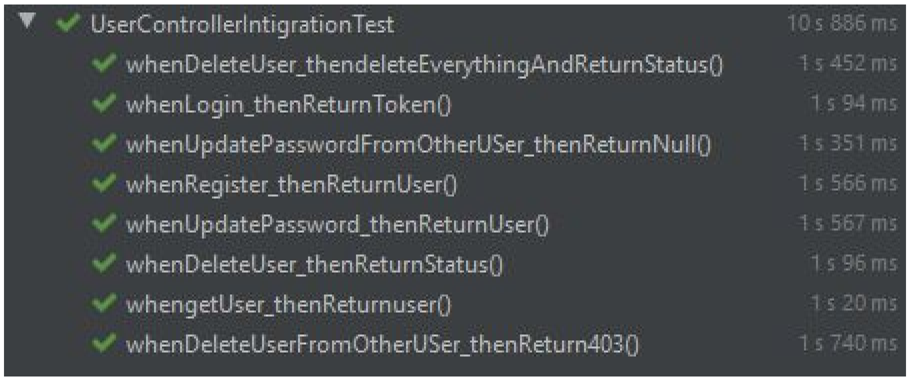
Robin Vranckx

Controller



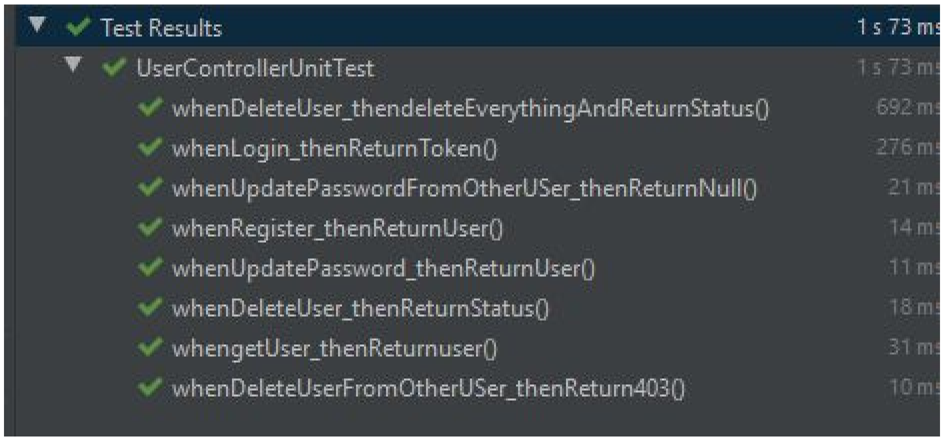
```
user-controller  User Controller
POST    /user  register
PUT     /user  updateUserPassword
GET     /user/{email}  getUser
DELETE  /user/{email}  deleteUser

jwt-authentication-controller  Jwt Authentication Controller
POST    /login  createAuthenticationToken
```

Integration Test



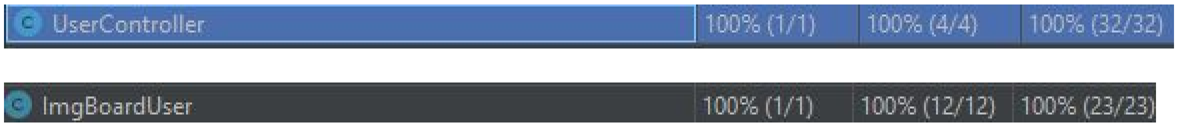
```
UserControllerIntigrationTest                                   10 s 886 ms
    whenDeleteUser_thendeleteEverythingAndReturnStatus()         1 s 452 ms
    whenLogin_thenReturnToken()                                  1 s 94 ms
    whenUpdatePasswordFromOtherUSer_thenReturnNull()             1 s 351 ms
    whenRegister_thenReturnUser()                                1 s 566 ms
    whenUpdatePassword_thenReturnUser()                          1 s 567 ms
    whenDeleteUser_thenReturnStatus()                            1 s 96 ms
    whengetUser_thenReturnuser()                                 1 s 20 ms
    whenDeleteUserFromOtherUSer_thenReturn403()                  1 s 740 ms
```

Unit Test



```
Test Results                                                  1 s 73 ms
  UserControllerUnitTest                                      1 s 73 ms
    whenDeleteUser_thendeleteEverythingAndReturnStatus()      692 ms
    whenLogin_thenReturnToken()                               276 ms
    whenUpdatePasswordFromOtherUSer_thenReturnNull()          21 ms
    whenRegister_thenReturnUser()                             14 ms
    whenUpdatePassword_thenReturnUser()                       11 ms
    whenDeleteUser_thenReturnStatus()                         18 ms
    whengetUser_thenReturnuser()                              31 ms
    whenDeleteUserFromOtherUSer_thenReturn403()               10 ms
```

Test Coverage



UserController	100% (1/1)	100% (4/4)	100% (32/32)
ImgBoardUser	100% (1/1)	100% (12/12)	100% (23/23)

Images

Gust van der Sanden

Controller

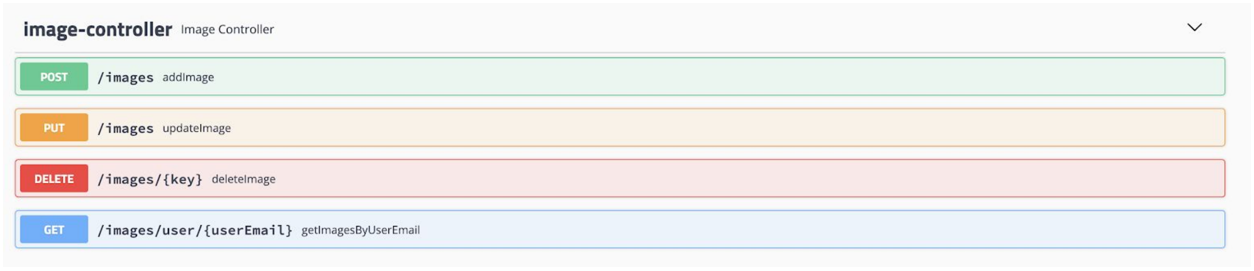
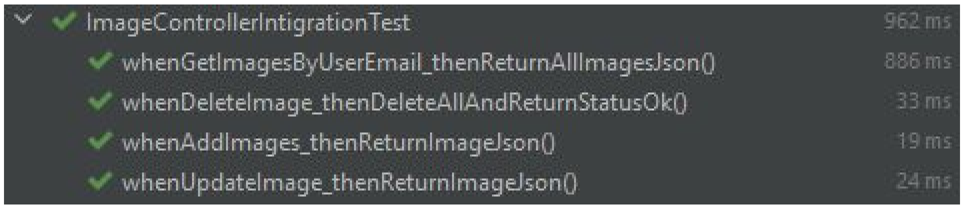


image-controller Image Controller

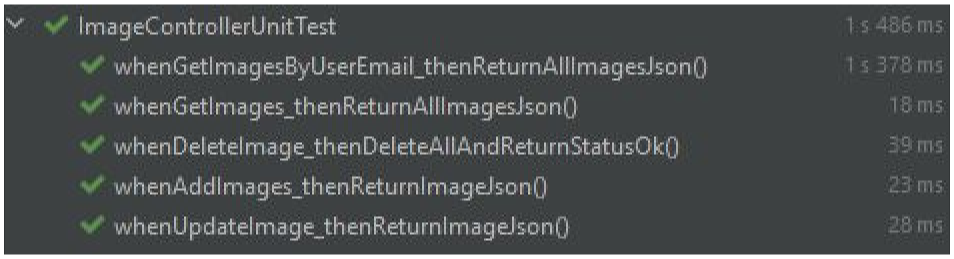
POST	/images	addImage
PUT	/images	updateImage
DELETE	/images/{key}	deleteImage
GET	/images/user/{userEmail}	getImagesByUserEmail

Integration Test



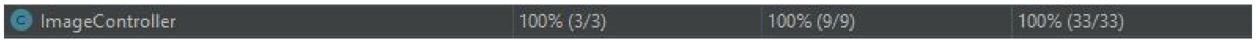
```
ImageControllerIntigrationTest                                962 ms
  whenGetImagesByUserEmail_thenReturnAllImagesJson()          886 ms
  whenDeleteImage_thenDeleteAllAndReturnStatusOk()            33 ms
  whenAddImages_thenReturnImageJson()                         19 ms
  whenUpdateImage_thenReturnImageJson()                       24 ms
```

Unit Test



```
ImageControllerUnitTest                                   1 s 486 ms
    whenGetImagesByUserEmail_thenReturnAllImagesJson()    1 s 378 ms
    whenGetImages_thenReturnAllImagesJson()                   18 ms
    whenDeleteImage_thenDeleteAllAndReturnStatusOk()          39 ms
    whenAddImages_thenReturnImageJson()                       23 ms
    whenUpdateImage_thenReturnImageJson()                     28 ms
```

Test Coverage



```
ImageController      100% (3/3)      100% (9/9)      100% (33/33)
```

Like

Robin Thoelen

Controller

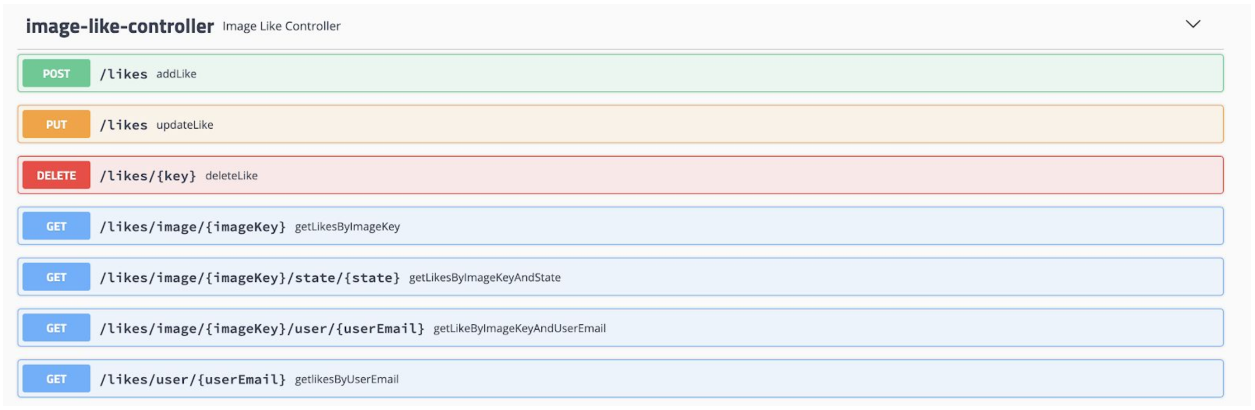
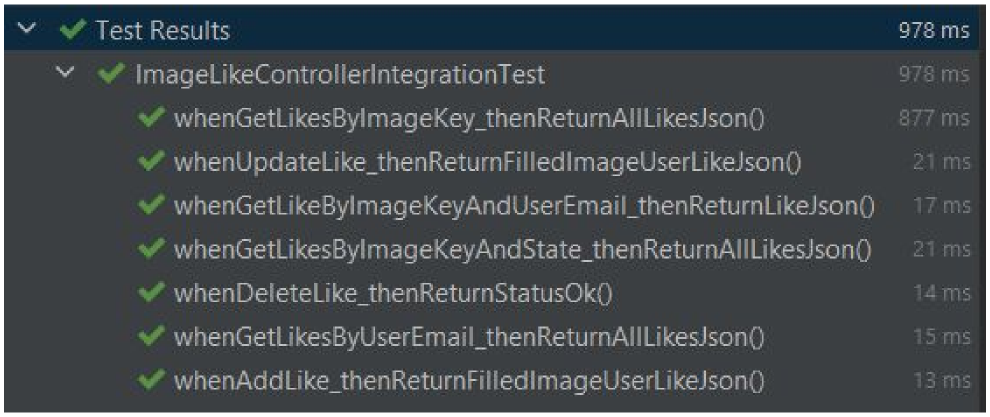


image-like-controller Image Like Controller

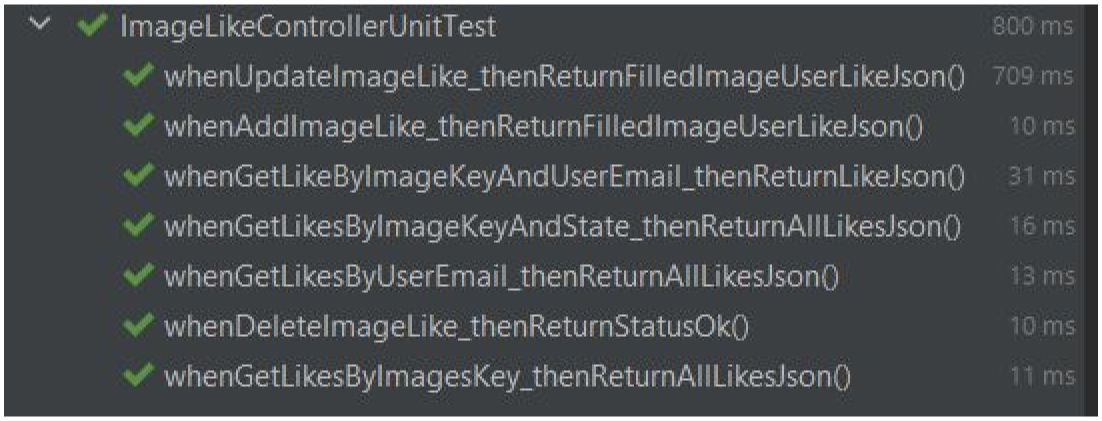
```
POST    /likes  addLike
PUT     /likes  updateLike
DELETE  /likes/{key}  deleteLike
GET     /likes/image/{imageKey}  getLikesByImageKey
GET     /likes/image/{imageKey}/state/{state}  getLikesByImageKeyAndState
GET     /likes/image/{imageKey}/user/{userEmail}  getLikeByImageKeyAndUserEmail
GET     /likes/user/{userEmail}  getlikesByUserEmail
```

Integration Test



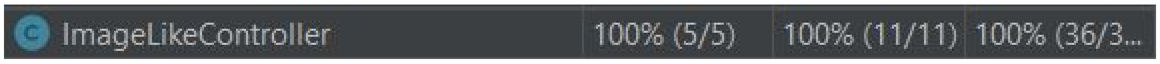
```
Test Results                                                    978 ms
    ImageLikeControllerIntegrationTest                          978 ms
        whenGetLikesByImageKey_thenReturnAllLikesJson()         877 ms
        whenUpdateLike_thenReturnFilledImageUserLikeJson()       21 ms
        whenGetLikeByImageKeyAndUserEmail_thenReturnLikeJson()   17 ms
        whenGetLikesByImageKeyAndState_thenReturnAllLikesJson()  21 ms
        whenDeleteLike_thenReturnStatusOk()                      14 ms
        whenGetLikesByUserEmail_thenReturnAllLikesJson()         15 ms
        whenAddLike_thenReturnFilledImageUserLikeJson()          13 ms
```

Unit Test



```
ImageLikeControllerUnitTest                                   800 ms
  whenUpdateImageLike_thenReturnFilledImageUserLikeJson()      709 ms
  whenAddImageLike_thenReturnFilledImageUserLikeJson()          10 ms
  whenGetLikeByImageKeyAndUserEmail_thenReturnLikeJson()        31 ms
  whenGetLikesByImageKeyAndState_thenReturnAllLikesJson()       16 ms
  whenGetLikesByUserEmail_thenReturnAllLikesJson()              13 ms
  whenDeleteImageLike_thenReturnStatusOk()                      10 ms
  whenGetLikesByImagesKey_thenReturnAllLikesJson()              11 ms
```

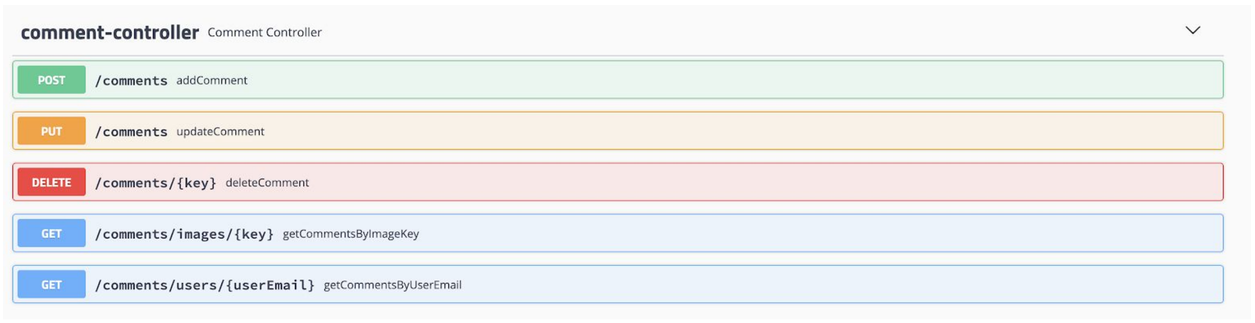
Test Coverage



ImageLikeController	100% (5/5)	100% (11/11)	100% (36/3...

Comment
Youri Van Laer

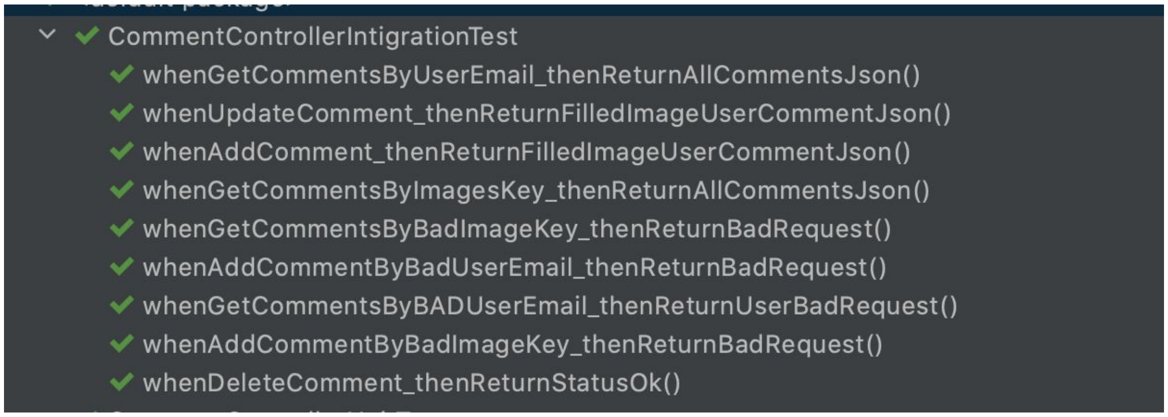
Controller



comment-controller Comment Controller

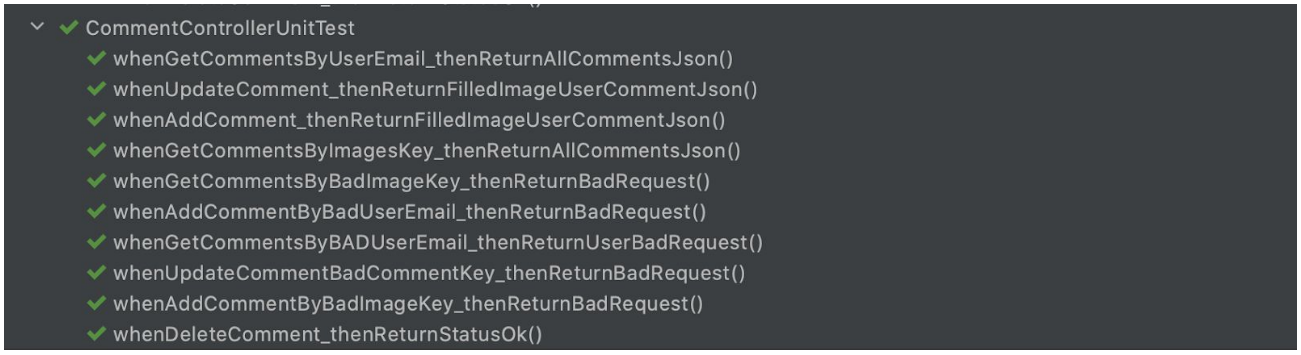
POST	/comments	addComment
PUT	/comments	updateComment
DELETE	/comments/{key}	deleteComment
GET	/comments/images/{key}	getCommentsByImageKey
GET	/comments/users/{userEmail}	getCommentsByUserEmail

Integration Test



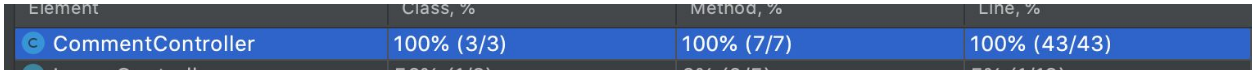
```
✓ CommentControllerIntigrationTest
    ✓ whenGetCommentsByUserEmail_thenReturnAllCommentsJson()
    ✓ whenUpdateComment_thenReturnFilledImageUserCommentJson()
    ✓ whenAddComment_thenReturnFilledImageUserCommentJson()
    ✓ whenGetCommentsByImagesKey_thenReturnAllCommentsJson()
    ✓ whenGetCommentsByBadImageKey_thenReturnBadRequest()
    ✓ whenAddCommentByBadUserEmail_thenReturnBadRequest()
    ✓ whenGetCommentsByBADUserEmail_thenReturnUserBadRequest()
    ✓ whenAddCommentByBadImageKey_thenReturnBadRequest()
    ✓ whenDeleteComment_thenReturnStatusOk()
```

Unit Test



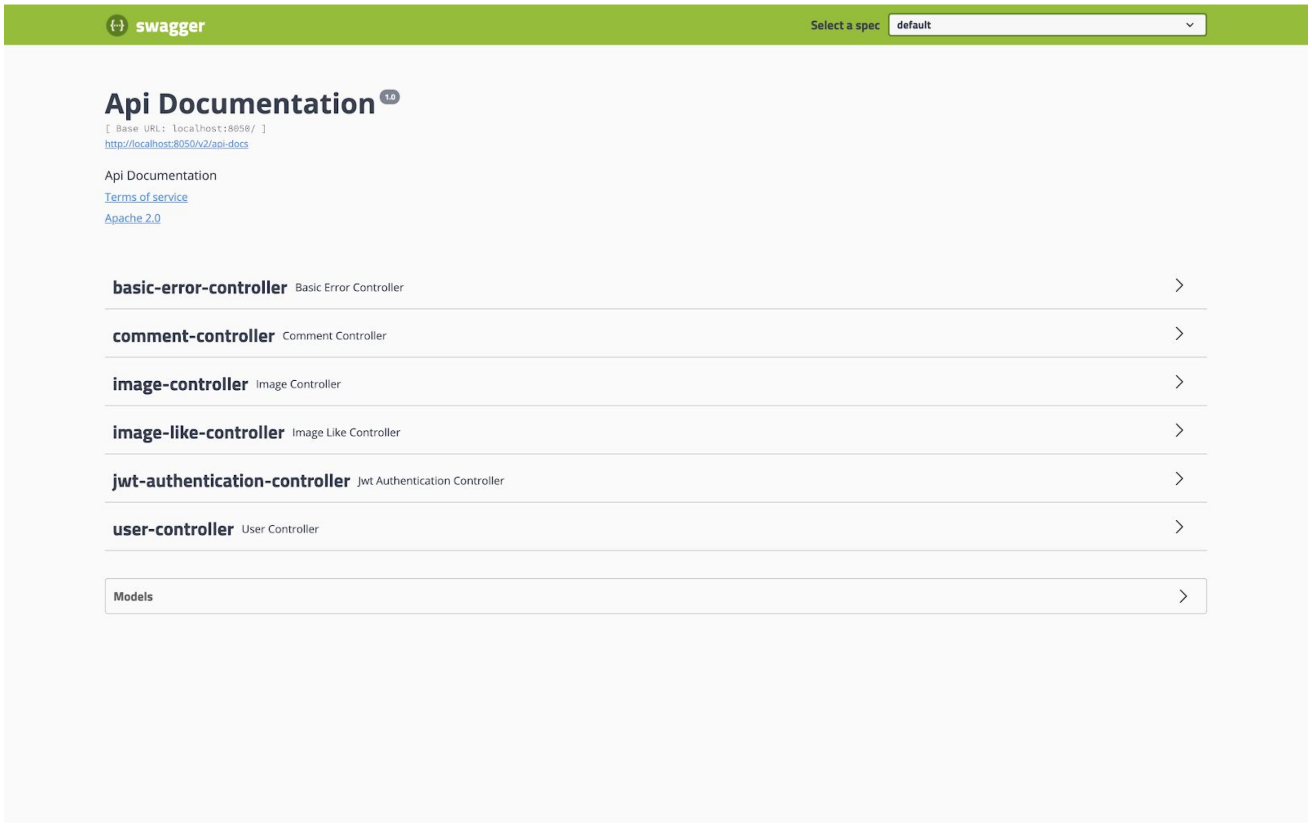
```
✓ CommentControllerUnitTest
    ✓ whenGetCommentsByUserEmail_thenReturnAllCommentsJson()
    ✓ whenUpdateComment_thenReturnFilledImageUserCommentJson()
    ✓ whenAddComment_thenReturnFilledImageUserCommentJson()
    ✓ whenGetCommentsByImagesKey_thenReturnAllCommentsJson()
    ✓ whenGetCommentsByBadImageKey_thenReturnBadRequest()
    ✓ whenAddCommentByBadUserEmail_thenReturnBadRequest()
    ✓ whenGetCommentsByBADUserEmail_thenReturnUserBadRequest()
    ✓ whenUpdateCommentBadCommentKey_thenReturnBadRequest()
    ✓ whenAddCommentByBadImageKey_thenReturnBadRequest()
    ✓ whenDeleteComment_thenReturnStatusOk()
```

Test Coverage



Element	Class, %	Method, %	Line, %
CommentController	100% (3/3)	100% (7/7)	100% (43/43)

SwaggerUI



Api Documentation 1.0

[Base URL: localhost:8050/]

http://localhost:8050/v2/api-docs

Api Documentation

Terms of service

Apache 2.0

basic-error-controller Basic Error Controller

comment-controller Comment Controller

image-controller Image Controller

image-like-controller Image Like Controller

jwt-authentication-controller Jwt Authentication Controller

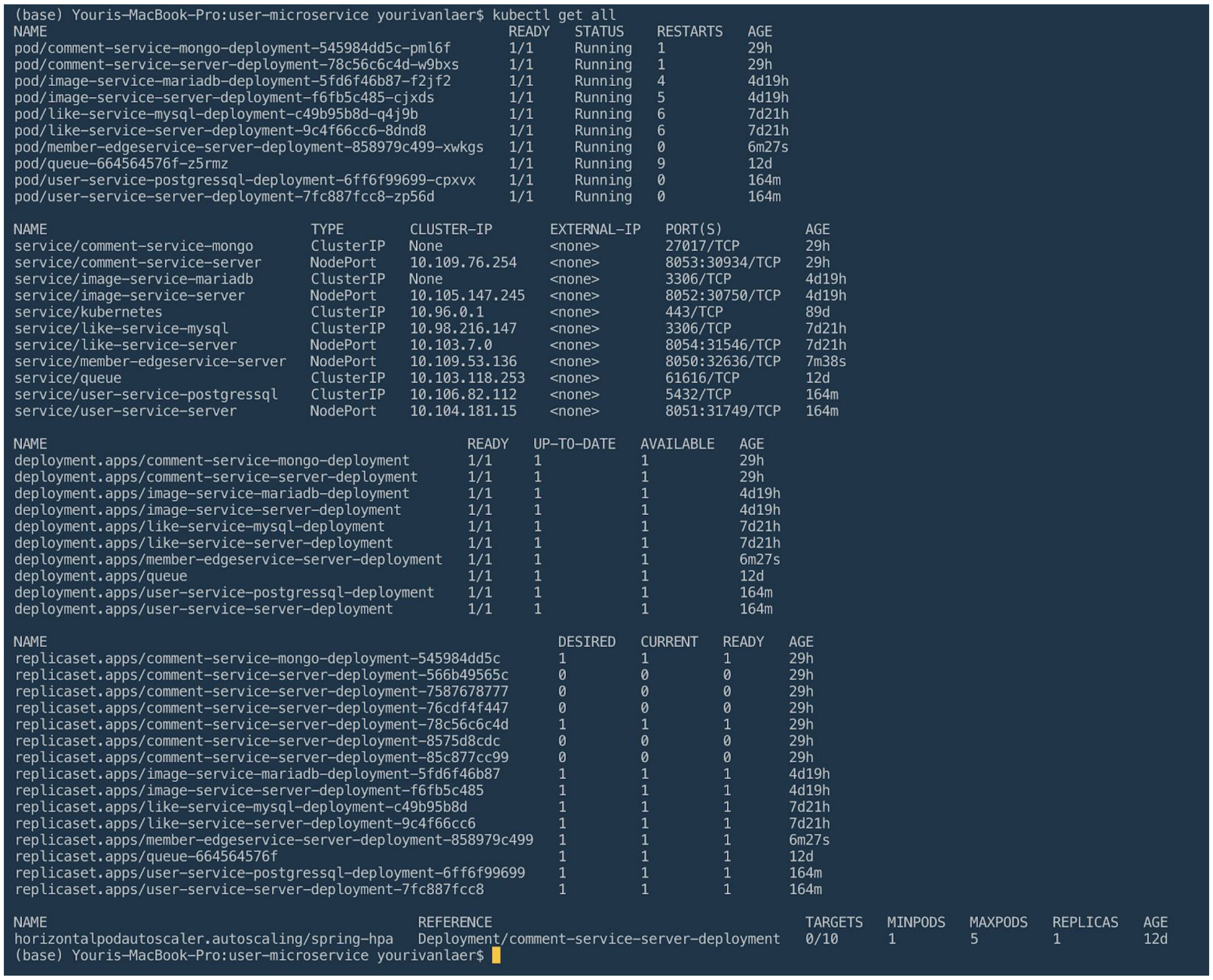
user-controller User Controller

Models

Extra's

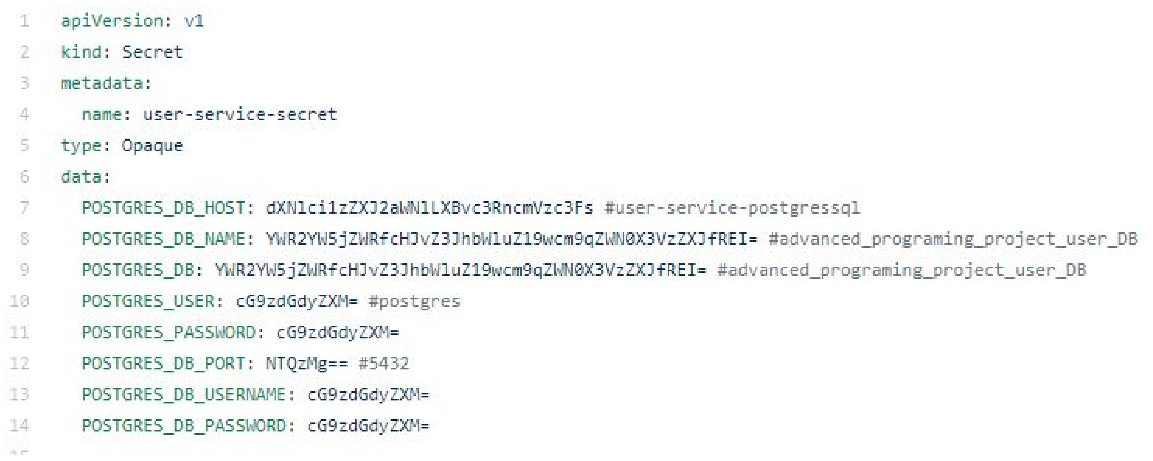
K8f

(na dev. microservice zijn veranderd naar ClusterIP)



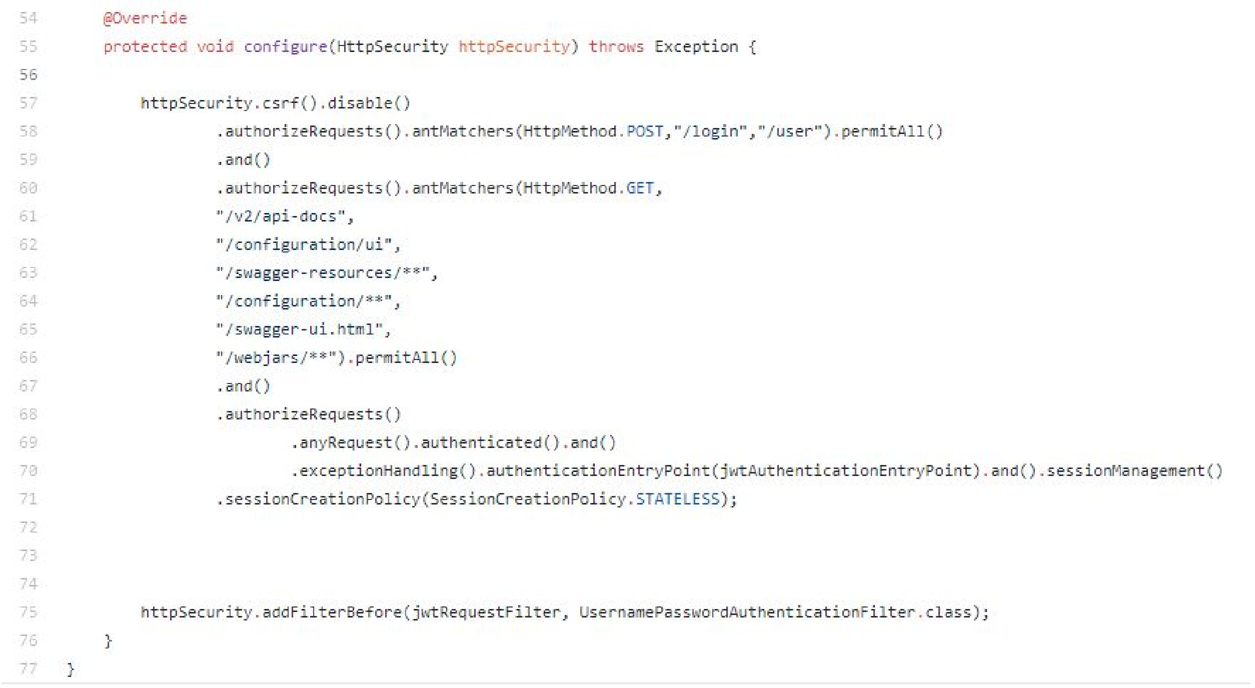
```
(base) Youris-MacBook-Pro:user-microservice yourivanlaer$ kubectl get all
NAME                                                    READY   STATUS    RESTARTS   AGE
pod/comment-service-mongo-deployment-545984dd5c-pml6f    1/1     Running   1          29h
pod/comment-service-server-deployment-78c56c6c4d-w9bxs   1/1     Running   1          29h
pod/image-service-mariadb-deployment-5fd6f46b87-f2jf2    1/1     Running   4          4d19h
pod/image-service-server-deployment-f6fb5c485-cjxds      1/1     Running   5          4d19h
pod/like-service-mysql-deployment-c49b95b8d-q4j9b        1/1     Running   6          7d21h
pod/like-service-server-deployment-9c4f66cc6-8dnd8       1/1     Running   6          7d21h
pod/member-edgeservice-server-deployment-858979c499-xwkgs  1/1     Running   0          6m27s
pod/queue-664564576f-z5rmz                               1/1     Running   9          12d
pod/user-service-postgressql-deployment-6ff6f99699-cpxvx  1/1     Running   0          164m
pod/user-service-server-deployment-7fc887fcc8-zp56d      1/1     Running   0          164m

NAME                                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/comment-service-mongo        ClusterIP   None             <none>        27017/TCP        29h
service/comment-service-server       NodePort    10.109.76.254    <none>        8053:30934/TCP   29h
service/image-service-mariadb        ClusterIP   None             <none>        3306/TCP         4d19h
service/image-service-server         NodePort    10.105.147.245   <none>        8052:30750/TCP   4d19h
service/kubernetes                   ClusterIP   10.96.0.1        <none>        443/TCP          89d
service/like-service-mysql           ClusterIP   10.98.216.147    <none>        3306/TCP         7d21h
service/like-service-server          NodePort    10.103.7.0       <none>        8054:31546/TCP   7d21h
service/member-edgeservice-server    NodePort    10.109.53.136    <none>        8050:32636/TCP   7m38s
service/queue                        ClusterIP   10.103.118.253   <none>        61616/TCP        12d
service/user-service-postgressql     ClusterIP   10.106.82.112    <none>        5432/TCP         164m
service/user-service-server          NodePort    10.104.181.15    <none>        8051:31749/TCP   164m

NAME                                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/comment-service-mongo-deployment      1/1     1            1           29h
deployment.apps/comment-service-server-deployment     1/1     1            1           29h
deployment.apps/image-service-mariadb-deployment      1/1     1            1           4d19h
deployment.apps/image-service-server-deployment       1/1     1            1           4d19h
deployment.apps/like-service-mysql-deployment         1/1     1            1           7d21h
deployment.apps/like-service-server-deployment        1/1     1            1           7d21h
deployment.apps/member-edgeservice-server-deployment  1/1     1            1           6m27s
deployment.apps/queue                                 1/1     1            1           12d
deployment.apps/user-service-postgressql-deployment   1/1     1            1           164m
deployment.apps/user-service-server-deployment        1/1     1            1           164m

NAME                                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/comment-service-mongo-deployment-545984dd5c     1         1         1       29h
replicaset.apps/comment-service-server-deployment-566b49565c    0         0         0       29h
replicaset.apps/comment-service-server-deployment-7587678777    0         0         0       29h
replicaset.apps/comment-service-server-deployment-76cdf4f447    0         0         0       29h
replicaset.apps/comment-service-server-deployment-78c56c6c4d    1         1         1       29h
replicaset.apps/comment-service-server-deployment-8575d8cdc     0         0         0       29h
replicaset.apps/comment-service-server-deployment-85c877cc99    0         0         0       29h
replicaset.apps/image-service-mariadb-deployment-5fd6f46b87     1         1         1       4d19h
replicaset.apps/image-service-server-deployment-f6fb5c485       1         1         1       4d19h
replicaset.apps/like-service-mysql-deployment-c49b95b8d         1         1         1       7d21h
replicaset.apps/like-service-server-deployment-9c4f66cc6        1         1         1       7d21h
replicaset.apps/member-edgeservice-server-deployment-858979c499  1         1         1       6m27s
replicaset.apps/queue-664564576f                                1         1         1       12d
replicaset.apps/user-service-postgressql-deployment-6ff6f99699  1         1         1       164m
replicaset.apps/user-service-server-deployment-7fc887fcc8       1         1         1       164m

NAME                                          REFERENCE                                      TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/spring-hpa   Deployment/comment-service-server-deployment   0/10      1         5         1          12d
(base) Youris-MacBook-Pro:user-microservice yourivanlaer$ 
```

K8-secrets



```yaml
apiVersion: v1
kind: Secret
metadata:
  name: user-service-secret
type: Opaque
data:
  POSTGRES_DB_HOST: dXNlci1zZXJ2aWNlLXBvc3RncmVzc3Fs #user-service-postgressql
  POSTGRES_DB_NAME: YWR2YW5jZWRfcHJvZ3JhbWluZ19wcm9qZWN0X3VzZXJfREI= #advanced_programing_project_user_DB
  POSTGRES_DB: YWR2YW5jZWRfcHJvZ3JhbWluZ19wcm9qZWN0X3VzZXJfREI= #advanced_programing_project_user_DB
  POSTGRES_USER: cG9zdGdyZXM= #postgres
  POSTGRES_PASSWORD: cG9zdGdyZXM=
  POSTGRES_DB_PORT: NTQzMg== #5432
  POSTGRES_DB_USERNAME: cG9zdGdyZXM=
  POSTGRES_DB_PASSWORD: cG9zdGdyZXM=
```

JWT-beveiliging

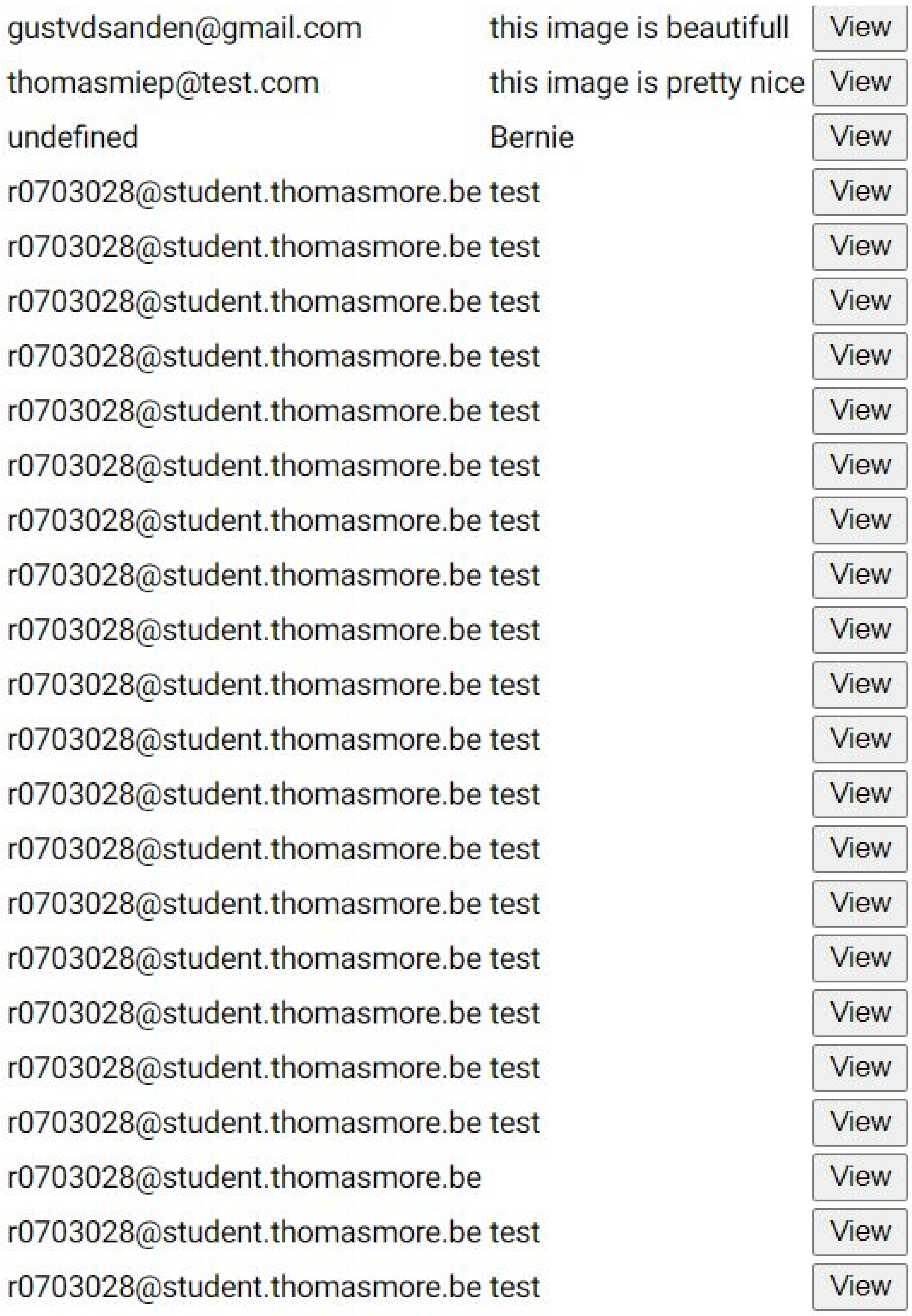


```java
    @Override
    protected void configure(HttpSecurity httpSecurity) throws Exception {

        httpSecurity.csrf().disable()
                .authorizeRequests().antMatchers(HttpMethod.POST,"/login","/user").permitAll()
                .and()
                .authorizeRequests().antMatchers(HttpMethod.GET,
                "/v2/api-docs",
                "/configuration/ui",
                "/swagger-resources/**",
                "/configuration/**",
                "/swagger-ui.html",
                "/webjars/**").permitAll()
                .and()
                .authorizeRequests()
                        .anyRequest().authenticated().and()
                        .exceptionHandling().authenticationEntryPoint(jwtAuthenticationEntryPoint).and().sessionManagement()
                .sessionCreationPolicy(SessionCreationPolicy.STATELESS);



        httpSecurity.addFilterBefore(jwtRequestFilter, UsernamePasswordAuthenticationFilter.class);
    }
}
```

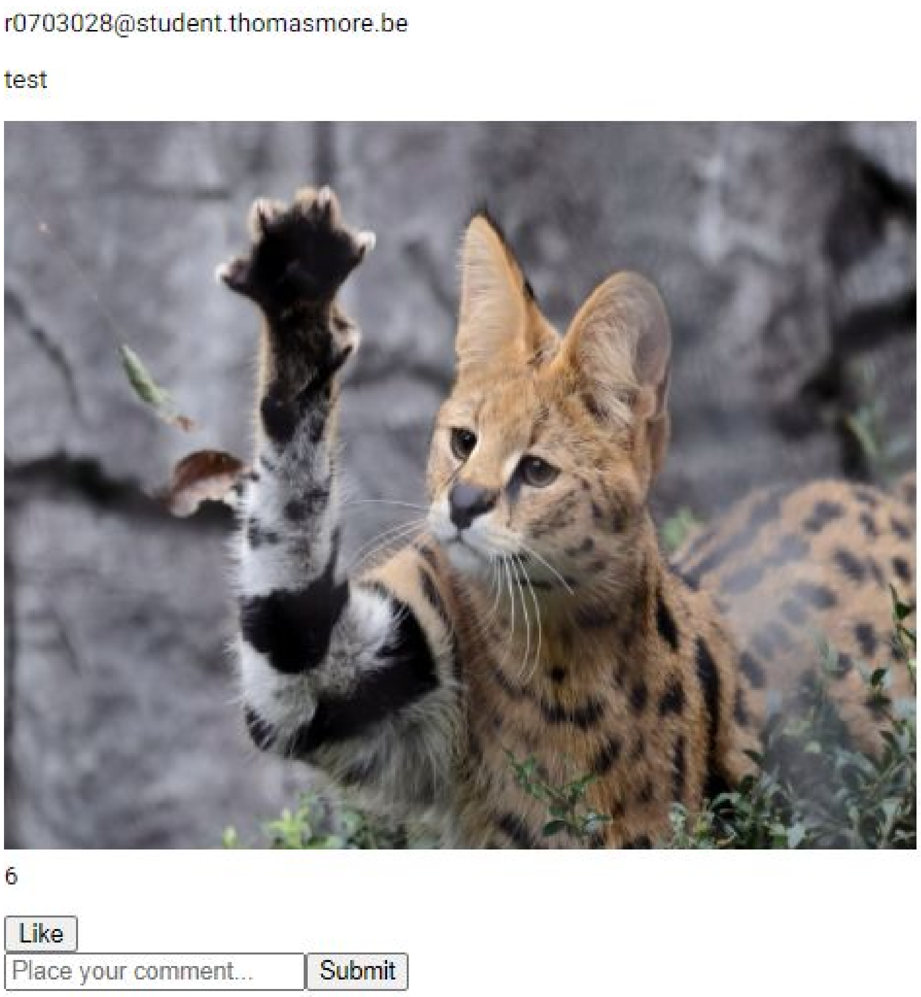
front end

Repo: https://github.com/RoboticRobin20/ImbBoardFrontEnd

Angular



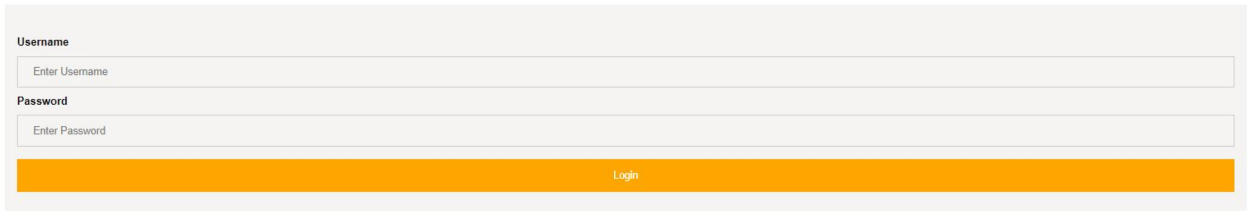
gustvdsanden@gmail.com	this image is beautifull	View
thomasmiep@test.com	this image is pretty nice	View
undefined	Bernie	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be		View
r0703028@student.thomasmore.be	test	View
r0703028@student.thomasmore.be	test	View



r0703028@student.thomasmore.be

test

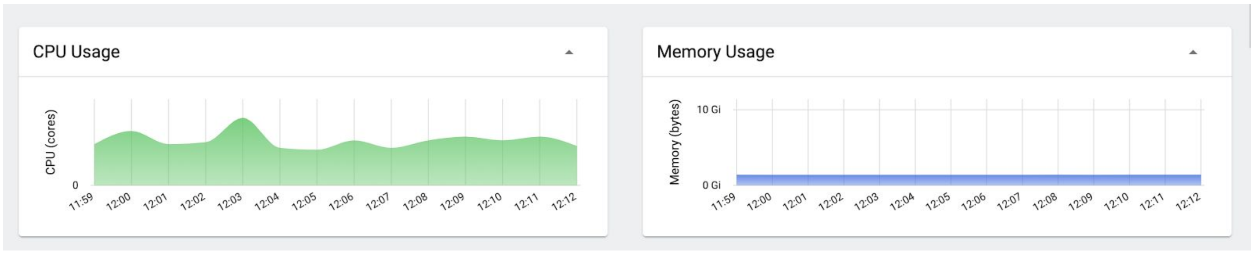
6



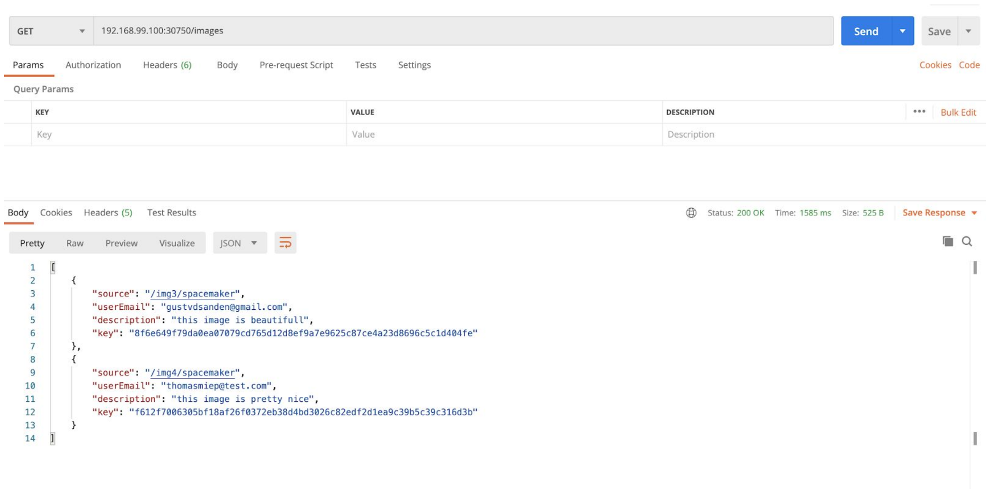
ActiveMQ

```
deployment.apps/queue                                    1/1     1            1            11d
```

Prometheus



Postman


GET 192.168.99.100:30750/images

Params | Authorization | Headers (6) | Body | Pre-request Script | Tests | Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

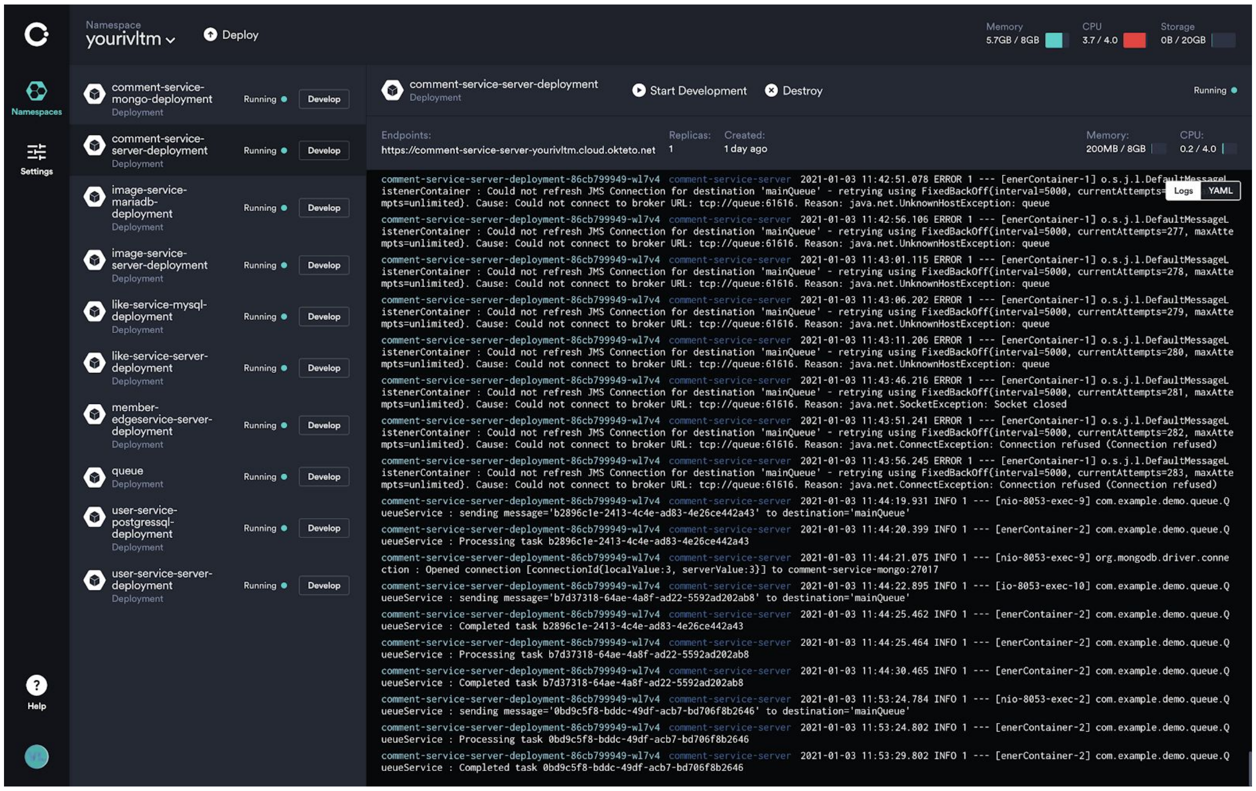
Body | Cookies | Headers (5) | Test Results — Status: 200 OK Time: 1585 ms Size: 525 B

```
[
    {
        "source": "/img3/spacemaker",
        "userEmail": "gustvdsanden@gmail.com",
        "description": "this image is beautifull",
        "key": "8f6e649f79da0ea07079cd765d12d8ef9a7e9625c87ce4a23d8696c5c1d404fe"
    },
    {
        "source": "/img4/spacemaker",
        "userEmail": "thomasmiep@test.com",
        "description": "this image is pretty nice",
        "key": "f612f7006305bf18af26f0372eb38d4bd3026c82edf2d1ea9c39b5c39c316d3b"
    }
]
```

Okteto

Okteto: https://cloud.okteto.com/#/spaces/yourivltm

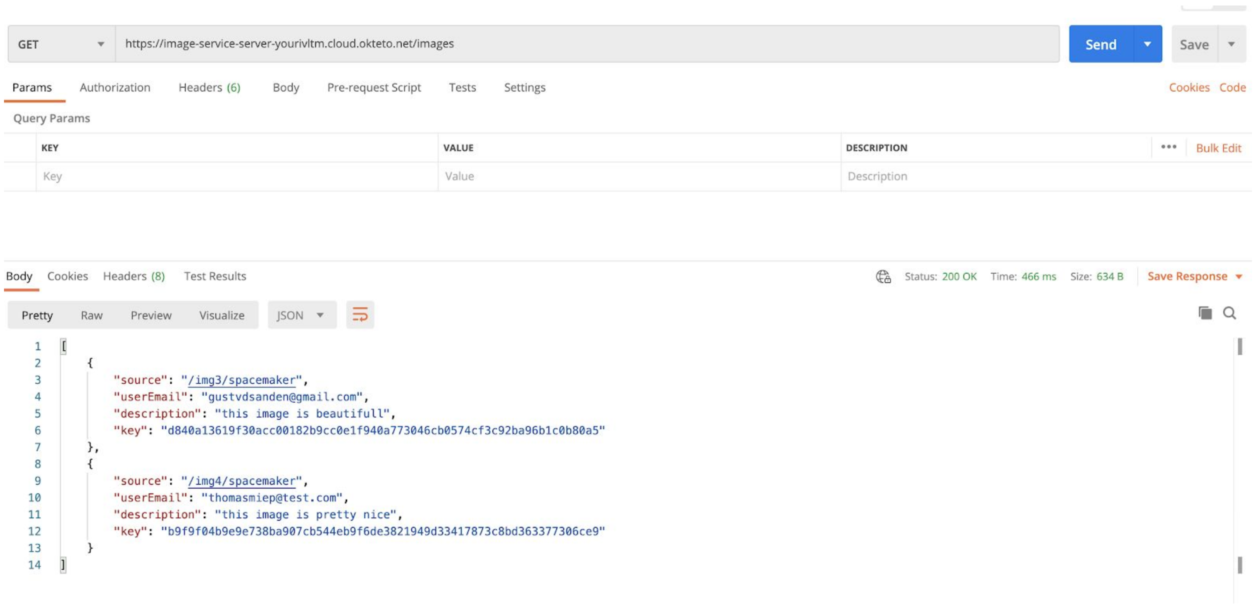
Dit is een online gratis platform waarbij u de yml zeer eenvoudig kan deployen.
Zie hier een voorbeeld van alle microservices:


Namespace
yourivltm
Deploy
Memory 5.7GB / 8GB
CPU 3.7 / 4.0
Storage 0B / 20GB
Namespaces
Settings
Help
comment-service-mongo-deployment Deployment Running Develop
comment-service-server-deployment Deployment Running Develop
image-service-mariadb-deployment Deployment Running Develop
image-service-server-deployment Deployment Running Develop
like-service-mysql-deployment Deployment Running Develop
like-service-server-deployment Deployment Running Develop
member-edgeservice-server-deployment Deployment Running Develop
queue Deployment Running Develop
user-service-postgressql-deployment Deployment Running Develop
user-service-server-deployment Deployment Running Develop
comment-service-server-deployment
Deployment
Start Development
Destroy
Running
Logs
YAML

Endpoints: https://comment-service-server-yourivltm.cloud.okteto.net
Replicas: 1
Created: 1 day ago
Memory: 200MB / 8GB
CPU: 0.2 / 4.0

```
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:42:51.078 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.UnknownHostException: queue
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:42:56.106 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=277, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.UnknownHostException: queue
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:43:01.115 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=278, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.UnknownHostException: queue
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:43:06.202 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=279, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.UnknownHostException: queue
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:43:11.206 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=280, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.UnknownHostException: queue
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:43:46.216 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=281, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.SocketException: Socket closed
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:43:51.241 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=282, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.ConnectException: Connection refused (Connection refused)
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:43:56.245 ERROR 1 --- [enerContainer-1] o.s.j.l.DefaultMessageL
istenerContainer : Could not refresh JMS Connection for destination 'mainQueue' - retrying using FixedBackOff{interval=5000, currentAttempts=283, maxAtte
mpts=unlimited}. Cause: Could not connect to broker URL: tcp://queue:61616. Reason: java.net.ConnectException: Connection refused (Connection refused)
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:19.931 INFO 1 --- [nio-8053-exec-9] com.example.demo.queue.Q
ueueService : sending message='b2896c1e-2413-4c4e-ad83-4e26ce442a43' to destination='mainQueue'
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:20.399 INFO 1 --- [enerContainer-2] com.example.demo.queue.Q
ueueService : Processing task b2896c1e-2413-4c4e-ad83-4e26ce442a43
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:21.075 INFO 1 --- [nio-8053-exec-9] org.mongodb.driver.conne
ction : Opened connection [connectionId{localValue:3, serverValue:3}] to comment-service-mongo:27017
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:22.895 INFO 1 --- [io-8053-exec-10] com.example.demo.queue.Q
ueueService : sending message='b7d37318-64ae-4a8f-ad22-5592ad202ab8' to destination='mainQueue'
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:25.462 INFO 1 --- [enerContainer-2] com.example.demo.queue.Q
ueueService : Completed task b2896c1e-2413-4c4e-ad83-4e26ce442a43
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:25.464 INFO 1 --- [enerContainer-2] com.example.demo.queue.Q
ueueService : Processing task b7d37318-64ae-4a8f-ad22-5592ad202ab8
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:44:30.465 INFO 1 --- [enerContainer-2] com.example.demo.queue.Q
ueueService : Completed task b7d37318-64ae-4a8f-ad22-5592ad202ab8
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:53:24.784 INFO 1 --- [nio-8053-exec-2] com.example.demo.queue.Q
ueueService : sending message='0bd9c5f8-bddc-49df-acb7-bd706f8b2646' to destination='mainQueue'
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:53:24.802 INFO 1 --- [enerContainer-2] com.example.demo.queue.Q
ueueService : Processing task 0bd9c5f8-bddc-49df-acb7-bd706f8b2646
comment-service-server-deployment-86cb799949-wl7v4 comment-service-server 2021-01-03 11:53:29.802 INFO 1 --- [enerContainer-2] com.example.demo.queue.Q
ueueService : Completed task 0bd9c5f8-bddc-49df-acb7-bd706f8b2646
```

Op onderstaande afbeelding ziet u een eenvoudig een GET request:



GET https://image-service-server-yourivltm.cloud.okteto.net/images

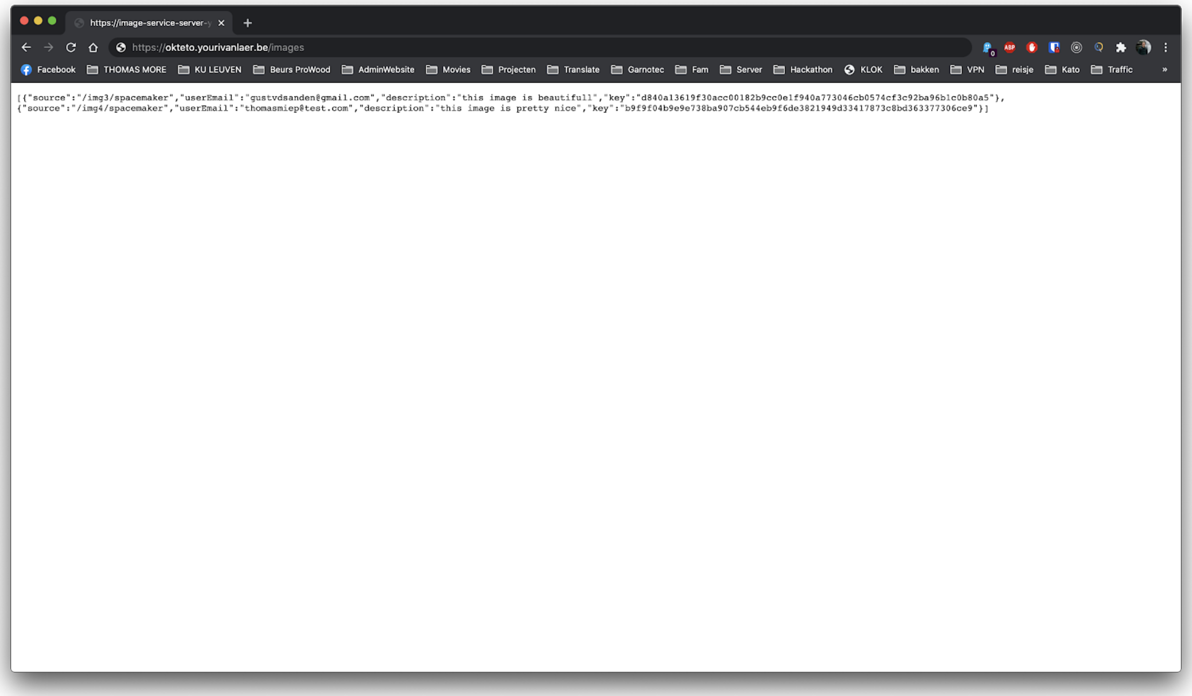
Status: 200 OK Time: 466 ms Size: 634 B

```
[
    {
        "source": "/img3/spacemaker",
        "userEmail": "gustvdsanden@gmail.com",
        "description": "this image is beautifull",
        "key": "d840a13619f30acc00182b9cc0e1f940a773046cb0574cf3c92ba96b1c0b80a5"
    },
    {
        "source": "/img4/spacemaker",
        "userEmail": "thomasmiep@test.com",
        "description": "this image is pretty nice",
        "key": "b9f9f04b9e9e738ba907cb544eb9f6de3821949d33417873c8bd363377306ce9"
    }
]
```

Domein

Deze url wordt gegenereerd door Okteto zelf. Maar u kan ook instellen op uw eigen domein naam. Waarbij we de DNS settings veranderen. Zie onderstaande afbeelding: **okteto.yourivanlaer.be**

Helaas is deze een betalend functie bij Okteto. We hebben hier een proefversie genomen van 14 dagen om dit toch te kunnen laten demonstreren.



```
[{"source":"/img3/spacemaker","userEmail":"gustvdsanden@gmail.com","description":"this image is beautifull","key":"d840a13619f30acc00182b9cc0e1f940a773046cb0574cf3c92ba96b1c0b80a5"},
{"source":"/img4/spacemaker","userEmail":"thomasmiep@test.com","description":"this image is pretty nice","key":"b9f9f04b9e9e738ba907cb544eb9f6de3821949d33417873c8bd363377306ce9"}]
```

Besluit

- ✓ Minimum aantal ‘Back-end’ microservices: 4 voor team met 4 personen / 3 voor teams met minder dan 4 personen
- ✓ 1 Edge microservice
- ✓ Dockerfile voor elke microservice
- ✓ Korte beschrijving van het gekozen thema + Diagram van de volledige microservices architectuur (met links naar ‘back-end’ repo’s) op GitHub README.md van de Edge microservice
- ✓ Aantoonbare werking totale architectuur door Postman requests op de Edge microservice (buiten containers)
- ✓ Volledige implementatie SwaggerUI voor de Edge microservice en screenshot(s) van de output op GitHub README.md

REST API

- ✓ Minstens 4 GET endpoints op de Edge microservice, nooit zoekende op DB id
- ✓ POST, PUT en DELETE endpoints op de Edge microservice
- ✓ Minstens 2 GET endpoints per ‘Back-end’ microservice, nooit zoekende op DB id
- ✓ POST, PUT en DELETE endpoints voor minstens 1 ‘Back-end’ microservice
- ✓ Gebruik van PostgreSQL en MongoDB
- ✓ Efficiënt gebruik @PathVariable vs. @RequestParam

TESTING

- ✓ Unit tests voor alle microservice controllers
- ✓ Integration tests alle microservice controllers
- ✓ 100% method test code coverage voor controllers, repositories en constructors van model classes

CI/CD

✓ Elke GitHub repo heeft een CI/CD pipeline die tests runt, de .jar upload als artifact en een Docker container naar Docker Hub pushed

1. BIJKOMENDE COMPONENTEN

✓ Deployment op K8s (+15%)

└→ ✓ K8s deployment op andere server (Niet lokaal maar via: Okteto) (nog te bepalen)

✓ └→ Gebruik van K8s secrets voor environment variables bij deployment (+5%)

☐ └→ Keycloak integratie (+15%)

✓ └→ Queue met bv. ActiveMQ (+10%)

✓ └→ Prometheus auto-scaling via een extra endpoint (+15%)

✓ Basic front-end dat communiceert met de edge-service (+15%)

☐

└→ Selenium testing van het front-end (+10%)