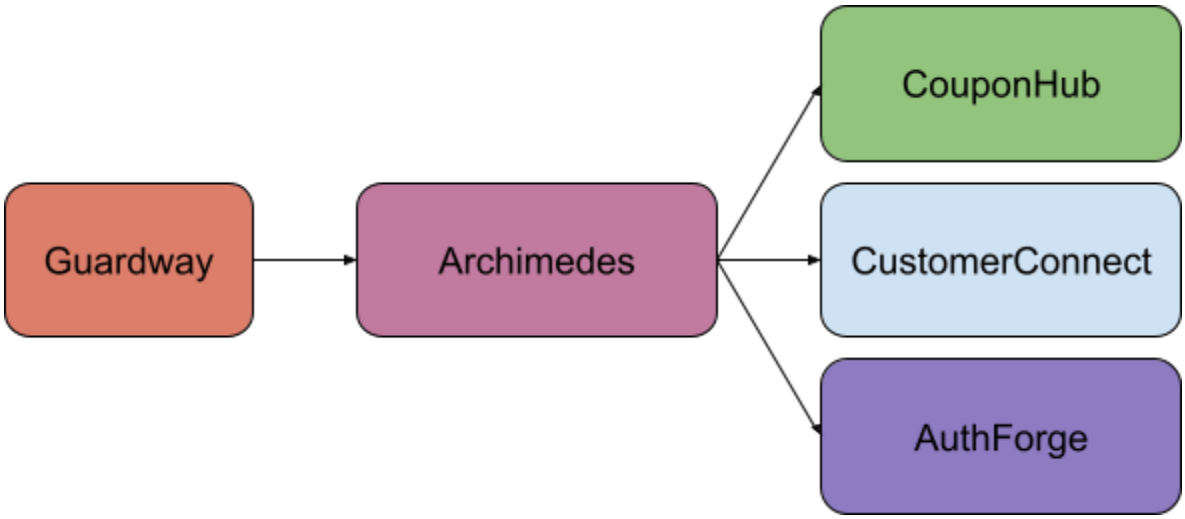
שלב ב' Coupon System

תיאור המערכת

בשלב זה ניקח את המערכת צעד משמעותי קדימה ונחלק את הפעילות והשירותים שלה למספר Microservices (בהמשך ms)

במבט-על המערכת נראית כך:



תפקיד	שם Microservice
שירות Gateway באמצעות Spring Cloud Gateway	Guardway
Discovery Service באמצעות Eureka	Archimedes
שליפה, רכישה ועריכת קופונים.	CouponHub
הכנסה, עדכון, מחיקה וקבלת מידע אודות לקוחות	CustomerConnect
אותנטיקציה (אימות) ואותוריזציה של לקוח (אישור גישה). את AuthForge **אתם מקבלים ממני.**	AuthForge

שימו לב

בהמשך המסמך יופיעו הנחיות כלליות לכתיבת המערכת כמו אילו Entities לכתוב ובאילו נקודות קצה לתמוך, עם זאת לא יופיעו הנחיות לכתיבת DTOs, Mappers וכו' - אך כמובן שהם צריכים להיות קיימים כחלק מכל שירות במערכת.

הנחיות לשימוש ב AuthForge

השירות מאפשר שלוש פעולות בלבד:

- רישום משתמש, user, חדש - פעולת Sign Up.
- התחברות למערכת וקבלת token - פעולת Login.
- אימות token וחילוץ פרטי המשתמש ממנו.

Sign Up

לשם רישום ראשוני למערכת, נשלח username ו password לבחירתנו:

```
POST http://localhost:1337/sign-up
Content-Type: application/json

{
  "username": "Rany",
  "password": "1234"
}
```

כל username ו password יתקבלו כל עוד עדיין לא קיים username במערכת עם השם המבוקש.

לאחר רישום, יווצר משתמש, user, בדטא בייס של AuthForge המכיל username ו password זהים לאלו הקיימים בבקשה בתוספת ל UUID שיווצר אוטומטית.

הנה דוגמא לנתוני הדטא-בייס של AuthForge לאחר רישום מוצלח:

uuid	bcrypt_password	username
7d456b9c-bb...	$2a$10$LWOMa4EJjDEHiN36jbWjW.Au556yvX...	Rany

הסיסמא יכולה להראות לכם מוזרה,
אך בפועל זו אותה 1234 שנשלחה במקור לאחר שעברה עיבוד לשם הגנתה (BCrypt[1]).
לא תידרשו לשלוח את הסיסמא בצורה הזו ולא תצטרכו לדעת מה היא BCrypt בכדי להשתמש ב AuthForge, אך אני ממליץ קריאה בנושא לאחר השלמת שלב זה.

[1] BCrypt

תהליך SignUp מוצלח יסתיים בסטטוס OK ולאחריו ניתן יהיה לבצע login לקבלת token.

Login

לאחר שביצענו Sign Up נוכל להתחבר למערכת - נשלח את פרטי המשתמש שבחרנו בעת הרישום ונקבל token:

```
POST http://localhost:1337/login
Content-Type: application/json

{
  "username": "Rany",
  "password": "1234"
}
```

```
HTTP/1.1 200
Content-Type: text/plain;charset=UTF-8
Content-Length: 260
Date: Fri, 06 Oct 2023 20:43:42 GMT
Keep-Alive: timeout=60
Connection: keep-alive

eyJhbGciOiJIUzUxMiJ9.eyJ1c2VySWQiOiI3ZDQ1NmI5Yy1iY
```

בכל פניה מעתה ואילך נידרש לשלוח token לשם זיהוי המשתמש במערכת - כמובן במקום הפרטים עצמם - Rany ו 1234 במקרה הנ"ל.

אימות token וחילוץ פרטי המשתמש ממנו

AuthForge מסוגלת לקבל token ולהחזיר את פרטי המשתמש המקושרים אליו.
שליחת בקשת GET לכתובת $http://localhost:1337/parse-token$ יחד עם ה Header המכיל את ה token עבור המפתח Authorization תראה כך לדוגמא:

```
###
GET http://localhost:1337/parse-token
Authorization: eyJhbGciOiJIUzUxMiJ9
.eyJ1dWlkIjoiN2Q0NTZiOWMtYmI1NyOGZTRiLTlmZDQtZDA1MjExMzlkYjIyIiwidXNlcm5hbWUiOiJSYW55Iiwic3YiIjoiUmFueSIsIm
lhdCI6MTY5Nzk5NDY1NiwiZXhwIjoxNjk3OTk4MjU2fQ
.CnWuk_qqRk1USWDRzBWkWhiyB1mu84dgWSQby8so-dVaTEaga_Gr[illegible]BiC7GvLR6v1GE31yJ9SDySLGsO29_EkA
```

Token

Details

```
{
  "username": "Rany",
  "uuid": "7d456b9c-bb57-4e4b-9fd4-d0521139db22"
}
```

במידה ופג תוקפו של ה token נקבל Unauthorized ונצטרך לבצע Login מחדש.

פעולת חילוץ פרטי המשתמש תהיה רלוונטית בכל endpoint במערכת.
כל endpoint שיקבל token יישלח אותו ל AuthForge תחילה, יקבל את פרטי המשתמש וימשיך הלאה בפעולתו.

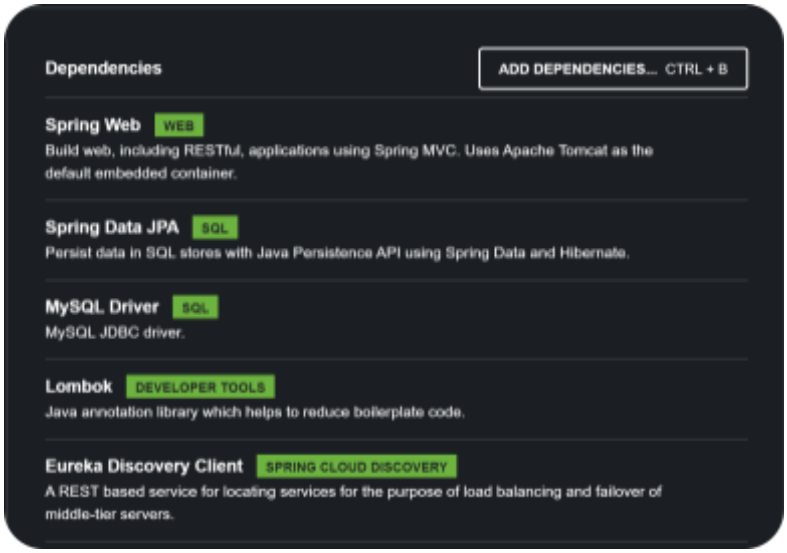
באופן כללי, תוספת של פנייה כזו ל AuthForge היא זמן יקר ובמערכות גדולות נרצה שכל ms יידע לבדו לחלץ את פרטי המשתמש הקיימים ב token, אמנם בפרוייקט זה התוספת זניחה והיא קיימת בכדי למקד את שלב ב' בשאר העניינים.

הנחיות לכתיבת CustomerConnect

תיאור

השירות ישמש לביצוע פעולות על לקוח מסוג צרכן בלבד.

תלויות



ייתכן ותראו לנכון לכלול תלויות נוספות לשירות - עשו זאת.

מאפיינים

- **פורט:** 8081
- **שם אפליקציה:** customer-connect
- **שם דטא-בייס:** customers

Customer Entity

ה Entity היחיד בשירות זה יהיה Customer המכיל את הנתונים הבאים:

- **id** - מזהה ייחודי בדטא בייס.
- **uuid** - מזהה אוניברסלי.
- **firstName** - שם פרטי של הצרכן.
- **lastName** - שם משפחה של הצרכן.
- **email** - דוא"ל הצרכן.
- במידת הצורך הוסיפו **creationTimestamp, updateTimestamp** ו **version**.

השתמשו ב *@PrePersist* בכדי לקבוע uuid רנדומלי טרם הכנסת Customer לדטא-בייס.

CustomerController

המחלקה CustomerController תכיל נקודות קצה לשם קבלת פניות HTTP, כרגיל.

- כל נקודת קצה תקבל token ב Header הבקשה המקושר למפתח Authorization ניתן לעשות זאת באמצעות האנוטציה *@RequestHeader* בעת הגדרת פרמטר בשם token, כך:

```
@RequestHeader("Authorization") String token
```

כפי שהזכרנו קודם לכן, עלינו לשלוח את ה token ל AuthForge תחילה בכדי לקבל את פרטי המשתמש.

הנה נקודות הקצה שיהיו ב CustomerController:

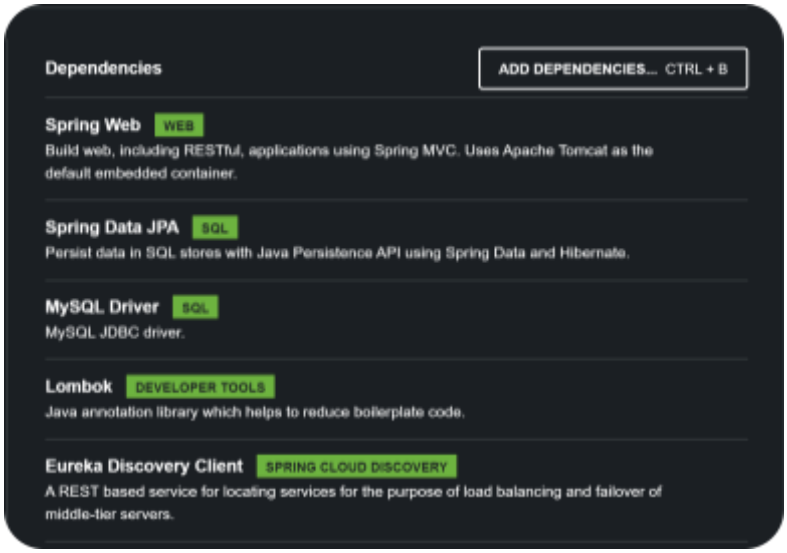
URL	HTTP Method	Description
/customers	POST	יצירת צרכן חדש במערכת - נקודת הקצה תקבל את הצרכן בגוף הבקשה ותגרום הכנסה שלו לדטא-בייס. לאחר פעולה זו ה UUID של הצרכן צריך להיות ה UUID של המשתמש - זה המתקבל מ AuthForge.
/customers/details	GET	קבלת פרטי הצרכן
/customers	GET	קבלת כל הצרכנים
/customers	PUT	עדכון השם הפרטי ושם המשפחה של הצרכן
/customers	DELETE	מחיקת צרכן

הנחיות לכתיבת CouponHub

תיאור

שירות המאפשר ביצוע פעולות הקשורות בקופונים.

תלויות



במידה ותראו לנכון לכלול תלויות נוספות - עשו כך.

מאפיינים

- **פורט:** 8080
- **שם אפליקציה:** coupon-hub
- **שם דטא-בייס:** coupon_hub

Coupon Entity

ה Entity היחיד בשירות זה יהיה Coupon המכיל את הנתונים הבאים:

- **id** - מזהה ייחודי בדטא בייס.
- **uuid** - מזהה אוניברסלי.
- **companyUuid** – מזהה אוניברסלי של החברה – הבעלים של הקופון.
- **category** – מספר המגדיר את הקטגוריה אליה משוייך הקופון.
- **title** – כותרת הקופון
- **startDate** ו **endDate** – תאריכי התחלה וסיום של הקופון.

- **amount** - כמות הקופונים מסוג זה.
- **description** - תיאור הקופון.
- **price** - מחיר הקופון.
- **image** - נתיב, URL, לתמונת הקופון.
- **customers** - סט מזהים אוניברסלים המשוייכים לצרכים אשר רכשו קופון זה.

פירוט על השדה customers

עד כה התרגלנו להגדיר את customers כרשימת הצרכנים עצמם, כלומר $Set < Customer >$ וזה בסדר גמור כשמדובר ב monolith.
אמנם, כעת בסביבה מרובת שירותים אנו עושים מאמץ להפריד את השירותים ועל כן, במקרה הזה, מורידים את הצורך מ CouponHub להגדיר Entity מסוג Customer - הגיוני, שירות זה עוסק בקופונים ולא בדבר אחר.
הטיפוס בו נרצה להשתמש עבור customers הוא $Set < UUID >$ ובכדי לבצע מיפוי באמצעות Hibernate נשתמש באנוטציות $@ElementCollection$ ו $@CollectionTable$ - זאת מאחר ואין לנו Customer.

CouponController

המחלקה CouponController תכיל נקודות קצת לשם עבודה עם קופונים.
גם כאן כל נקודת קצה תקבל token ב Header הבקשה המקושר למפתח Authorization - כך נוכל לדעת מי הוא המשתמש הפונה אלינו - האם מדובר בצרכן או חברה. גם כאן, בדומה ל CustomerController תתבצע פניה ל AuthForge עם ה token ובחזרה יתקבלו הפרטים המאפיינים את המשתמש הפונה.

הנה נקודות הקצה שיהיו ב CouponController

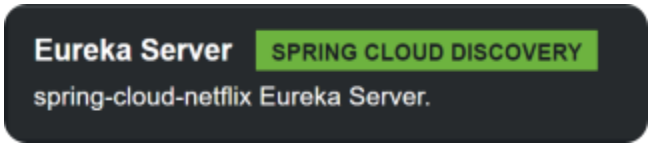
URL	HTTP Method	Description
/coupons/{couponUuid}	GET	קבלת קופון באמצעות UUID
/coupons/customer	GET	קבלת כל הקופונים של הצרכן
/coupons/company	GET	קבלת כל הקופונים של החברה
/coupons/purchase/{couponUuid}	POST	רכישת קופון באמצעות UUID

הנחיות לכתיבת Archimedes

תיאור

השירות ישתמש ב Eureka Server בכדי לאפשר Service Discovery לשאר השירותים במערכת.

תלויות



מאפיינים

פורט: 8761

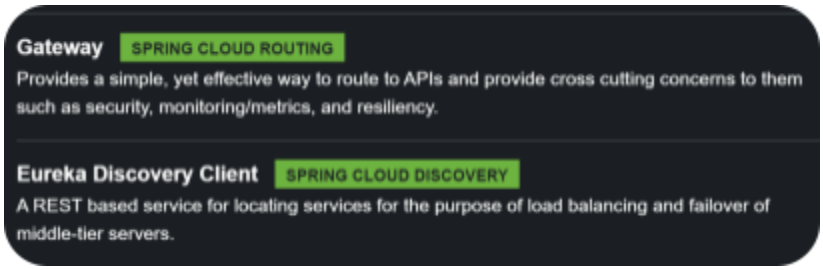
השירות הבסיסי מטבעו לא מכיל לוגיקה, על כן Archimedes מסתיים כאן והוא פועל באופן אוטומטי למען קישור מספר ms-ים.

הנחיות לכתיבת Guardway

תיאור

השירות ישתמש ב Spring Cloud Gateway בכדי ליצור API אחיד לגישה לנקודות המערכת השונות.

תלויות



מאפיינים

- פורט: 9090
- שם אפליקציה: guardway

מטרתו של Guardway היא יצירת Routing בלבד ולשם כך ניתנת לכם הבחירה החופשית בהגדרת ניתוב הפניות - בחרו בנתיבים הגיוניים ואחידים ככל האפשר בכדי שמפתחי ה Front יקבלו את ה API הנוח ביותר לדעתכם - כזה שגם אתם הייתי רוצים לקבל.

עליכם להגדיר נתיבי גישה עבור Auth Forge, Customer Connect ו Coupon Hub.

שאלות נפוצות

- **מדוע אין ms עבור Company?**
  - תשובה: למען הפשטות החלטתי שלא להכביד בשלב זה עם ms נוסף. אתם מוזמנים להוסיף אחד כשתסיימו - כרצונכם.
- **כיצד אפשר להתייחס ל companyUuid אם אין ms עבור Company?**
  - תשובה: חסרונו של ms כזה לא משפיע - הערך עבור companyUuid יכול להיות קיים ללא כל בעיה, כרגיל, אפילו אם אין ms שמסוגל לתמוך בבקשותיה של חברה בעלת אותו המזהה - ניתן לאחסן אותו בדטא בייס, לשלוח אותו, לקבל אותו ולמעשה לעשות בו כל דבר שנרצה. כמובן במידה ובעתיד נכתוב ms עבור Company התמיכה בקבלת בקשות מחברה בעלת companyUuid תהיה טריוויאלית.
- **בכתיבת CustomerController, למה הכוונה בהנחייה "לאחר פעולה זו ה UUID של הצרכן צריך להיות ה UUID של המשתמש - זה המתקבל מ AuthForge." ?**
  - תשובה: בסעיף זה המטרה היא יצירת צרכן, על כן יש לשייך אליו UUID. מאחר ונקודת הקצה פונה ל AuthForge, נקבל בחזרה UUID המזהה את המשתמש. את ה UUID הזה נשייך לצרכן.
- **מה הוא בעצם משתמש?**
  - תשובה: משתמש כרגע הוא צרכן בלבד. בעתיד אם נחליט לכתוב ms עבור Company משתמש יוכל להיות גם כן חברה. באופן שכזה משתמש הוא דרך אחת לזיהוי האינטרקציה עם המערכת וצרכן או חברה (בעתיד) הם מידע ספציפי יותר בנוגע למשתמש - בדרך כזו ניתן ליצור כמה סוגי משתמשים שנרצה ובכך להרחיב את המערכת עם הזמן, כאשר הכל בנוי על תשתית בלתי משתנה של user.

בהצלחה!