בפעם הקודמת עשינו דברים ממש hard coded

בוא נחליף את זה במשתנים

אז נוסיף את המשתנים ב application.properties

```
application.jwt.secretKey = secure_secure_secure_secure_secure_secure_secure_secure_
application.jwt.tokenPrefix =Bearer
application.jwt.tokenExpirationAfter =14
```

נוסיף ממחלקה חדשה שתהיה אחראית לכל ההגדרות של ה JWT

```
import com.google.common.net.HttpHeaders;

@Component
@ConfigurationProperties(prefix = "application.jwt")
public class JwtConfig {
    private String secretKey;
    private String tokenPrefix;
    private Integer tokenExpirationAfter;

    public JwtConfig() {
    }

    public String getSecretKey() {
        return secretKey;
    }

    public void setSecretKey(String secretKey) { this.secretKey = secretKey; }

    public String getTokenPrefix() { return tokenPrefix; }

    public void setTokenPrefix(String tokenPrefix) { this.tokenPrefix = tokenPrefix; }

    public Integer getTokenExpirationAfter() { return tokenExpirationAfter; }

    public void setTokenExpirationAfter(Integer tokenExpirationAfter) {
        this.tokenExpirationAfter = tokenExpirationAfter;
    }

    public String getAuthorizationHeader(){
        return HttpHeaders.AUTHORIZATION;
```

ניצור עוד מחלקה שתהיה אחראית לקודד את המפתח שלנו

נשים את המתודה ב bean כדי שנוכל להשתמש בה ולא נצטרך את המחלקה

```
@Configuration
public class JwtSecretKey {
    private final JwtConfig jwtConfig;

    public JwtSecretKey(JwtConfig jwtConfig) {
        this.jwtConfig = jwtConfig;
    }

    @Bean
    public SecretKey secretKey(){
        return Keys.hmacShaKeyFor(jwtConfig.getSecretKey().getBytes());
    }
}
```

נוסיף את שתי המשתנים האלה

```
private final AuthenticationManager authenticationManager;
private final JwtConfig jwtConfig;
private final SecretKey secretKey;

public JwtUsernameAndPasswordAuthenticationFilter(AuthenticationManager authenticationManager,
                                                  JwtConfig jwtConfig,
                                                  SecretKey secretKey) {
    this.authenticationManager = authenticationManager;
    this.jwtConfig = jwtConfig;
    this.secretKey = secretKey;
}
```

ואז נוכל להחליף אותם במה שעשינו קודם

```
@Override
protected void successfulAuthentication(HttpServletRequest request,
                                        HttpServletResponse response,
                                        FilterChain chain, Authentication authResult
) throws IOException, ServletException {
    String token = Jwts.builder()
            .setSubject(authResult.getName()) // השם משתמש
            .claim( s: "authorities", authResult.getAuthorities()) // ההרשאות שלו
            .setIssuedAt(new java.util.Date()) // מתי זה קרה
            .setExpiration(Date.valueOf(LocalDate.now().plusDays(jwtConfig.getTokenExpirationAfter()))) // מתי זה נגמר
            .signWith(secretKey); // ה signature עם מפתח
            .compact(); // יצירה שלו

    response.addHeader(jwtConfig.getAuthorizationHeader(), s1: jwtConfig.getTokenPrefix() + token);
}
```

(עשינו בשתי המחלקות של הפילטר)

עכשיו וסיף במחלקה של ההגדרות גם את שתי המשתנים ומעדכן בבנאי.

```
@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class ApplicationSecurityConfig extends WebSecurityConfigurerAdapter {

    private final PasswordEncoder passwordEncoder;
    private final ApplicationUserService applicationUserService;
    private final SecretKey secretKey;
    private final JwtConfig jwtConfig;

    @Autowired
    public ApplicationSecurityConfig(
            PasswordEncoder passwordEncoder,
            ApplicationUserService applicationUserService, SecretKey secretKey, JwtConfig jwtConfig) {
        this.passwordEncoder = passwordEncoder;
        this.applicationUserService = applicationUserService;
        this.secretKey = secretKey;
        this.jwtConfig = jwtConfig;
    }
```

נעדכן את הקריאות לבנאים כשיצרנו אותם בפילטר

```
.addFilter(new JwtUsernameAndPasswordAuthenticationFilter(authenticationManager(), jwtConfig, secretKey))
.addFilterAfter(new JwtTokenVerifier(jwtConfig, secretKey),JwtUsernameAndPasswordAuthenticationFilter.class)
```