

DOKUMENTACE KNIHOVNY PORTÁL ZP C#.NET

**ŘEŠENÍ A SPRÁVA
INTERNETOVÉHO PORTÁLU**

ZDRAVOTNÍCH POJIŠŤOVEN

**říjen 2022 – srpen 2023
Verze 1.4**

**Asseco Central Europe, a.s.
Copyright © 2023 Asseco Central Europe, a.s.**

Autor: Pavel Koukal

Obsah

1.	<i>Popis knihovny.....</i>	3
2.	<i>Licence.....</i>	3
3.	<i>Závislosti knihovny.....</i>	4
4.	<i>Rozhraní třídy PzpKomunikace.....</i>	4
4.1	<i>Datové struktury.....</i>	4
4.1.1	SWInfoRec.....	4
4.1.2	PzpAuthTyp.....	5
4.1.3	PzpAuthRec.....	5
4.1.4	PzpKomunikaceFormatOdpovedi.....	5
4.1.5	PzpKomunikaceRawData.....	5
4.1.6	PzpKomunikaceOdpoved.....	5
4.1.7	PzpKomunikaceResult.....	6
4.2	<i>Vlastnosti.....</i>	7
4.2.1	Timeout.....	7
4.2.2	PosledniResult.....	7
4.3	<i>Události.....</i>	7
4.3.1	ZahajenoOdesilani.....	7
4.3.2	UkoncenoOdesilani.....	7
4.4	<i>Metody.....</i>	7
4.4.1	Konstruktor třídy PzpKomunikace.....	8
4.4.2	OdesliPozadavek.....	8
4.4.3	OdesliSMSAutentizaci.....	8
4.4.4	VratSeznamSchraneK.....	8
4.4.5	VratSeznamZpravSchranky.....	9
4.4.6	VratDetailZpravySchranky.....	9
4.4.7	OznacZpravuSchrankyJakoPrectenou.....	10
4.4.8	SmazZpravuSchranky.....	10
5.	<i>Průběh komunikace.....</i>	10
5.1	<i>Sestavení a podepsání XML výzvy.....</i>	11
5.2	<i>Odeslání XML výzvy na klientskou bránu ZP.....</i>	12
5.3	<i>Přijetí XML odpovědi a její zpracování.....</i>	12
5.4	<i>Ověření podpisu odpovědi serveru.....</i>	12
6.	<i>Příklady použití knihovny v .NET projektech.....</i>	12
6.1	<i>Synchronní volání metod.....</i>	12
6.1.1	Odeslání SMS autentizace.....	12
6.1.2	Odeslání podání podepsané SMS kódem.....	13
6.1.3	Odeslání podání podepsané certifikátem.....	13
6.2	<i>Asynchronní volání metod s využitím událostí.....</i>	13
6.2.1	Odeslání SMS autentizace.....	13
6.2.2	Odeslání podání podepsané SMS kódem.....	13
6.2.3	Odeslání podání podepsané certifikátem.....	14
7.	<i>Chybové stavy.....</i>	14
8.	<i>Debug výstup knihovny.....</i>	15

1. Popis knihovny

Knihovna Portál ZP C#.NET (název souboru PortalZPCsNet.dll) rozšiřuje již existující knihovnu PortalZP.dll o možnost ji využít přímo v .NET projektech a je určena pro implementaci komunikace softwaru třetích stran s portálem zdravotních pojišťoven (<https://www.portalzp.cz>), a to pro klienty tohoto portálu formou komunikační brány. Vlastní komunikace s portálem je popsána v Technickém řešení č. 6, které je k dispozici na https://www.portalzp.cz/distribuce.ext/PZP-kom_brana_klient/. Knihovna obsahuje jednu .NET třídu PzpKomunikace, která je určena pro komunikaci s klientskými komunikačními bránami portálu ZP, a lze ji využít pro odeslání souborů:

- vyúčtování plateb (VYU)
- hromadného odeslání zaměstnavatele (HOZ)
- přehledu plateb pojistného zaměstnavatele (PPPZ)
- ověření pojištěnce (VERPOJ)
- ověření zdravotnického zařízení (VERZZ)
- odeslání registračních lístků (ORL)
- přesměrování (PRESMER)
- dotaz na schránky klienta (SCHRANKA)
- dotaz exekutora na právnickou osobu (EXE_PO)
- dotaz exekutora na fyzickou osobu (EXE_FO)
- prvotní import N Příloh 2 (P2_IMPORT)
- zavedení nového certifikátu (NOVY_CERT)
- formuláře schránky (SCHRANKA_PODANI)
- žádost Celní správy o bezdlužnost (BZD)
- hlášení Policie o dopravních nehodách (HDN)

Tyto soubory či dotazy lze odeslat jak do pěti pojišťoven na Portálu ZP (ČPZP, OZP, RBP, ZPŠ a VoZP ČR), tak do společné zóny všech pojišťoven (PZP) nebo do pilotního prostředí projektu Portál ZP a jeho společné zóny (PILOT, PILOT-PZP).

2. Licence

Copyright (c) 2005-2023 Asseco Central Europe, a.s.

Copyright (c) 2021-2023 Svaz zdravotních pojišťoven ČR, z.s.

Copyright (c) 2005-2023 Česká průmyslová zdravotní pojišťovna

Copyright (c) 2005-2023 Oborová zdravotní pojišťovna

Copyright (c) 2005-2023 RBP, zdravotní pojišťovna

Copyright (c) 2007-2023 Vojenská zdravotní pojišťovna České republiky

Copyright (c) 2005-2023 Zaměstnanecká pojišťovna Škoda

Copyright (c) 2023 Zdravotní pojišťovna ministerstva vnitra České republiky

Všechna práva vyhrazena.

Šíření a využití je dovoleno za těchto podmínek:

1. Tato knihovna je dána k dispozici zdravotním pojišťovnám účastnícím se projektu Portál ZP, výrobcům SW vyvíjejícím software pro komunikaci s tímto Portálem a tím i klientům využívajícím software těchto výrobců SW.
2. Jakékoli zásahy do binární verze knihovny mimo Asseco Central Europe, a.s. jsou nepřipustné.
3. Závazným pro vlastní komunikaci je jen Technické řešení č. 6 (POPIS KOMUNIKACE S PORTÁLEM (KLIENT)), nikoliv tato knihovna.

TENTO SOFTWARE JE POSKYTOVÁN SPOLEČNOSTÍ Asseco Central Europe, a.s. "TAK JAK JE" A VEŠKERÉ ZÁRUKY A PRÁVA NA POUŽITÍ, OBCHODOVATELNOST, ZPŮSOBILOST K URČITÉMU ÚČELU JSOU VYHRAZENY. SPOLEČNOST Asseco Central Europe, a.s. NENÍ V ŽÁDNÉM PŘÍPADĚ ZODPOVĚDNÁ ZA JAKÉKOLIV PŘÍMÉ, NEPŘÍMÉ, NÁHODNÉ ČI NÁSLEDNÉ POŠKOZENÍ ZAPŘÍČINUJÍCÍ ZTRÁTU DAT, ZISKU NEBO PODOBNÉ ZPŮSOBENÉ JAKÝMKOLIV POUŽÍVÁNÍM SOFTWARE.

3. Závislosti knihovny

Knihovna Portál ZP C#.NET je složena pro dvě verze .NET:

- a) pro .NET Framework 4.5.2
- b) pro .NET 6

Obě verze jsou určeny pouze pro Windows systémy kvůli závislosti na DLL knihovně Signer.dll/Signer64.dll v její minimální verzi 5.2. Tato závislá knihovna Signer.dll (či Signer64.dll) slouží pro kryptografické operace, mimo jiné je použita pro podepisování dat pomocí Windows CryptoAPI a následném ověření podpisu odpovědi z Portálu ZP.

Knihovna Signer.dll (či Signer64.dll) je využita jako standardní Windows DLL knihovna a je nutné, aby byla dostupná buď přímo ve stejném adresáři s touto knihovnou nebo v adresáři dostupném systému pomocí proměnné prostředí PATH.

Knihovna Portál ZP C#.NET dle cílové architektury (32bit/64bit) volá buď 32bit knihovnu Signer.dll nebo 64bit knihovnu Signer64.dll. Při distribuci koncového řešení záleží tedy na zvolené architektuře a pokud bude použito jen 64bit sestavení, nemusí se 32bit knihovna Signer.dll distribuovat s koncovou aplikací a naopak.

Upozornění: Knihovny Signer.dll a Signer64.dll je možné licenčně využít pouze s touto knihovnou PortalZPCsNet.dll jakožto závislou knihovnou. Pro použití Signer.dll nebo Signer64.dll mimo účely této knihovny je nutné získat licenci od firmy Asseco Central Europe, a.s.

4. Rozhraní třídy PzpKomunikace

4.1 Datové struktury

4.1.1 SWInfoRec

Datová struktura pro předání informací o softwaru, který knihovnu Portál ZP C#.NET využívá.

Informace jsou využity k případnému řešení problému s klientem Portálu ZP. Stává se totiž, že je sice zalogována XML výzva pro komunikační bránu klienta, nicméně dotyčný klient pak vznese dotaz, po jehož analýze dospěje technická podpora Portálu ZP k závěru, že je to vlastně dotaz na technickou podporu výrobce příslušného SW, nicméně ne vždy je do XML výzvy vložen komentář s identifikací daného SW a / nebo příslušné technické podpory.

Následující údaje se vztahují k software, který knihovnu využívá a údaje se předávají do konstruktoru instance třídy PzpKomunikace. Všechny údaje jsou nepovinné (mohou být předány prázdné řetězce), ale doporučujeme je vyplnit z výše popsaných důvodů.

```
public struct SWInfoRec
{
    public string Nazev;
    public string Verze;
    public string Vyrobc;
    public string Helpline;
}
```

Popis jednotlivých údajů:

Nazev	Název software, např. „Moje ordinace“
Verze	Verze software, např. „3.15“
Vyrobc	Výrobce software, např. „LekarSoft a.s.“
Helpline	Kontakt na helpline daného software a výrobce. Jedná se o volný text, např: „web: http://www.lekarsoft.cz , email: podpora@lekarsoft.cz , tel: 777 123 123“

4.1.2 PzpAuthTyp

Typ autentizace, která má být použita při komunikaci s Portálem ZP. Aktuálně je možné použít buď certifikát nebo SMS login+kód. Typ autentizace nabývá jednu z následujících hodnot:

```
public enum PzpAuthTyp { CERT, SMS };
```

4.1.3 PzpAuthRec

Struktura pro přenos autentizačních údajů. V případě AuthTyp = PzpAuthTyp.CERT jsou údaje SmsLogin a SmsKod ignorovány.

```
public struct PzpAuthRec
{
    public PzpAuthTyp AuthTyp;
    public string SmsLogin;
    public string SmsKod;
}
```

Popis jednotlivých údajů:

AuthTyp	Typ autentizace – viz enum <i>PzpAuthTyp</i>
SmsLogin	SMS přihlašovací jméno v případě AuthTyp = PzpAuthTyp.SMS
SmsKod	Platný SMS kód zasláný na mobilní telefon registrovaný u klientského konta, který byl zaslán klientovi na základě žádosti o SMS autentizaci pomocí metody <i>OdesliSMSAutentizaci</i> . Je vyžadován pouze v případě AuthTyp = PzpAuthTyp.SMS

4.1.4 PzpKomunikaceFormatOdpovedi

Portál ZP nebo přímo knihovna během zpracování vrací odpověď v různých formátech. Buď se jedná o textovou odpověď, nebo Portál ZP odpoví strukturovaným XML či to může být HTML stránka s protokolem. Knihovna pomocí analýzy hlaviček a výstupních dat stanoví, o jaký formát odpovědi se jedná pro lepší zpracování na straně výrobce software. Formát odpovědi nabývá jednu z následujících hodnot:

```
public enum PzpKomunikaceFormatOdpovedi { TEXT, XML, HTML };
```

4.1.5 PzpKomunikaceRawData

Vzhledem k rozmanitým možným kódováním jak požadavku, tak odpovědi se v rámci zachování originálního kódování předávají některé obsahy dat jako dvojice údajů – kódování a pole bytů. Tak je možné na straně výrobce SW uložit XML odpověď v originálním kódování či jinak pracovat se surovými daty. Pro přenos této dvojice údajů se využívá struktura PzpKomunikaceRawData:

```
public struct PzpKomunikaceRawData
{
    public byte[] Data;
    public Encoding Kodovani;
}
```

4.1.6 PzpKomunikaceOdpoved

Obalující struktura pro jednu odpověď Portálu ZP. Odpovědi může být v některých případech více, tedy tato struktura obaluje potřebné údaje vztažené vždy k jedné odpovědi.

```
public struct PzpKomunikaceOdpoved
{
    public string NazevDat;
    public PzpKomunikaceRawData DataOdpovedi;
}
```

```
    public PzpKomunikaceFormatOdpovedi FormatOdpovedi;
}
```

Popis jednotlivých údajů:

NazevDat	Název dat uvedený v elementu a atributu Soubor.Jmeno v dané odpovědi Portálu ZP
DataOdpovedi	Struktura <i>PzpKomunikaceRawData</i> s jakoukoliv odpovědí, tj. např. odpověď z Portálu ZP, chybové hlášení podepisovací komponenty Signer či přímo text chyby knihovny Portál ZP C#.NET.
FormatOdpovedi	Detekovaný formát odpovědi – viz enum <i>PzpKomunikaceFormatOdpovedi</i> .

4.1.7 PzpKomunikaceResult

Výsledek volání komunikace s Portálem ZP je předáván v následující datové struktuře:

```
public struct PzpKomunikaceResult
{
    public int IdPodani;
    public string KodChyby;
    public PzpKomunikaceOdpoved[] Odpoved;
    public string Varovani;
    public string DebugInterniVarovani;
    public PzpKomunikaceRawData DebugXmlVyzva;
    public PzpKomunikaceRawData DebugXmlOdpoved;
    public string PodpisCertSN;
    public string PodpisCertIDN;
}
```

Popis jednotlivých údajů:

IdPodani	Číslo uloženého podání, došlo-li ke komunikaci s portálem, jinak nula.
KodChyby	Kód případné chyby. Nemusí se jednat vždy o číselný kód, jsou přípustné i řetězcové hodnoty. "0"=OK, více v seznamu chyb v kapitole 9 a v Technickém řešení č. 6
Odpoved	Pole struktur <i>PzpKomunikaceOdpoved</i> s jakoukoliv odpovědí, tj. např. odpověď z Portálu ZP, chybové hlášení podepisovací komponenty Signer či přímo text chyby knihovny Portál ZP C#.NET. Standardně je odpověď z Portálu pouze jedna, tedy pole bude jednorázové. Ve speciálních případech a pouze u služby SCHRANKA může Portál ZP odeslat více odpovědí a potom toto pole bude naplněno všemi rozparovanými odpovědi.
Varovani	Případný text varování, který je určen pro zobrazení klientovi, např. že odpověď ze serveru není podepsána certifikátem, který je na serveru vystaven, či je podpis neplatný.
DebugInterniVarovani	Případné interní varování knihovny určené pro výrobce SW.
DebugXmlVyzva	Struktura <i>PzpKomunikaceRawData</i> s XML výzvou, která byla poslána na Portál ZP. Tato výzva je vždy v kódování UTF-8 a předané vstupní soubory jsou předávány do Portálu ZP v Base64 kódování.
DebugXmlOdpoved	Struktura <i>PzpKomunikaceRawData</i> s XML odpovědí, pokud byly splněny všechny kontroly a došlo k přijetí odpovědi z Portálu ZP. Tato XML odpověď může být v různém kódování, nejčastěji bývá ISO-8859-2.
PodpisCertSN	Sériové číslo certifikátu, jímž byla podepsána odpověď serveru Portálu ZP.
PodpisCertIDN	IDN (DN položky vystavitele) certifikátu, jímž byla podepsána odpověď serveru Portálu ZP.

4.2 Vlastnosti

4.2.1 Timeout

Timeout komunikace typu Integer s defaultní hodnotou 300 sekund (5 minut). Zadává se počet sekund, po kterých bude čekání na odpověď serveru Portálu ZP přerušeno a bude vrácena chyba timeoutu. Vzhledem k tomu, že je na server odeslán HTTPS požadavek a bude přerušeno jen čekání na odpověď, s největší pravděpodobností zpracování na serveru Portálu ZP bude dokončeno a o výsledku se klient nedozví.

V případě nastavení hodnoty pro timeout menší než je 10, knihovna přenastaví hodnotu na defaultních 300.

V případě odesílání velkých dat v řádu jednotek MB je vhodné zvětšit úměrně hodnotu timeoutu, aby se do hodnoty timeoutu podařilo velká data nejen po síti odeslat, ale server je i stačil zpracovat a odeslat odpověď.

4.2.2 PosledniResult

PosledniResult je plněn strukturou *PzpKomunikaceResult* po každém volání jakékoliv komunikační metody. Jedná se o stejnou hodnotu, jaká je vrácena jako result po volání dané komunikační metody. Jde o poslední result, který knihovna naposledy poskytla.

4.3 Události

4.3.1 ZahajenoOdesilani

Standardní EventHandler (*object* sender, *EventArgs* e) pro možnost zapojení své obsluhy v momentě, kdy se začne odesílat XML výzva na server Portálu ZP. Událost je vyvolána až na začátku samotné komunikace, tedy pokud nastane nějaká chyba vstupních dat nebo klient např. podání certifikátem nepodepíše, nedojde ani k samotné komunikaci a událost není vyvolána.

V této události je vhodné např. zobrazit formulář s informací, že probíhá odesílání dat či nějaký svůj vizuální progress bar apod.

4.3.2 UkoncenoOdesilani

Standardní EventHandler (*object* sender, *EventArgs* e) pro možnost zapojení své obsluhy v momentě, kdy celá komunikace včetně ověření podpisu a parsování odpovědi byla ukončena. Jedná se o událost těsně před vrácením výsledku volání metody, ovšem jedná se o párovou událost k *ZahajenoOdesilani*, tj. není-li provolána událost *ZahajenoOdesilani* z důvodů popsaných v 4.3.1, potom ani tato událost není po skončení komunikace provolána.

V této události je vhodné zavřít formulář s průběhem či provést finalizační akce započaté v události *ZahajenoOdesilani*.

4.4 Metody

Pro běžnou komunikaci s Portálem ZP se používá metoda *OdesliPozadavek*, která specifikuje, jaká služba se volá a s jakými vstupními soubory. Pro prvotní SMS autentizaci a získání SMS kódu se používá metoda *OdesliSMSAutentizaci*. Všechny tyto metody vracejí vždy pouze jednu odpověď a je možné je využít i pro práci se schránkami Portálu ZP pomocí služby *SCHRANKA*.

V případě využití vícenásobných dotazů na schránky Portálu ZP dle dokumentace „Datové rozhraní pro přístup k datům ve schránkách“ (P4ZP-PHP_APP_SCH_P01.doc) je výhodnější použít specializované metody *VratSeznamSchraneK*, *VratSeznamZpravSchrany*, *VratDetailZpravySchrany*, *OznacZpravuSchranyJakoPrectenou* či *SmazZpravuSchrany*. Tyto metody zjednodušují práci se schránkami v následujících bodech:

- není nutné znát datové rozhraní ani názvy požadovaných vstupních XML souborů
- v případě komunikace pomocí metody *OdesliPozadavek* a odeslání většího počtu požadavků, např. přečtení 10 zpráv, je nutné vytvořit 10 dočasných souborů v 10 dočasných adresářích (kvůli shodě názvu souborů) a předání 10 plných cest k souborům do vstupního parametru *vstupniSoubory* metody *OdesliPozadavek*. V případě použití těchto

specializovaných metod pro práci se schránkou odpadá nutnost vytváření dočasných souborů a adresářů.

4.4.1 Konstruktor třídy PzpKomunikace

Konstruktor třídy PzpKomunikace očekává na vstupu předaný jediný parametr swInfoRec struktury SWInfoRec – viz kapitola 4.1.1

```
public PzpKomunikace(SWInfoRec swInfoRec)
```

4.4.2 OdesliPozadavek

Metoda sestává ze vstupních parametrů XML požadavek dle datového rozhraní. Pokud se jedná o CERT autentizaci, nechá klientem podepsat požadavek pomocí klientského certifikátu. Následně odešle připravený požadavek na zvolený server Portálu ZP určený dle parametru ZP. Výsledek komunikace ve formě struktury PzpKomunikaceResult vrátí jako result volání této metody.

```
public PzpKomunikaceResult OdesliPozadavek(PzpAuthRec auth, string zp, string sluzba, string vstupniSoubory)
```

Parametry metody:

auth	Struktura PzpAuthRec s autentizačními údaji udávajícími zda se podání podepíše certifikátem nebo budou použity údaje pro SMS autentizaci – viz kapitola 4.1.3
zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot: CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
sluzba	Zkratka služby, o jejíž podání se jedná. Jedna z hodnot: VYU, HOZ, PPPZ, VERPOJ, VERZZ, ORL, PRESMER, SCHRANKA, EXE_PO, EXE_FO, P2_IMPORT, NOVY_CERT, SCHRANKA_PODANI, BZD, HDN
vstupniSoubory	Plné cesty ke vstupním souborům oddělené od sebe znakem " "

4.4.3 OdesliSMSAutentizaci

Metoda sestává ze vstupních parametrů XML žádost o SMS autentizaci pomocí přihlašovacího jména a hesla. Po úspěšné autentizaci Portálem ZP je zaslán SMS kód na zaregistrovaný mobilní telefon u daného uživatelského konta pro další použití v rámci SMS údajů ve struktuře PzpAuthRec, která se předává jako autentizační parametr do metod OdesliPozadavek, VratSeznamSchraneK, VratSeznamZpravSchranky, VratDetailZpravySchranky, OznacZpravuSchrankyJakoPrectenou či SmazZpravuSchranky.

Platnost tohoto SMS kódu je omezena na 30 minut a během této doby nelze zažádat Portál ZP o nový SMS kód pro stejného klienta.

Výsledek komunikace ve formě struktury PzpKomunikaceResult vrátí jako result volání této metody.

```
public PzpKomunikaceResult OdesliSMSAutentizaci(string zp, string smsLogin, string smsHeslo)
```

Parametry metody:

zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot: CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
smsLogin	Přihlašovací jméno pro SMS autentizaci evidované u klientského konta v Portálu ZP.
smsHeslo	Přihlašovací heslo pro SMS autentizaci uživatele nebo prázdný řetězec, pokud klient heslo nepoužívá.

4.4.4 VratSeznamSchraneK

Metoda sestaví požadavek dle datového rozhraní pro získání seznamu všech schránek, ke kterým má klient přístup u dané zdravotní pojišťovny. Pokud se jedná o CERT autentizaci, nechá klientem podepsat požadavek pomocí klientského certifikátu. Následně odešle připravený požadavek na zvolený server Portálu ZP určený dle parametru ZP. Výsledek komunikace ve formě struktury *PzpKomunikaceResult* vrátí jako result volání této metody. Výsledkem je vždy jedna odpověď.

```
public PzpKomunikaceResult VratSeznamSchraneK(PzpAuthRec auth, string zp)
```

Parametry metody:

auth	Struktura PzpAuthRec s autentizačními údaji udávajícími zda se podání podepíše certifikátem nebo budou použity údaje pro SMS autentizaci – viz kapitola 4.1.3
zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot: CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP

4.4.5 VratSeznamZpravSchranky

Metoda sestaví požadavek dle datového rozhraní pro získání seznamu zpráv schránek pro dané identifikátory schránek. Pokud se jedná o CERT autentizaci, nechá klientem podepsat požadavek pomocí klientského certifikátu. Následně odešle připravený požadavek na zvolený server Portálu ZP určený dle parametru ZP. Výsledek komunikace ve formě struktury *PzpKomunikaceResult* vrátí jako result volání této metody. V případě chybné odpovědi Portál ZP vrací jen jednu odpověď s chybou, v případě úspěšné odpovědi je počet odpovědí roven počtu prvků v parametru idSchranky.

```
public PzpKomunikaceResult VratSeznamZpravSchranky(PzpAuthRec auth, string zp, string[] idSchranky, bool pouzeNove)
```

Parametry metody:

auth	Struktura PzpAuthRec s autentizačními údaji udávajícími zda se podání podepíše certifikátem nebo budou použity údaje pro SMS autentizaci – viz kapitola 4.1.3
zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot: CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
idSchranky	Pole textových identifikátorů schránek, pro které mají být zprávy vráceny.
pouzeNove	Příznak, zda se mají vrátit pouze nové zprávy (nepřečtené) nebo všechny zprávy (nepřečtené i přečtené).

4.4.6 VratDetailZpravySchranky

Metoda sestaví požadavek dle datového rozhraní pro získání detailu požadovaných zpráv schránek pro dané identifikátory zpráv. Pokud se jedná o CERT autentizaci, nechá klientem podepsat požadavek pomocí klientského certifikátu. Následně odešle připravený požadavek na zvolený server Portálu ZP určený dle parametru ZP. Výsledek komunikace ve formě struktury *PzpKomunikaceResult* vrátí jako result volání této metody. V případě chybné odpovědi Portál ZP vrací jen jednu odpověď s chybou, v případě úspěšné odpovědi je počet odpovědí roven počtu prvků v parametru idZpravy.

```
public PzpKomunikaceResult VratDetailZpravySchranky(PzpAuthRec auth, string zp, Int32[] idZpravy)
```

Parametry metody:

auth	Struktura PzpAuthRec s autentizačními údaji udávajícími zda se podání podepíše certifikátem nebo budou použity údaje pro SMS autentizaci – viz kapitola 4.1.3
zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot:

	CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
idZpravy	Pole číselných identifikátorů zpráv schránek, pro které mají být detaily vráceny.

4.4.7 OznacZpravuSchrankyJakoPrectenou

Metoda sestaví požadavek dle datového rozhraní pro označení zpráv jako přečtené pro dané identifikátory zpráv. Pokud se jedná o CERT autentizaci, nechá klientem podepsat požadavek pomocí klientského certifikátu. Následně odešle připravený požadavek na zvolený server Portálu ZP určený dle parametru ZP. Výsledek komunikace ve formě struktury *PzpKomunikaceResult* vrátí jako result volání této metody. V případě chybné odpovědi Portál ZP vrací jen jednu odpověď s chybou, v případě úspěšné odpovědi je počet odpovědí roven počtu prvků v parametru idZpravy.

```
public PzpKomunikaceResult OznacZpravuSchrankyJakoPrectenou(PzpAuthRec auth, string zp, Int32[] idZpravy)
```

Parametry metody:

auth	Struktura PzpAuthRec s autentizačními údaji udávajícími zda se podání podepíše certifikátem nebo budou použity údaje pro SMS autentizaci – viz kapitola 4.1.3
zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot: CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
idZpravy	Pole číselných identifikátorů zpráv schránek, které mají být označeny jako přečtené.

4.4.8 SmazZpravuSchranky

Metoda sestaví požadavek dle datového rozhraní pro označení zpráv jako smazané pro dané identifikátory zpráv. Pokud se jedná o CERT autentizaci, nechá klientem podepsat požadavek pomocí klientského certifikátu. Následně odešle připravený požadavek na zvolený server Portálu ZP určený dle parametru ZP. Výsledek komunikace ve formě struktury *PzpKomunikaceResult* vrátí jako result volání této metody. V případě chybné odpovědi Portál ZP vrací jen jednu odpověď s chybou, v případě úspěšné odpovědi je počet odpovědí roven počtu prvků v parametru idZpravy.

```
public PzpKomunikaceResult SmazZpravuSchranky(PzpAuthRec auth, string zp, Int32[] idZpravy)
```

Parametry metody:

auth	Struktura PzpAuthRec s autentizačními údaji udávajícími zda se podání podepíše certifikátem nebo budou použity údaje pro SMS autentizaci – viz kapitola 4.1.3
zp	Zkratka prostředí zdravotní pojišťovny bez interpunkce. Jedna z hodnot: CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
idZpravy	Pole číselných identifikátorů zpráv schránek, které mají být označeny jako smazané.

5. Průběh komunikace

Při komunikaci s Portálem ZP proběhne ověření klienta a to buď:

- na základě podpisu podání certifikátem klienta
- nebo pomocí přihlašovacího jména a hesla.

V případě podepsání certifikátem dochází na serveru k jednorázovému ověření klienta, jeho přihlášení, předání podání a následnému odhlášení klienta z Portálu ZP.

V případě přihlášení pomocí jména a hesla dojde na serveru k ověření uživatele a následně k jeho přihlášení do všech zdravotních pojišťoven v produkčním či pilotním prostředí – podle toho, která ZP

byla adresována. Poté je klientovi odeslána SMS zpráva na jeho mobilní telefon, kde je uveden SMS kód, který společně s přihlašovacím jménem slouží k ověření identity klienta. Platnost tohoto SMS kódu je časově omezena a po vypršení platnosti kódu je nutné se znovu autentizovat pomocí metody *OdesliSMSAutentizaci*. Během platnosti SMS kódu pro dané přihlašovací jméno odmítne Portál ZP poslat další SMS kód stejnému klientovi (přihlašovacímu jménu) a je nutné při volání metod *OdesliPozadavek*, *VratSeznamSchraneK*, *VratSeznamZpravSchranky*, *VratDetailZpravySchranky*, *OznacZpravuSchrankyJakoPrectenou* či *SmazZpravuSchranky* použít již dříve zaslaný SMS kód.

Celá komunikace s portálem ZP probíhá pomocí metod popsanych v kapitolách 4.4.X, kde je potřeba předat správně vstupní parametry, zejména *auth* a *zp*.

Pokud proběhnou v pořádku testy vstupních parametrů, začíná vlastní komunikace s klientskou bránou zdravotní pojišťovny předané v parametru *zp*. Pokud jsou data odesílaná na bránu zdravotní pojišťovny větší než 10kB, dojde k automatické kompresi za účelem zmenšení objemu přenášených dat.

5.1 Sestavení a podepsání XML výzvy

Z předaných vstupních parametrů je sestavena XML výzva, která se doplní buď podpisem nebo identifikací uživatele formou jména a SMS kódu. To záleží na předaném typu autentizace:

a) PzpAuthTyp.CERT

V případě předání autentizačního parametru s *PzpAuthTyp.CERT* musí klient podepsat XML výzvu certifikátem, který má zaveden v Portálu ZP u svého konta. XML výzva dále obsahuje přihlašovací řetězec, který je součástí předávaných dat a podepsáním celé XML výzvy dává klient souhlas k přihlášení se k Portálu ZP za účelem předání požadovaného podání (souborů).

Pokud operační systém podporuje hashovací funkci SHA-256, je podpis vytvořen s SHA-256 miniaturoou, v opačném případě s SHA-1 miniaturoou. Pokud je použita SHA-1, je tato informace vrácena ve výstupním parametru *PzpKomunikaceResult.DebugInterniVarovani*.

Upozornění

Zpráva:

Signer verze 5.2 © 2000-2013 Asseco Central Europe, a. s.

```

<Data Typ="POZADAVEK" Ucel="VYU">
<Prihlaseni UnikatniInfo="26.10.2022 14:31:18">
Tímto se přihlašuji k Portálu PILOT.
Současně předávám podání pro vyúčtování zdravotní péče v definovaném datovém rozhraní.
Po předání tohoto podání a příjmu odpovědi Portálu PILOT se z Portálu PILOT odhlašuji.
</Prihlaseni>
</Data>

```

§ SHA-256 miniaturoou:

1FA6 D993 709E F7B0 AE8E 656F EA31 16CE 3676 AADF 191B 636D 48C5 4FF8 A607 D954

Podepíšete certifikátem, který nepřipojíte k podepsané zprávě:

Změnit...

Pavel Koukal PZP [2030-12-31][44DBADFF28DD9953][Portal ZP Internal CA]

Nepřipojíte žádný další certifikát.

Zpráva nebude šifrována.

Přejete si tuto akci provést?

Áno

Ne

Storno

Pokud podepsání proběhne v pořádku (uživatel nestornuje podepsání), je celá XML výzva, která se posílá na server, uložena ve výstupním parametru *PzpKomunikaceResult.DebugXmlVyzva*.

b) PzpAuthTyp.SMS

V případě předání autentizačního parametru s *PzpAuthTyp.SMS* již musí být klient přihlášen k Portálu ZP a musí mít platný SMS kód. Pokud ještě není klient přihlášen (nemá SMS kód), je nutné jej přihlásit voláním metody *OdesliSMSAutentizaci*, kde předá svoje přihlašovací jméno a heslo a po úspěšném

ověření mu bude zaslána SMS zpráva s ověřovacím SMS kódem. Tento SMS kód a své přihlašovací jméno potom předá společně s dalšími vstupními parametry ostatním komunikačním metodám do struktury *PzpAuthTyp*, které je následně přidají do XML výzvy sestavené ze vstupních parametrů.

5.2 Odeslání XML výzvy na klientskou bránu ZP

Po sestavení a případném podepsání XML výzvy je tato výzva odeslána na klientskou bránu zdravotní pojišťovny uvedené ve vstupním parametru *zp*. Jedná se o standardní HTTPS komunikaci.

Během komunikace není zobrazen žádný průběh, je-li požadováno zobrazení průběhu, je možné se napojit na události třídy *PzpKomunikace* a zobrazovat si průběh odesílání dle svých preferencí.

Celá komunikace je také časově omezena počtem sekund uvedeným v property *TimeOut*. Toto omezení je defaultně nastavené na hodnotu 300, což je 5 minut. Více o timeoutu komunikace naleznete v kapitole 4.2.1.

5.3 Přijetí XML odpovědi a její zpracování

Po skončení komunikace očekává knihovna na výstupu odpověď ve formě XML, a pokud ji neobdrží, například z důvodu stornování podpisu XML výzvy, potom je na výstupu textové chybové hlášení. Stejně tak výstup může obsahovat HTML stránku s chybou, s hlášením odstávky Portálu ZP apod. Knihovna pomocí hlaviček a výstupních dat stanoví o jaký formát odpovědi se jedná (TEXT, XML, HTML), v jakém kódování byla poskytnuta odpověď a plní všechny výstupní atributy struktury *PzpKomunikaceResult* – viz kapitola 4.1.4.

5.4 Ověření podpisu odpovědi serveru

Pokud komunikace skončila v pořádku, tj. nevyskytla se chyba vstupních dat, uživatelské storno, chyba síťového připojení, timeout apod., a odpověď ze serveru je ve formě XML, knihovna ověří podpis této odpovědi a poskytne v *PzpKomunikaceResult.PodpisCertSN* a *PzpKomunikaceResult.PodpisCertIDN* identifikaci certifikátu, jímž byla odpověď Portálu ZP podepsána. Podpis se kontroluje proti certifikátu serveru, s kterým proběhla komunikace. Tento certifikát automaticky získá stažením ze serveru, s kterým komunikuje, z jeho https adresy https://.../portal_sign.pem.

V případě, že podpis je neplatný, generuje se do *PzpKomunikaceResult.Varovani* odpovídající text varování, který je určen pro koncového uživatele softwaru. Je dále na uvážení výrobce SW, zda danou identifikaci certifikátu bude nadále evidovat či provádět dodatečnou kontrolu, že certifikát s daným SN a IDN patří serveru Portálu ZP.

6. Příklady použití knihovny v .NET projektech

Následující příklady ukazují použití knihovny v C#.NET projektech. Funkční DEMO aplikace je k dispozici v distribučním balíku komponenty. Následující příklady předpokládají přidání reference na knihovnu *PortalZPCsNet* a použití jejího *namespace* ve kterém se třída *PzpKomunikace* nachází.

Dále je nutné naplnit a předávat strukturu *SWInfoRec* pro použití v konstruktoru třídy *PzpKomunikace*.

```
using Asseco.PortalZP;

internal SWInfoRec SWInfo = new SWInfoRec() {
    Nazev = "Software pro lékaře",
    Verze = "1.0",
    Vyrobce = "LekarSoft a.s.",
    HelpLine = "web: http://www.lekarsoft.cz, email: podpora@lekarsoft.cz, tel: 777 123 123"
    - volný text"
};
```

6.1 Synchronní volání metod

6.1.1 Odeslání SMS autentizace

```
PzpKomunikace pzpKomunikace = new PzpKomunikace(SWInfo);
```

```
PzpKomunikaceResult odpoved = pzpKomunikace.OdesliSMSAutentizaci("PILOT", "novak",  
"heslo");  
ZpracujOdpovedPortaluZP(odpoved);
```

6.1.2 Odeslání podání podepsané SMS kódem

```
PzpAuthRec auth = new PzpAuthRec() { AuthTyp = PzpAuthTyp.SMS, SmsLogin = "novak", SmsKod =  
"SMSKOD" };  
PzpKomunikace pzpKomunikace = new PzpKomunikace(SWInfo);  
PzpKomunikaceResult odpoved = pzpKomunikace.OdesliPozadavek(auth, "PILOT", "VYU", "c:\\  
FDAVKA.999|c:\\KDAVKA.999");  
ZpracujOdpovedPortaluZP(odpoved);
```

6.1.3 Odeslání podání podepsané certifikátem

```
PzpAuthRec auth = new PzpAuthRec() { AuthTyp = PzpAuthTyp.CERT, SmsLogin = "", SmsKod = ""  
};  
PzpKomunikace pzpKomunikace = new PzpKomunikace(SWInfo);  
PzpKomunikaceResult odpoved = pzpKomunikace.OdesliPozadavek(auth, "PILOT", "VYU", "c:\\  
FDAVKA.999|c:\\KDAVKA.999");  
ZpracujOdpovedPortaluZP(odpoved);
```

6.2 Asynchronní volání metod s využitím událostí

Následující příklady ukazují komplexnější kód s využitím asynchronního volání a událostí pro zobrazení vizuálního průběhu včetně využití možnosti přístupu k poslednímu resultu.

```
private frmProgress ProgressForm;  
  
private void onKomZahajeni(object sender, EventArgs e)  
{  
    if (InvokeRequired)  
        Invoke(new EventHandler(onKomZahajeni), new object[] { sender, e });  
    else {  
        ProgressForm = new frmProgress();  
        ProgressForm.Show();  
    }  
}  
  
private void onKomUkonceni(object sender, EventArgs e)  
{  
    if (InvokeRequired)  
        Invoke(new EventHandler(onKomUkonceni), new object[] { sender, e });  
    else {  
        ProgressForm.Close();  
    }  
}
```

6.2.1 Odeslání SMS autentizace

```
PzpKomunikace pzpKomunikace = new PzpKomunikace(SWInfo);  
pzpKomunikace.ZahajenoOdesilani += new System.EventHandler(this.onKomZahajeni);  
pzpKomunikace.UkoncenoOdesilani += new System.EventHandler(this.onKomUkonceni);  
await Task.Run(() => {  
    pzpKomunikace.OdesliSMSAutentizaci("PILOT", "novak", "heslo");  
});  
ZpracujOdpovedPortaluZP(pzpKomunikace.PosledniResult);
```

6.2.2 Odeslání podání podepsané SMS kódem

```
PzpAuthRec auth = new PzpAuthRec() { AuthTyp = PzpAuthTyp.SMS, SmsLogin = "novak", SmsKod =  
"SMSKOD" };  
PzpKomunikace pzpKomunikace = new PzpKomunikace(SWInfo);  
pzpKomunikace.ZahajenoOdesilani += new System.EventHandler(this.onKomZahajeni);  
pzpKomunikace.UkoncenoOdesilani += new System.EventHandler(this.onKomUkonceni);  
await Task.Run(() => {
```

```

        pzpKomunikace.OdesliPozadavek(auth, "PILOT", "VYU", "c:\\FDAVKA.999|c:\\KDAVKA.999");
    });
    ZpracujOdpovedPortaluZP(pzpKomunikace.PosledniResult);

```

6.2.3 Odeslání podání podepsané certifikátem

```

PzpAuthRec auth = new PzpAuthRec() { AuthTyp = PzpAuthTyp.CERT, SmsLogin = "", SmsKod = ""
};
PzpKomunikace pzpKomunikace = new PzpKomunikace(SWInfo);
pzpKomunikace.ZahajenoOdesilani += new System.EventHandler(this.onKomZahajeni);
pzpKomunikace.UkoncenoOdesilani += new System.EventHandler(this.onKomUkonceni);
await Task.Run(() => {
    pzpKomunikace.OdesliPozadavek(auth, "PILOT", "VYU", "c:\\FDAVKA.999|c:\\KDAVKA.999");
});
ZpracujOdpovedPortaluZP(pzpKomunikace.PosledniResult);

```

7. Chybové stavy

Chybové stavy jsou rozděleny do dvou skupin.

Jednu skupinu chyb vrací knihovna a jsou to chyby, které se staly při kontrole vstupních dat, podepisování či přímo při komunikaci s portálem ZP. Druhá skupina chyb jsou chyby, které vrací portál ZP při zpracování požadavku.

Popis chyb:

-1001	Neznámá zkratka ZP. ZP může být pouze jedna z hodnot CPZP, OZP, RBP, ZPS, VoZP, PZP, PILOT, PILOT-PZP
-1002	Neznámá služba. Služba může být pouze jedna z hodnot VYU, HOZ, PPPZ, VERPOJ, VERZZ, ORL, PRESMER, SCHRANKA, EXE_PO, EXE_FO, P2_IMPORT, NOVY_CERT, SCHRANKA_PODANI, BZD, HDN
-1003	Nenalezen předaný soubor ... Soubor předaný v parametru vstupniSoubory nebyl nalezen.
-1004	Vypršel časový limit XXX sekund.
-1005	Nastala chyba podepisovací komponenty Signer.
-1006	Nebylo přijato validní XML, obsahem výstupních parametrů Odpoved a DebugXmlOdpoved je blíže neurčená chyba (může být HTML např. o odstávce portálu, prostý text nebo nevalidní XML)
-1007	Chyba XML parseru při zpracovávání odpovědi
-1010	HTTP kód odpovědi nebyl 200, nejednalo se o validní odpověď XML brány
-1011	Server Portálu ZP není dostupný
-1099	Interní chyba knihovny
0	Zpracování skončilo v pořádku, výstupní data byla zpracována a uložena do výstupních parametrů. Podání je uloženo na serveru a v PzpKomunikaceResult.IdPodani je jeho číslo.
ostatní	Chybové stavy při zpracování podání na portálu ZP – validní odpověď stejně jako při nulové chybě, akorát nedojde ke správnému zpracování podání vlivem nesrovnalostí údajů ve vstupních datech (není oprávnění, chybná struktura dat, nebyly předány všechny informace, chyba kontrol vstupních dat apod.). Chyba není pouze číselného charakteru, ale jedná se o jakýkoliv alfanumerický kód jiný než v předchozích případech.

8. Debug výstup knihovny

Knihovna obsahuje podporu pro logování běhu při nutnosti detekování problémů na straně klienta. Logování výstupu se provádí vytvořením adresáře c:\PortalZP_Log s RW právy. Pokud knihovna nalezne adresář c:\PortalZP_Log, vytvoří do něj soubor PzpCSharpDll.log v UTF-8 kódování, kam bude ukládat logy o průběhu komunikace. Tento logový soubor lze použít jako podklad pro hledání případných problémů.