

```

/**
 * Spring webservice kliens java alapu konfiguracios fajlja
 */
@Configuration
public class WebServiceConfig {

    @Autowired
    private ApplicationContext applicationContext;

    /**
     * Az alkalmazas kivulrol megadhato property-jeit tartalmazo osztaly. A peldaban vegpont url-ek, keystore jelszo es alias
     * adatok kinyeresere szolgal
     */
    @Autowired
    private AppProperties applicationProperties;

    /**
     * Az uzenet alairasahoz szukseges adatok
     */
    private static final String SIGNATURE_ALG_METHOD = "http://www.w3.org/2001/04/xmldsig-more#rsa-sha384";
    private static final String SIGNATURE_DIGEST_ALG_METHOD = "http://www.w3.org/2001/04/xmldsig-more#sha384";
    private static final String SIGNATURE_KEY_IDENTIFIER = "DirectReference";
    private static final String SIGNATURE_PARTS = "{http://schemas.xmlsoap.org/soap/envelope/}Body;"
        + "{http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd}Timestamp";

    /**
     * Az uzenet ki es becsomagolasaert felelos marshaller megvalositas
     */
    @Bean
    public Jaxb2Marshaller marshaller() {
        final Jaxb2Marshaller marshaller = new Jaxb2Marshaller();
        marshaller.setPackagesToScan(PackageMarker.class.getPackageName());
        return marshaller;
    }

    /**
     * Az esemenyvezerelt adatkuldeshez hasznalando web service kliens konfiguracioja
     * Az NTAK adatfogadast vegzo vegpontjanak URL-jet, a fent definialt marshallert, illetve a lent definialt security
     * interceptort adjuk meg ennek konfiguraciojaket
     */
    @Bean
    public EsemenyvezereltAdatkuldesWebServiceClient esemenyvezereltAdatkuldesWebServiceClient(final Jaxb2Marshaller
    marshaller)
        throws Exception {
        final EsemenyvezereltAdatkuldesWebServiceClient client = new EsemenyvezereltAdatkuldesWebServiceClient();
        client.setDefaultUri(applicationProperties.getNtakEsemenyvezereltAdatkuldesVegpont());
        client.setMarshaller(marshaller);
        client.setUnmarshaller(marshaller);
        final ClientInterceptor[] interceptors = new ClientInterceptor[]{securityInterceptor()};
        client.setInterceptors(interceptors);
        return client;
    }

    /**
     * Alairas eloallitasahoz hasznalando security interceptor
     */
    @Bean
    public Wss4jSecurityInterceptor securityInterceptor() throws Exception {
        final Wss4jSecurityInterceptor securityInterceptor = new Wss4jSecurityInterceptor();
        // Alairas es Timestamp az uzenethez adasa
        securityInterceptor.setSecurementActions("Signature Timestamp");
        securityInterceptor.setSecurementMustUnderstand(false);

        // Kliens oldali tanusitvany adatainak megadasa
        securityInterceptor.setSecurementUsername(applicationProperties.getClientKeyAlias());
        securityInterceptor.setSecurementPassword(applicationProperties.getClientKeyPassword());
        // Alairashoz hasznalatos alairo algoritmus adatainak megadasa
        securityInterceptor.setSecurementSignatureAlgorithm(SIGNATURE_ALG_METHOD);
        securityInterceptor.setSecurementSignatureDigestAlgorithm(SIGNATURE_DIGEST_ALG_METHOD);
        securityInterceptor.setSecurementSignatureKeyIdentifier(SIGNATURE_KEY_IDENTIFIER);
        securityInterceptor.setSecurementSignatureParts(SIGNATURE_PARTS);
        securityInterceptor.setSecurementSignatureCrypto(getCryptoFactoryBean().getObject());
        return securityInterceptor;
    }

    @Bean
    public CryptoFactoryBean getCryptoFactoryBean() throws IOException {
        final CryptoFactoryBean cryptoFactoryBean = new CryptoFactoryBean();
        cryptoFactoryBean.setKeyStorePassword(applicationProperties.getKeystorePassword());
        cryptoFactoryBean.setKeyStoreLocation(applicationContext.getResource(applicationProperties.getKeystorePath()));
        return cryptoFactoryBean;
    }
}

```