

KINDEROPVANG

IKG Web Services

Versie 1.4

© Kind & Gezin

Inhoud

1	DOCUMENTBEHEER	4
1.1	Historiek van het document	4
2	DOEL VAN DIT DOCUMENT	5
3	INTRODUCTIE TOT IKG WEBSERVICE	6
3.1	Context	6
4	TECHNISCHE SPECIFICATIES VOOR DE IKG WEB SERVICE	7
4.1	Security Concepten	7
4.1.1	Transport-level security	7
4.1.2	Message-level security	7
4.2	Security Policy	7
4.2.1	Kind en Gezin certificaat	8
4.2.2	Ws-Security in IKG WebServices	8
4.3	Security Implementatie op client - voorbeeld	12
4.3.1	Java	12
4.3.2	Spring configuratie	12
4.4	WsdI	14
4.5	Message Payload voor MatchKind	15
4.5.1	Waarvoor wordt dit gebruikt?	15
4.5.2	Voorbeeld xml	15
4.5.3	Elementen	15
4.6	MatchKind response	16
4.6.1	Voorbeeld xml	16
4.6.2	Elementen	16
4.7	Message Payload voor MatchKinderen	17
4.7.1	Waarvoor wordt dit gebruikt?	17
4.7.2	Voorbeeld xml	17
4.7.3	Elementen	17
4.8	MatchKinderen response	18
4.8.1	Voorbeeld xml	18
4.8.2	Elementen	18
4.9	Message Payload voor MatchKinderenResultaat	18
4.9.1	Waarvoor wordt dit gebruikt?	18
4.9.2	Voorbeeld xml	18
4.9.3	Elementen	18
4.10	MatchKinderenResultaat response	18
4.10.1	Voorbeeld xml	19

4.10.2	Elementen	19
4.11	Message Payload voor OpgangPrestatiesResultaat	20
4.11.1	Waarvoor wordt dit gebruikt?	20
4.11.2	Voorbeeld xml	20
4.11.3	Elementen	20
4.12	OpgangPrestatiesResultaat response	20
4.12.1	Voorbeeld xml	20
4.12.2	Elementen	21
4.13	element KindInput	21
4.14	element KindOutput	21
4.14.1	mogelijke opmerking codes in KindOutput	22
4.15	element OpgangPrestatiesResultaat	22
5	FOUTMELDINGEN	23
5.1	Mogelijke return codes in xml responses	23
5.2	Security validatie	23
5.3	Validatie fout in request xml	24
6	VOORBEELD APPLICATIE	25
6.1	Beschikbaarheid	25
6.2	Gebruik van de voorbeeld client applicatie	25
7	REFERENTIES	26
7.1	Certificaat via Kind en Gezin	26
7.2	Web-service Security	26
8	CONTACTGEGEVENS	27
8.1	Contactgegevens voor de IKG Webservice	27

1 DOCUMENTBEHEER

1.1 Historiek van het document

Versie	Datum	Beschrijving van de wijzigingen / opmerkingen
0.1	26/08/2015	Initiële (draft) versie van het cookbook, gebaseerd op template.
0.2	09/09/2015	Toevoegen van payloads
0.3	14/09/2015	Toevoegen opmerkingCodes en contact e-mail adressen
0.4	14/09/2015	Tekstuele aanpassingen aan o.a. context
1.0	15/09/2015	Finalisering
1.1	23/02/2016	Toevoegen van referentiedatum voor MatchKind & MatchKinderen
1.1	29/02/2016	Toevoegen referenties certificaten Onderwijs Vlaanderen
1.1	15/03/2016	Tekstuele aanpassingen
1.1	7/6/2016	Enkele verwijzingen naar onderwijs vlaanderen vervangen
1.2	20/06/2016	Aangeven dat K&G certificaat voor beide omgevingen geldig is (test en prod)
1.3	09/03/2017	Certificaat per softwareleverancier mogelijk
1.4	28/10/2021	Wijziging mbt keysize CSR en te gebruiken certificaat

2 DOEL VAN DIT DOCUMENT

Dit document moet beschouwd worden als een handleiding voor het implementeren van de IKG Web Services. Een degelijke technische kennis met betrekking tot Web Services, Web Service Security en het gebruik van certificaten is aangewezen.

In dit document wordt verwezen naar een implementatie die gebruik maakt van Java en Spring. Indien u een andere taal en tools voor de implementatie gebruikt verwijzen we u naar de beschikbare documentatie hierover.

Dit document is dus hoofdzakelijk bestemd voor Web Service Developers.

3 INTRODUCTIE TOT IKG WEBSERVICE

3.1 Context

Om toe te laten dat organisatoren van kinderopvangvoorzieningen hun gegevens kunnen toetsen aan de gegevens zoals die in de IKG-databank van Kind & Gezin aanwezig zijn wordt de mogelijkheid geboden om die gegevens via webservice(s) te bevragen.

Via deze webservice kan een organisator, die kinderopvang organiseert in locaties die werken volgens het principe van 'inkomensgerelateerde opvang', op een geautomatiseerde manier informatie zoals attesten inkomenstarieven opvragen, opvangprestaties doorgeven en het resultaat van de verwerking van deze opvangprestaties opvragen.

Deze webservice wordt verder in dit document benoemd als de "IKG Web Service".

Er wordt gebruik gemaakt van 'synchrone' en 'asynchrone' webservices. Concreet betekent dit dat er, voor de mogelijkheid tot het matchen van één kindcode, gebruik wordt gemaakt van een 'synchrone' webservice waarbij het resultaat onmiddellijk teruggestuurd wordt, terwijl er voor de matching van kindcodes 'in bulk' er gebruik gemaakt wordt van een 'asynchrone' manier van werken. Daarbij worden in eerste instantie de 'te checken' kindcodes doorgegeven en in tweede instantie kan het resultaat van de matching opgevraagd worden. Hierdoor zijn de aanlevering van de data en de verwerking van deze data in de IKG-database van elkaar gescheiden. Dit is noodzakelijk omdat, de tijd nodig om de matching uit te voeren, nogal hoog kan oplopen afhankelijk van het aantal te verwerken kindgegevens.

Voor de identificatie van de organisator wordt het volgende certificaat gebruikt:

- Kind en Gezin certificaten

4 TECHNISCHE SPECIFICATIES VOOR DE IKG WEB SERVICE

4.1 Security Concepten

4.1.1 Transport-level security

De communicatie tussen de cliënt en de Kind & Gezin server gebeurt steeds via een beveiligde verbinding (HTTPS).

4.1.2 Message-level security

Voor de beveiliging van de requests moeten de SOAP requests beveiligd worden door gebruik te maken van WS-Security. De requirements hiervoor worden verder toegelicht in volgende paragraaf.

De responses worden niet beveiligd aangezien het gaat om een synchroon proces en de connectie steeds via https verloopt.

4.2 Security Policy

De IKG Web Service maakt gebruik van de X509 standaard voor het beveiligen van de web service. Dit betekent dat de gebruiker in het bezit moet zijn van een geldig certificaat en de bijhorende private sleutel van de organisatie.

Het certificaat kan nog steeds individueel per organisator maar om efficiëntieredenen kan men ook kiezen voor certificaat op naam van leverancier.

4.2.1 Kind en Gezin certificaat

Indien uw organisatie nog niet in het bezit is van een certificaat van Kind en Gezin, dient u deze aanvraag te doen:

- **Genereer een key-pair (keysize 4096)**
U kan dit bijvoorbeeld doen via de tool Keystore Explorer (zie Certificaat via Kind en Gezin).
In het veld CN moet prefix **CBE=** staan en als suffix **KG**
In het veld E staat het emailadres van de beheerder van certificaat (dit is belangrijk want zo kan je op de hoogte gebracht worden om tijdig je certificaat te vernieuwen)
 - CN = CBE=1234567890KG
(het nummer dient het ondernemingsnummer te zijn)
 - OU= Opvangvoorziening x
 - O = Opvangvoorziening x
 - L = BRUSSELS
 - ST=BELGIUM
 - C=BE
 - E= jan@voorziening.be
- **Maak een CSR (Certificate Signing Request) aan.**
- **Stuur een email naar Software-ontwikkeling@kindengezin.be**
 - In bijlage het CSR-bestand
 - In bijlage volmachtsformulier
 - In het onderwerp moet **[CSR-<ondernemingsnummer>]** staan
Indien bvb het ondernemingsnummer van de organisator 12345 zou zijn ziet een goed opgebouwd onderwerp er dus als volgt uit: **[CSR-12345]**
- **Zodra het certificaat is aangemaakt door de certificaatbeheerder zal u via email het certificaat ontvangen**

Het certificaat kan voor zowel testomgeving als productieomgeving gebruikt worden !

4.2.2 Ws-Security in IKG WebServices

Daar de authenticatie gebeurt op organisator-niveau, dient het KBO nummer (=ondernemingsnummer) opgenomen te worden in de Common Name van het certificaat.

Het certificaat kan nog steeds individueel per organisator maar om efficiëntieredenen kan men ook kiezen voor certificaat op naam van leverancier. Dit is ook te verkiezen volgens principe van subsidiariteit. Ook bij een certificaat op naam van de leverancier dient er een KBO-nummer ingevuld te worden, nl. die van de softwareleverancier.

Een request dient digitaal ondertekend te worden en het certificaat moet meegestuurd worden met het request. Een voorbeeld van een request vindt u terug op een volgende pagina.

Volgende onderdelen kunnen we onderscheiden in een uitgaande request en moeten allen aanwezig zijn zoals hieronder beschreven :

1. **Timestamp**: timestamp dat de request is aangemaakt
2. **BinarySecurityToken**: Dit element bevat het certificaat dat meegestuurd wordt in Base64 formaat.
3. **CanonicalizationMethod**: <http://www.w3.org/2001/10/xml-exc-c14n#>
4. **SignatureMethod**: <http://www.w3.org/2000/09/xmldsig#rsa-sha1>
5. **DigestMethod**: <http://www.w3.org/2000/09/xmldsig#sha1>
6. **SignatureValue**: Bevat de digitale handtekening
7. **Body**: Bevat de message payload zoals beschreven in hoofdstuk 4.5, 4.7, 4.9 en 4.11

De waarden van de security elementen worden geleverd door het framework wanneer dit correct werd geconfigureerd. (In het voorbeeld in de volgende sectie is de verantwoordelijke klasse [org.springframework.ws.soap.security.xwss.XwsSecurityInterceptor](#))

De volgende elementen moeten getekend zijn:

- Het header element Timestamp
- Het header element BinarySecurityToken
- De body

Hieronder bevindt zich een voorbeeld van een SOAP bericht. De nummering verwijst naar de nummering die hierboven is gebruikt bij de opsomming van welke onderdelen er aanwezig moeten zijn in een uitgaande request om data aan te leveren:

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      SOAP-ENV:mustUnderstand="1">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
        wsu:Id="XWSSGID-1440752507788-129749642">
          <wsu:Created>2015-08-28T09:01:47Z</wsu:Created>
          <wsu:Expires>2015-08-28T09:02:47Z</wsu:Expires>
        </wsu:Timestamp>
      <wsse:BinarySecurityToken
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
        ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
        wsu:Id="XWSSGID-1440752507793716767283">M...=</wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="XWSSGID-1440752507791204898592">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <ds:Reference URI="#XWSSGID-14407525079871883729808">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>r...=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#XWSSGID-1440752507793716767283">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>P...=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#XWSSGID-1440752507788-129749642">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>V...=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>G...=</ds:SignatureValue>
      </ds:Signature>
    </wsse:Security>
  </SOAP-ENV:Header>
</SOAP-ENV:Envelope>

```

1

2

3

4

5

6

```
<ds:KeyInfo>
  <wss:SecurityTokenReference
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="XWSSGID-1440752507977-2078894953">
    <wss:Reference URI="#XWSSGID-144075250793716767283"
      ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
    </wss:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
</wss:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  wsu:Id="XWSSGID-14407525079871883729808">
  <ns2:matchKindRequest xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
    <ns2:ondernemingsNummer>...</ns2:ondernemingsNummer>
    <ns2:kindInput>
      <ns2:naam>...</ns2:naam>
      <ns2:voornaam>...</ns2:voornaam>
      <ns2:geboorteDatum>...</ns2:geboorteDatum>
      <ns2:kindCode>...</ns2:kindCode>
    </ns2:kindInput>
  </ns2:matchKindRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7

4.3 Security Implementatie op client - voorbeeld

4.3.1 Java

Indien men opteert voor een java implementatie is het aanbevolen om minstens java 1.7 te gebruiken.

Hoogstwaarschijnlijk zal bij de implementatie van de webservice gebruik gemaakt worden van een 'security' framework zoals XWSS (<http://xwss.java.net/>) of WSS4J(<http://ws.apache.org/wss4j/>). Onafhankelijk van de implementatie dient aan de eisen uit de voorgaande sectie voldaan te worden.

4.3.2 Spring configuratie

Als mogelijke aanzet wordt een voorbeeld client meegegeven die gebruik maakt van Spring-Webservices en Spring-WS-Security(2.2.1.RELEASE) en XWSS als security framework.

De spring configuratie in de voorbeeld client wordt geladen via de class ClientConfiguration. Spring herkent dat deze class de configuratie bevat door middel van de @Configuration annotatie

```
@Configuration
@ImportResource("classpath:config/client-config.xml")
@PropertySources({
    @PropertySource("classpath:config/kinderopvang-ikg-ws.properties"),
    @PropertySource("file:${user.home}/kinderopvang-ikg-webservice/config/kinderopvang-ikg-
ws.properties")
})
public class ClientConfiguration {

    @Value("${generated.classes.dir}") private String generatedClassDir;
    @Value("${webservice.uri}") private String webserviceUri;
    @Value("${policy.file}") private String policyFile;
    @Value("${org.cert.alias}") private String defaultAlias;
    @Value("${org.cert.password}") private String certPassword;
    @Value("${org.keystore.file}") private String keyStoreLocation;
    @Value("${org.keystore.password}") private String keyStorePassword;

    @Bean
    public static PropertySourcesPlaceholderConfigurer propertySourcesPlaceholderConfigurer() {
        return new PropertySourcesPlaceholderConfigurer();
    }

    @Bean
    public Jaxb2Marshaller marshaller() {
        Jaxb2Marshaller marshaller = new Jaxb2Marshaller();
        marshaller.setContextPath(generatedClassDir);
        return marshaller;
    }
}
```

Client-config.xml:

```
<bean id="webServiceTemplate" class="org.springframework.ws.client.core.WebServiceTemplate">
  <property name="marshaller" ref="marshaller" />
  <property name="unmarshaller" ref="marshaller" />
  <property name="defaultUri" value="{webservice.uri}" />
  <property name="interceptors" ref="xwsSecurityInterceptor" />
</bean>

<bean id="xwsSecurityInterceptor"
  class="org.springframework.ws.soap.security.xwss.XwsSecurityInterceptor">
  <property name="policyConfiguration" value="classpath:{policy.file}" />
  <property name="callbackHandler" ref="keystoreCallbackHandler" />
</bean>

<bean id="keystoreCallbackHandler"
  class="org.springframework.ws.soap.security.xwss.callback.KeyStoreCallbackHandler">
  <property name="keyStore" ref="keyStore" />
  <property name="defaultAlias" value="{org.cert.alias}" />
  <property name="privateKeyPassword" value="{org.cert.password}" />
</bean>

<bean id="keyStore" class="org.springframework.ws.soap.security.support.KeyStoreFactoryBean">
  <property name="location" value="file:{org.keystore.file}" />
  <property name="password" value="{org.keystore.password}" />
</bean>
```

In client-config.xml worden volgende beans gedefinieerd:

webServiceTemplate

Beschrijving: Centrale klasse voor client-side web services. Verantwoordelijk voor opzetten connectie, aanmaken bericht, versturen bericht, delegeren security,...

Library: spring-ws-core-2.2.1.RELEASE

xwsSecurityInterceptor

Beschrijving: Interceptor die gebruikt wordt om de vereiste security toe te voegen. Dit verwijst naar de securitypolicy.xml en de callbackHandler. Dit is nodig voor het aanmaken van de digitale handtekening en om te weten wat er moet worden getekend. Dit is custom code omdat hierin nog de extra header tag "extrainfo" wordt aangemaakt. Deze custom code extends van org.springframework.ws.soap.security.xwss.XwsSecurityInterceptor

Library: custom code die gebruik maakt van spring-ws-security-2.1.0.RELEASE

keyStoreCallbackHandler

Beschrijving: Bevat de gegevens van het certificaat en verwijst naar de bean die de locatie van de keystore weergeeft.

Library: spring-ws-security-2.2.1.RELEASE

keyStore

Beschrijving: Bevat de locatie en het paswoord van de keystore

Library: spring-ws-security-2.1.0.RELEASE

In de bean `xwsSecurityInterceptor` wordt er verwezen naar de `securityPolicy.xml` (`${policy.file}`). In deze xml wordt er opgegeven wat er moet worden getekend (en hoe). Deze xml ziet er als volgt uit:

```
<?xml version="1.0" encoding="UTF-8"?>
<xwss:SecurityConfiguration dumpMessages="false"
  xmlns:xwss="http://java.sun.com/xml/ns/xwss/config">
  <xwss:Timestamp timeout="60" />
  <xwss:Sign includeTimestamp="false">
    <xwss:X509Token />
    <xwss:CanonicalizationMethod
      algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
      disableInclusivePrefix="true" />
    <xwss:SignatureMethod algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <xwss:SignatureTarget type="qname"
      value="{http://www.w3.org/2003/05/soap-envelope}Body">
      <xwss:DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <xwss:Transform disableInclusivePrefix="true"
        algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    </xwss:SignatureTarget>
    <xwss:SignatureTarget type="qname"
      value="{http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd}BinarySecurityToken">
      <xwss:DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <xwss:Transform disableInclusivePrefix="true"
        algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    </xwss:SignatureTarget>
    <xwss:SignatureTarget type="qname"
      value="{http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd}Timestamp">
      <xwss:DigestMethod algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <xwss:Transform disableInclusivePrefix="true"
        algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    </xwss:SignatureTarget>
  </xwss:Sign>
</xwss:SecurityConfiguration>
```

4.4 Wsdl

De definitie van de webservice is terug te vinden via de url:

testomgeving	https://tstarws.kindengezin.be/kinderopvang-ikg-webservices/service/ikg.wsdl
productieomgeving	https://arws.kindengezin.be/kinderopvang-ikg-webservices/service/ikg.wsdl

De webservice bevat 4 operaties:

- **MatchKind:** Matchen van de kind gegevens van 1 kind.
- **MatchKinderen:** Matchen van de gegevens van meerdere kinderen.
- **MatchKinderenResultaat:** Ophalen van de resultaten van het 'MatchKinderen'-request.
- **OpvangPrestatiesResultaat:** Opvragen van de resultaten van de verwerking van doorgestuurde OpvangPrestaties.

4.5 Message Payload voor MatchKind

4.5.1 Waarvoor wordt dit gebruikt?

Aan de hand van de opgegeven kindgegevens wordt een matching gedaan met de gegevens zoals die in de IKG databank aanwezig zijn.

Optioneel kan gebruik gemaakt worden van een 'Referentiedatum'. Indien dit wordt meegegeven zal het resultaat teruggegeven worden van het attest inkomenstarief waarbij de referentiedatum gelegen is in de geldigheidsperiode van het attest. Indien deze datum niet wordt meegegeven wordt de systeemdatum als defaultwaarde gebruikt.

4.5.2 Voorbeeld xml

```
<matchKindRequest xmlns="http://www.kindengezin.be/kinderopvang/webservice">
  <ondernemingsNummer>0123456789</ondernemingsNummer>
  <kindInput>
    <naam>naam</naam>
    <voornaam>voornaam</voornaam>
    <geboorteDatum>2001-01-01</geboorteDatum>
    <kindCode>01010100101</kindCode>
  </kindInput>
  <referentieDatum>2016-02-16+01:00</referentieDatum>
</matchKindRequest>
```

4.5.3 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
ondernemingsNummer	int	Max lengte: 10	1	1
kindInput	kindInput	Zie (4.13)	1	1
referentieDatum	date		0	1

4.6 MatchKind response

4.6.1 Voorbeeld xml

```
<ns2:matchKindResponse xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:returnCode>0000</ns2:returnCode>
  <ns2:kindOutput>
    <ns2:input>
      <ns2:naam>naam</ns2:naam>
      <ns2:voornaam>voornaam</ns2:voornaam>
      <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
      <ns2:kindCode>12091000703</ns2:kindCode>
    </ns2:input>
    <ns2:naam>naam</ns2:naam>
    <ns2:voornaam>voornaam</ns2:voornaam>
    <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
    <ns2:kindCode>120910-007</ns2:kindCode>
    <ns2:tariefCode>120910-007-03</ns2:tariefCode>
    <ns2:inkomenstarief>12.92</ns2:inkomenstarief>
    <ns2:geldigVan>2015-01-01+01:00</ns2:geldigVan>
    <ns2:geldigTot>2015-12-31+01:00</ns2:geldigTot>
    <ns2:opmerkingCode>1000</ns2:opmerkingCode>
    <ns2:referentieDatum>2016-02-16+01:00</ns2:referentieDatum>
  </ns2:kindOutput>
</ns2:matchKindResponse>
```

4.6.2 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
returnCode	int	Max lengte: 4	1	1
kindOutput	kindOutput	Zie (4.14)	1	1

4.7 Message Payload voor MatchKinderen

4.7.1 Waarvoor wordt dit gebruikt?

Laat toe om voor een groot aantal kind gegevens een matching te doen met de gegevens zoals die in de IKG-databank aanwezig zijn. Deze operatie dient om de kind gegevens te uploaden naar Kind & Gezin. Het resultaat van de matching kan met de operatie MatchKinderenResultaat nadien opgehaald worden.

Optioneel kan gebruik gemaakt worden van een 'Referentiedatum'. Indien dit wordt meegegeven zal het resultaat teruggegeven worden van het attest inkomenstarief waarbij de referentiedatum gelegen is in de geldigheidsperiode van het attest. Indien deze datum niet wordt meegegeven wordt de systeemdatum als defaultwaarde gebruikt.

4.7.2 Voorbeeld xml

```
<ns2:matchKinderenRequest xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:ondernemingsNummer>0123456789</ns2:ondernemingsNummer>
  <ns2:kindInput>
    <ns2:naam>naam kind1</ns2:naam>
    <ns2:voornaam>voornaam kind1</ns2:voornaam>
    <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
    <ns2:kindCode>12091000703</ns2:kindCode>
  </ns2:kindInput>
  <ns2:kindInput>
    <ns2:naam>naam kind2</ns2:naam>
    <ns2:voornaam>voornaam kind2</ns2:voornaam>
    <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
    <ns2:kindCode>12091000801</ns2:kindCode>
  </ns2:kindInput>
  <ns2:referentieDatum>2016-02-16+01:00</ns2:referentieDatum>
</ns2:matchKinderenRequest>
```

4.7.3 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
ondernemingsNummer	int	Max lengte: 10	1	1
kindInput	kindInput	Zie (4.13)	1	~
referentieDatum	date		0	1

4.8 MatchKinderen response

4.8.1 Voorbeeld xml

```
<ns2:matchKinderenResponse xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:returnCode>0000</ns2:returnCode>
  <ns2:matchKinderenRequestID>447</ns2:matchKinderenRequestID>
</ns2:matchKinderenResponse>
```

4.8.2 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
returnCode	int	Max lengte: 4	1	1
matchKinderenRequestID	int		1	1

4.9 Message Payload voor MatchKinderenResultaat

4.9.1 Waarvoor wordt dit gebruikt?

Deze geeft de resultaten terug van de matching van de kind gegevens die met MatchKinderen werden aangeleverd.

Wanneer deze resultaten nog niet beschikbaar zijn, m.a.w. nog verwerkt worden, dan zal return code '0005' teruggegeven worden en blijft de lijst met resultaten leeg.

Van zodra de resultaten wel beschikbaar zijn wordt return code '0000' teruggegeven en is de lijst met resultaten aanwezig in de response.

4.9.2 Voorbeeld xml

```
<ns2:matchKinderenResultaatRequest xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:ondernemingsNummer>0123456789</ns2:ondernemingsNummer>
  <ns2:matchKinderenRequestID>447</ns2:matchKinderenRequestID>
</ns2:matchKinderenResultaatRequest>
```

4.9.3 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
ondernemingsNummer	int	Max lengte: 10	1	1
matchKinderenRequestID	int		1	1

4.10 MatchKinderenResultaat response

4.10.1 Voorbeeld xml

Nog in verwerking:

```
<ns2:matchKinderenResultaatResponse xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:returnCode>0005</ns2:returnCode>
  <ns2:procentVerwerkt>0</ns2:procentVerwerkt>
</ns2:matchKinderenResultaatResponse>
```

Verwerkt:

```
<ns2:matchKinderenResultaatResponse xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:returnCode>0000</ns2:returnCode>
  <ns2:kindOutput>
    <ns2:kindInput>
      <ns2:naam>naam kind1</ns2:naam>
      <ns2:voornaam>voornaam kind1</ns2:voornaam>
      <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
      <ns2:kindCode>12091000703</ns2:kindCode>
    </ns2:kindInput>
    <ns2:naam>naam kind1</ns2:naam>
    <ns2:voornaam>voornaam kind1</ns2:voornaam>
    <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
    <ns2:kindCode>120910-007</ns2:kindCode>
    <ns2:tariefCode>120910-007-03</ns2:tariefCode>
    <ns2:inkomenstarief>5.00</ns2:inkomenstarief>
    <ns2:geldigVan>2015-01-01+01:00</ns2:geldigVan>
    <ns2:geldigTot>2015-12-31+01:00</ns2:geldigTot>
    <ns2:opmerkingCode>1000</ns2:opmerkingCode>
    <ns2:referentieDatum>2016-02-16+01:00</ns2:referentieDatum>
  </ns2:kindOutput>
  <ns2:kindOutput>
    <ns2:kindInput>
      <ns2:naam>naam kind2</ns2:naam>
      <ns2:voornaam>voornaam kind2</ns2:voornaam>
      <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
      <ns2:kindCode>12091000803</ns2:kindCode>
    </ns2:kindInput>
    <ns2:naam>naam kind2</ns2:naam>
    <ns2:voornaam>voornaam kind2</ns2:voornaam>
    <ns2:geboorteDatum>2012-09-10+02:00</ns2:geboorteDatum>
    <ns2:kindCode>120910-008</ns2:kindCode>
    <ns2:tariefCode>120910-008-03</ns2:tariefCode>
    <ns2:inkomenstarief>5.00</ns2:inkomenstarief>
    <ns2:geldigVan>2015-01-01+01:00</ns2:geldigVan>
    <ns2:geldigTot>2015-12-31+01:00</ns2:geldigTot>
    <ns2:opmerkingCode>1000</ns2:opmerkingCode>
    <ns2:referentieDatum>2016-02-16+01:00</ns2:referentieDatum>
  </ns2:kindOutput>
  <ns2:procentVerwerkt>100</ns2:procentVerwerkt>
</ns2:matchKinderenResultaatResponse>
```

4.10.2 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
returnCode	int	Max lengte: 4	1	1
kindOutput	kindOutput	Zie (4.14)	0	~
procentVerwerkt	int	Getal tussen 0 en 100	1	1

4.11 Message Payload voor OpvangPrestatiesResultaat

4.11.1 Waarvoor wordt dit gebruikt?

Laat toe om het resultaat van de verwerking van de opvangprestaties op te vragen die zijn doorgegeven vanaf een bepaald tijdstip (datumVanaf).

4.11.2 Voorbeeld xml

```
<ns2:opvangPrestatiesResultaatRequest
xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:ondernemingsNummer>0123456789</ns2:ondernemingsNummer>
  <ns2:datumVanaf>2015-06-01+02:00</ns2:datumVanaf>
</ns2:opvangPrestatiesResultaatRequest>
```

4.11.3 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
ondernemingsNummer	int	Max lengte: 10	1	1
datumVanaf	date		1	1

4.12 OpvangPrestatiesResultaat response

4.12.1 Voorbeeld xml

```
<ns2:opvangPrestatiesResultaatResponse
xmlns:ns2="http://www.kindengezin.be/kinderopvang/webservice">
  <ns2:returnCode>0000</ns2:returnCode>
  <ns2:opvangPrestatiesResultaat>
    <ns2:locatieId>910000001</ns2:locatieId>
    <ns2:maand>2015-03-01+01:00</ns2:maand>
    <ns2:openingsDagen>18</ns2:openingsDagen>
    <ns2:statusVerwerking>VERWERKT</ns2:statusVerwerking>
    <ns2:foutVolgNummer>0</ns2:foutVolgNummer>
    <ns2:geregistreerdOp>2015-06-03+02:00</ns2:geregistreerdOp>
  </ns2:opvangPrestatiesResultaat>
  <ns2:opvangPrestatiesResultaat>
    <ns2:locatieId>910000002</ns2:locatieId>
    <ns2:maand>2015-04-01+02:00</ns2:maand>
    <ns2:statusVerwerking>GEEN AANGIFTE</ns2:statusVerwerking>
    <ns2:foutVolgNummer>0</ns2:foutVolgNummer>
  </ns2:opvangPrestatiesResultaat>
  <ns2:opvangPrestatiesResultaat>
    <ns2:locatieId>910000003</ns2:locatieId>
    <ns2:maand>2015-03-01+01:00</ns2:maand>
    <ns2:statusVerwerking>NIET VERWERKT</ns2:statusVerwerking>
    <ns2:foutVolgNummer>1</ns2:foutVolgNummer>
    <ns2:foutBoodschap>De locatie (910000003) heeft in de opgegeven maand 03/2015 geen recht
om opvangprestaties door te geven.</ns2:foutBoodschap>
    <ns2:geregistreerdOp>2015-06-04+02:00</ns2:geregistreerdOp>
  </ns2:opvangPrestatiesResultaat>
  <ns2:opvangPrestatiesResultaat>
    <ns2:locatieId>910000004</ns2:locatieId>
    <ns2:maand>2015-03-01+01:00</ns2:maand>
    <ns2:statusVerwerking>NIET VERWERKT</ns2:statusVerwerking>
    <ns2:kindCode>100101-001</ns2:kindCode>
```

```

<ns2:foutVolgNummer>1</ns2:foutVolgNummer>
<ns2:foutBoodschap>Voor Naam Voornaam (100101-001) zijn er gegevens teruggevonden in onze databank, maar er is geen attest gevonden dat geldig is in de opgegeven maand (03/2015).</ns2:foutBoodschap>
<ns2:geregistreerdOp>2015-06-03+02:00</ns2:geregistreerdOp>
</ns2:opvangPrestatiesResultaat>
...
</ns2:opvangPrestatiesResultaatResponse>

```

4.12.2 Elementen

Element	Type	Eigenschappen	Minimum aantal	Maximum aantal
returnCode	int	Max lengte: 4	1	1
opvangPrestatiesResultaat	opvangPrestatiesResultaat	Zie (4.15)	0	~

4.13 element KindInput

Element	Type	Omschrijving	Minimum aantal	Maximum aantal
naam	string	Max lengte: 50	1	1
voornaam	string	Max lengte: 50	1	1
geboorteDatum	date		1	1
kindCode	string	Max lengte: 13	0	1

4.14 element KindOutput

Element	Type	Omschrijving	Minimum aantal	Maximum aantal
kindInput	kindInput (zie 4.13)	De originele input	1	1
naam	string	Max length: 50	1	1
voornaam	string	Max length: 50	0	1
geboorteDatum	date		0	1
kindCode	string	Max length: 13	0	1
tariefCode	string	Max length: 13	0	1
inkomenstarief	double		0	1

geldigVan	date		0	1
geldigTot	date		0	1
opmerkingCode	string	Max length: 4 Voor mogelijke codes, zie hieronder (4.14.1)	1	1
opmerking	string	Max length: 200	0	1
referentieDatum	date		0	1

4.14.1 mogelijke opmerking codes in KindOutput

opmerkingCode	Omschrijving
1000	kindcode gematched
1001	attest afgekeurd
1002	kindcode YYMMDD-999 gevonden, maar naam/voornaam komt niet overeen
1003	geen geldig attest op DD-MM-YYYY
1004	niet gevonden
1005	Aanvraag geregistreerd maar nog niet behandeld

4.15 element OpvangPrestatiesResultaat

Element	Type	Omschrijving	Minimum aantal	Maximum aantal
locatieId	int	De originele input	1	1
maand	date		1	1
openingsDagen	int		0	1
statusVerwerking	string		1	1
kindCode	kindCode	Max lengte: 13	0	1
foutVolgNummer	long		0	1
foutBoodschap	string	Max lengte: 2000	0	1
geregistreerdOp	date		0	1

5 FOUTMELDINGEN

Hieronder volgt een voorlopige oplistng van de foutmeldingen. Deze wordt mogelijks later vervolledigd.

5.1 Mogelijke return codes in xml responses

returnCode	Omschrijving
0000	OK (request werd succesvol afgehandeld)
0002	Marshalling fout
0003	Firewall fout
0004	Het ondernemingsnummer heeft geen rechten om de webservice te gebruiken
0005	Aanvraag is in verwerking
9999	Er is een ongekennde fout opgetreden

5.2 Security validatie

Foutmelding:

Internal Server Error [500]

Oplossing:

- Controleer of uw het correcte certificaat gebruikt voor de correcte omgeving (test-certificaat in testomgeving; productiecertificaat in productie-omgeving)
- Controleer of er in de payload geen overbodige tabs en spaties zijn gebruikt.
- Controleer of in de security header alle onderdelen aanwezig zijn zoals beschreven in hoofdstuk 4.2.2: Ws-Security in IKG WebServices
- Neem contact op met Software-ontwikkeling@kindengezin.be en stuur het request mee dat uw doorgestuurd hebt. Dit met inbegrip van het volledige certificaat (*BinarySecurityToken*)

5.3 Validatie fout in request xml

Foutmelding:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring xml:lang="nl">De request werd geblokkeerd door de firewall. VerzoekID:
13794390277028862548</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Oplossing:

- Controleer of uw bericht alle benodigde elementen bevat en valideer het aan de hand van de WSDL.
- Neem contact op met Software-ontwikkeling@kindengezin.be en stuur het request mee dat uw doorgestuurd hebt, samen met het verzoek ID dat vermeld werd in de response.

6 VOORBEELD APPLICATIE

6.1 Beschikbaarheid

Er is een voorbeeld Spring applicatie beschikbaar. Deze applicatie en de bron code kan men opvragen door een mail te sturen naar Software-ontwikkeling@kindengezin.be.

6.2 Gebruik van de voorbeeld client applicatie

Onze voorbeeld applicatie maakt gebruik van java, maven en spring. Om de client applicatie te kunnen uitvoeren dienen java en maven correct geïnstalleerd te zijn.

In de home folder van de ingelogde gebruiker maak je volgende properties file aan:
`%UserProfile%\kinderopvang-ikg-webservice\config\kinderopvang-ikg-ws.properties`

Met daarin deze parameters

<code>org.keystore.file=</code>		<code># pad naar het keystore bestand</code>
<code>org.keystore.password=</code>		<code># het wachtwoord van het keystore bestand</code>
<code>org.cert.alias</code>	<code>=</code>	<code># alias van het geïmporteerde eHaelth certificaat</code>
<code>org.cert.password</code>	<code>=</code>	<code># wachtwoord van het geïmporteerde eHaelth certificaat</code>

7 REFERENTIES

7.1 Certificaat via Kind en Gezin

- Algemene voorwaarden
<http://www.kindengezin.be/img/AlgemeneVoorwaarden-ov-certificaat.pdf>
- Handleiding Keystore Explorer
<http://www.kindengezin.be/img/HandleidingKeystoreExplorer.pdf>
- Volmactsformulier
<http://www.kindengezin.be/img/VolmachtformulierCertificaataanvraag-vo.docx>

7.2 Web-service Security

- Spring Web Services (Java) : <http://static.springsource.org/spring-ws/site/reference/html/index.html>
- Oasis X509 Specs: <http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf>

8 CONTACTGEGEVENS

8.1 Contactgegevens voor de IKG Webservice

U kan een mail sturen naar software-ontwikkeling@kindengezin.be.

- Voor het verkrijgen van toegang tot de webservice
- Met functionele of technische vragen