

Κατανεμημένα Συστήματα

Παπαγρηγορίου Αλέξανδρος: it2023054

Ντόκου Ασημίνα: it2023121

Πεταλά Όλγα: it2023058

Περίληψη

Το My City Gov είναι ένας τρόπος online επικοινωνίας των πολιτών με τον δήμο τους. Η εφαρμογή έχει 3 τύπους χρηστών, τους πολίτες, τους υπαλλήλους και τους διαχειριστές. Κάθε χρήστης μπορεί να αξιοποιήσει διαφορετικές λειτουργίες ανάλογα τον τύπο χρήστη.

Οι πολίτες μπορούν να δημιουργήσουν αιτήματα και να κλείσουν ραντεβού με συγκεκριμένες υπηρεσίες του δήμου.

Οι υπάλληλοι μπορούν να επεξεργαστούν αιτήματα και να αποδεχτούν ραντεβού των υπηρεσιών τους.

Οι διαχειριστές μπορούν να αναθέσουν αιτήματα και ραντεβού σε συγκεκριμένες υπηρεσίες και να δουν στατιστικά στοιχεία της εφαρμογής.

Κώδικας

full-gov repository:

<https://github.com/it2023054/full-gov.git>

Projects inside full-gov:

- my-city-gov
- noc-gov
- noc-sms

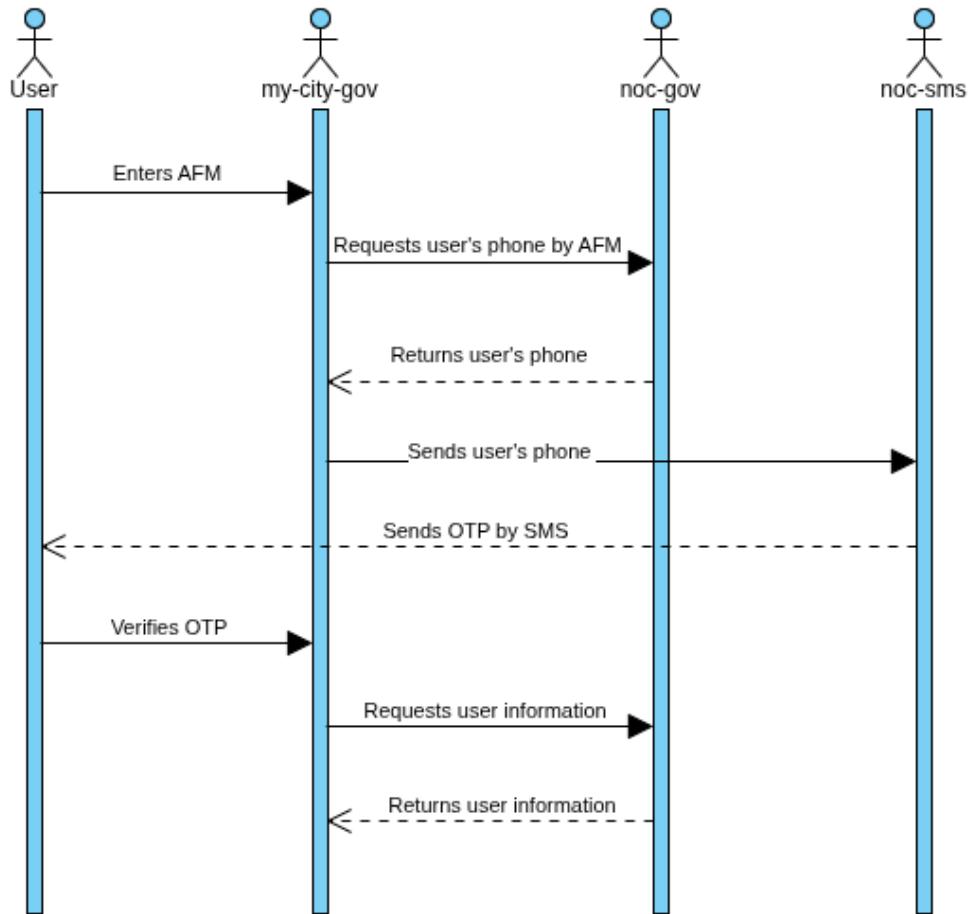
Οδηγίες εκτέλεσης

Και τα 3 projects στο full-gov περιέχουν το δικό τους README αρχείο με τις κατάλληλες οδηγίες για να εκτελεστού.

Οδηγίες χρήσης

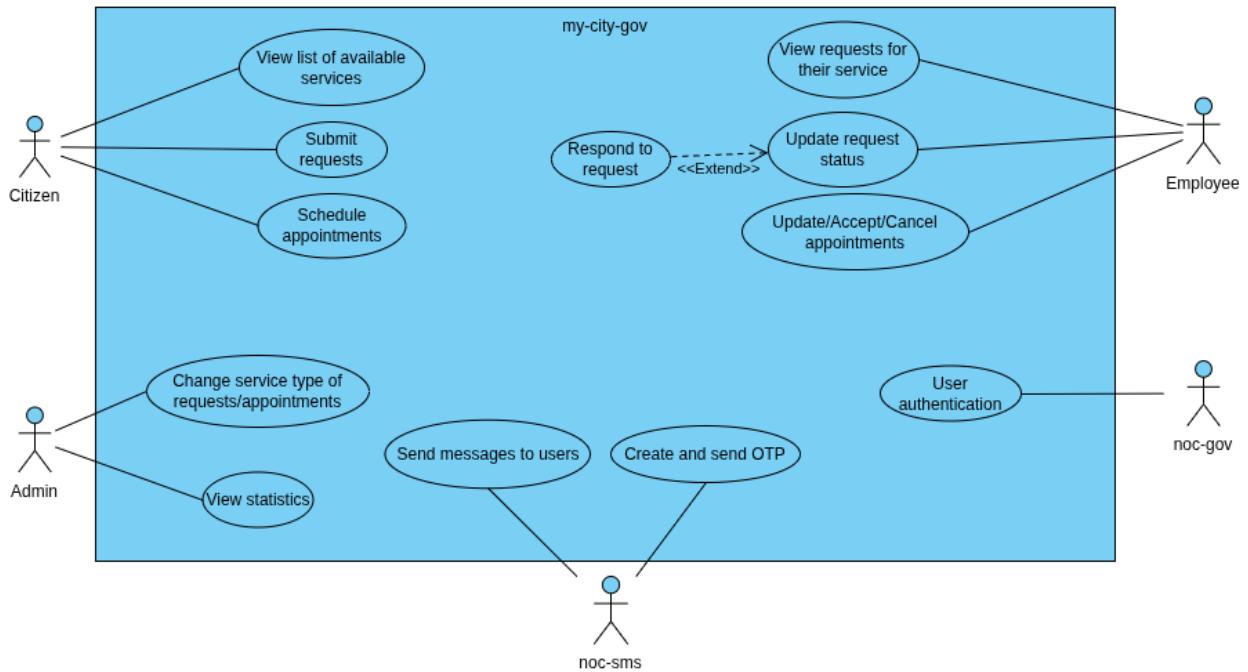
Για να κάνετε login στην εφαρμογή my-city-gov θα πρέπει να δοθεί ένα έγκυρο ΑΦΜ ενός χρήστη της υπηρεσίας noc-gov. Για να δείτε όλους τους χρήστες της υπηρεσίας noc-gov μπορείτε να καλέσετε με μέθοδο get το <http://localhost:8081/api/v1/person>. Για ευκολία μπορείτε να καλέσετε αυτό το αρι μέσω του swagger στο <http://localhost:8081/swagger-ui.html>. Αφού δείτε τα ΑΦΜ όλων των χρηστών και διαλέξετε με ποιο θέλετε να κάνετε login, θα σταλθεί ένα otp μέσω sms στον αριθμό του χρήστη με το ΑΦΜ που δώσατε. Επειδή δεν έχουμε πρόσβαση στο τηλέφωνο που έχουμε αντιστοιχίσει σε κάθε χρήστη, κάνουμε log το otp στο noc-sms για να μπορούμε να κάνουμε login. Αφού δώσουμε το otp στην εφαρμογή, το noc-gov μας επιστρέφει όλα τα στοιχεία του χρήστη και κάνουμε login με επιτυχία.

Sequence διάγραμμα για login:



Αφού κάνουμε login, ανάλογα τον τύπο χρήστη που διαλέξαμε έχουμε και διαφορετικό μενού με διαφορετικές λειτουργίες.

Use case διάγραμμα με λειτουργίες χρηστών:



Αρχιτεκτονική

Layers:

1. Presentation Layer (packages = templates/, web/ui/, web/rest/):

Επικοινωνία με χρήστες μέσω UI και πελάτες μέσω REST API με την βοήθεια των Controllers και RestControllers.

2. Application Layer (packages = core/model/, core/service/, core/util/):

Κεντρική επιχειρησιακή λογική, τα models, τα services και τα utilities.

3. Port Layer (packages = core/repository/, core/port/):

Interfaces από repositories και external services.

4. Config Layer (packages = config/, nocgov/, nocsms/):

Configuration εφαρμογής και εξωτερικών υπηρεσιών.

External Services:

Noc GOV -> Person data

Noc SMS -> Αποστολή OTP / μηνυμάτων

Minio -> Αποθήκευση αρχείων (τρέχει σε container)

JWT Authentication -> Client authentication

Swagger -> REST endpoints

Request Flow:

1. Χρήστης ή client κάνει request στο UI ή REST API.
2. Controller δέχεται request αφού ελέγξει authentication.
3. Η εφαρμογή χρησιμοποιεί την επιχειρησιακή λογική μέσω των services.
4. Τα services χρησιμοποιούν repositories και external services.
5. Επιστρέφει response στον controller (UI ή JSON response).

Χειρισμός σφαλμάτων

Τα περισσότερα exceptions που γίνονται throw από τα services τα κάνουν catch οι controllers και επιστρέφουν την κατάλληλα αλλαγή στο template σαν error για να ενημερώθει ο χρήστης ή τα κάνουν catch οι REST controllers και επιστρέφουν το κατάλληλο JSON response. Υπάρχουν και exceptions που δεν κάνουμε catch επειδή είμαστε ευχαριστημένοι με την default συμπεριφορά του spring. Για παράδειγμα, το spring όταν έχουμε 404 error, για χρήστες αυτόματα εμφανίζει το 404.html template και για clients επιστρέφει ένα JSON response με πληροφορίες για το exception.

Μηχανισμοί ασφάλειας

Υπάρχουν 2 μηχανισμοί ασφάλειας:

1. Session attribute για users μέσω UI
2. JWT authentication για clients μέσω REST API

Για το UI security κάθε φορά που ένας χρήστης κάνει login αποθηκεύεται σαν session attribute. Αυτό το attribute υπάρχει όσο ο χρήστης έχει ανοιχτό το browser του και γίνεται invalidate όταν κλείσει το browser ή κάνει logout.

Για το REST API security, για να καλεστούν τα mapping με πρόθεμα “/api/v1/” πρέπει ο client να είναι authenticated μέσω ενός JWT κλειδιού.

Deployment

Τα 3 projects που στάλθηκαν από το repository του github στην αρχή τρέχουν σε localhost με βάση δεδομένων την h2 που τρέχει και αυτή τοπικά. Όμως υπάρχει και ένα άλλο version της εφαρμογής που τρέχει σε ένα droplet (ubuntu vm).

Το vm έχει μια σταθερή IP και ένα DNS:

IP: 207.154.212.225

DNS: mycitygov.duckdns.org

Το main project τρέχει στο port 8080, και τα άλλα δυο τρέχουν στην 8081 και 8082. Το minio τρέχει στο σε container στο port 9000, η postgres για το mycitygov τρεχει σε container στο port 5432 και η postgres για το nocgov τρεχει σε container στο port 5433. Όλα αυτά τα ports είναι κλειστά από το firewall.

Όλα τα projects τρέχουν σαν ένα service του συστήματος εκτός από το minio που τρέχει σε container.

Τα μόνα ports που είναι ανοιχτά από το firewall είναι το 22 που χρειάζεται το droplet του digital ocean και το 80 που έχουμε ανοιχρό εμείς. Το vm έχει τον nginx που δέχεται τα requests στο port 80 και τα κάνει redirect στο port 8080 που βρήσκεται το main project. Για να γίνει πιο κατανεμημένη η εφαρμογή, χρησιμοποιούμε postresql σε docker container αντί για h2 τοπικά.

Key points

Η main εφαρμογή καταναλώνει 3 external services (2 δικά μας + 1 file service).

To file service (minio) τρέχει σε ένα docker container.

To deployment version της εφαρμογής τρέχει με postgresql αντί h2 και κρύβει όλα τα ports εκτός από το 80 και το nginx είναι υπεύθυνο να πιάνει τα requests από το 80 και να κάνει redirect στο main project.

Σημείωση

Για να τρέξει το minio τοπικά, όταν τρέξετε το container από το αρχείο “start-minio.sh” πρέπει να κάνετε login με username και password “minioadmin” στο localhost:9000 και να φτιάξετε ένα bucket με όνομα “request-files”. Οι ίδιες οδηγίες εμφανίζονται όταν εκτελείτε το “start-minio.sh”.

Επίσης, στο deployed application δεν θα μπορείτε να κάνετε login γιατί το otp που στέλνετε από το noc-sms γίνεται log στον server (το vm) άρα δεν θα έχετε πρόσβαση σε αυτό. Ο λόγος που γίνεται log στον server και δεν στέλνεται πραγματικά κάποιο sms είναι επειδή οι αριθμοί τηλεφώνων είναι προκαθορισμένοι από το noc-gov και δεν έχουμε πρόσβαση σε αυτά για να μας έρθει το sms, οπότε βλέπουμε το otp από τα logs. Το ίδιο γίνεται και όταν το τρέχουμε τοπικά, αλλά τότε εμείς τρέχουμε το noc-sms άρα έχουμε πρόσβαση στα logs.

Πιθανό Bonus

Η εργασία υποστηρίζει:

- Deployment σε πραγματικό server
 - IP: 207.154.212.225
 - DNS: mycitygov.duckdns.org
- Χρήση Postgres (με Docker container)
- Εγκατάσταση Nginx + ρύθμιση reverse proxy
- Χρήση 3+ εξωτερικών υπηρεσιών
 - noc-gov
 - noc-sms
 - minio
- Προχωρημένες τεχνολογίες (DNS: mycitygov.duckdns.org) από <https://www.duckdns.org>

Postgres σε Container

Για να τρέξουμε τις 2 postgres βάσεις στο deployment version (μια για το mycitygov και μια για το nocgov) τρέχουμε 2 διαφορετικά containers (ένα στο port 5432 και ένα στο port 5433). Για να ξεκινήσουμε τα containers έχουμε 2 shell scripts που ξεκινάνε την postgres.

Για mycitygov:

```
#!/bin/bash

CONTAINER_NAME="mycitygov-postgres"

if [ "$(docker ps -aq -f name=$CONTAINER_NAME)" ]; then
    echo "Container $CONTAINER_NAME already exists. Starting it..."
    docker start $CONTAINER_NAME
else
    echo "Running new Postgres container..."
    docker run -d \
        --name $CONTAINER_NAME \
        -e POSTGRES_DB=mycitygovdb \
        -e POSTGRES_USER=mycitygovuser \
        -e POSTGRES_PASSWORD=mycitygovpassword \
        -p 127.0.0.1:5432:5432 \
        -v mycitygov_postgres_data:/var/lib/postgresql/data \
        postgres:16
fi

echo "Postgres is up!"
```

Για nocgov:

```
#!/bin/bash

CONTAINER_NAME="nocgov-postgres"

if [ "$(docker ps -aq -f name=$CONTAINER_NAME)" ]; then
    echo "Container $CONTAINER_NAME already exists. Starting it..."
    docker start $CONTAINER_NAME
else
    echo "Running new Postgres container..."
    docker run -d \
        --name $CONTAINER_NAME \
        -e POSTGRES_DB=nocgovdb \
        -e POSTGRES_USER=nocgovuser \
        -e POSTGRES_PASSWORD=nocgovpassword \
        -p 127.0.0.1:5433:5432 \
        -v nocgov_postgres_data:/var/lib/postgresql/data \
        postgres:16
fi

echo "Postgres is up!"
```