سرویس‌های سامانه آسان امضا

راهنمای بهره‌برداران API

واحد مرکز میانی صدور گواهی

نسخه	۱.۱۰
تاریخ	اردیبهشت ماه ۱۴۰۳
شناسه	PKI-SS-API-DG
طبقه بندی	عمومی

فهرست مطالب

تاریخچه

نسخه	تاریخ	تهیه کنندگان	مرور کنندگان	توضیحات
۱.۰	۱۴۰۲/۰۷/۰۵	واحد مرکز میانی	واحد کنترل کیفیت	تهیه سند
۱/۱	۱۴۰۲/۰۹/۱۱	واحد مرکز میانی	واحد کنترل کیفیت	اصلاح ساختار Data در JSON ورودی
۱/۲	۱۴۰۲/۱۰/۲۰	توسعه نرم افزار	واحد کنترل کیفیت	تغییر ساختار امضا در درخواست از utf8 به unicode
۱/۳	۱۴۰۲/۱۱/۱۰	توسعه نرم افزار	واحد کنترل کیفیت	اضافه شدن تصویر امضا به خروجی SignRequest
۱/۴	۱۴۰۲/۱۱/۱۵	توسعه نرم افزار	واحد کنترل کیفیت	اضافه شدن کد خطای هر متد در هر بخش
۱/۵	۱۴۰۲/۱۱/۲۴	توسعه نرم افزار	واحد کنترل کیفیت	اضافه شدن متد SignCancel
۱/۶	۱۴۰۳/۰۲/۰۵	توسعه نرم افزار	واحد کنترل کیفیت	اصلاح در متد SignProcess
۱/۷	۱۴۰۳/۰۲/۰۹	توسعه نرم افزار	واحد کنترل کیفیت	ایجاد پیوست شماره ۴ جهت تولید زوج کلید
۱/۸	۱۴۰۳/۰۲/۱۱	توسعه نرم افزار	واحد کنترل کیفیت	بازنگری و ویرایش
۱/۹	۱۴۰۳/۰۲/۱۵	توسعه نرم افزار	واحد کنترل کیفیت	اضافه شدن پیوست ۵
۱/۱۰	۱۴۰۳/۰۲/۱۵	توسعه نرم افزار	واحد کنترل کیفیت	اضافه شدن نمونه کد رمزنگاری به SignProcess

پیوست‌ها

شماره	عنوان	نسخه	توضیحات
۱	جدول کد خطا	۱.۲	در این پیوست کدهای خطا برنامه وجود دارد
۲	فرم اخذ کد مشتری	۱.۰	لیست اطلاعاتی که برای دریافت کد مشتری باید ارسال شود
۳	توالی درخواست گواهی	۱.۰	نحوه و ترتیب اجرای سرویس‌ها جهت دریافت گواهی توسط متقاضی
۴	روش ایجاد گواهی	۱.۰	روش ایجاد گواهی امضای دیجیتال برای امضای درخواستها
۵	نمونه کد	۱.۰	نمونه کد به زبان #C برای امضا محتوا

پیشگفتار

استنادپذیری اسناد و عملیات الکترونیک یکی از اساسی‌ترین پایه‌های خدمات الکترونیکی بخصوصی در حوزه‌هایی که مسائل حقوقی در آن وجود دارد می‌باشد.

طبق قوانین جمهوری اسلامی ایران تنها راه اعطای وجاهت حقوقی به یک سند الکترونیکی امضای دیجیتال آن سند است. به منظور امضای دیجیتال هر شخص باید گواهی امضا دریافت کند. در این سند نحوه استفاده از سرویس‌های امضای دیجیتال در سامانه آسان امضای شرکت پندار کوشک ایمن به منظور امضای دیجیتال اسناد الکترونیک ارائه شده است.

از ویژگی‌های اصلی این سرویس می تواند به موارد ذیل اشاره کرد:

۱- سهولت بسیار بالا در استفاده از سرویس
۲- پایداری بالای سرویس
۳- سرعت بالای سرویس
۴- پشتیبانی از امضای پایه PKCS1 به منظور امکان امضای همه‌ی انواع اسناد با توجه به نیاز کسب و کار
۵- عدم دریافت اصل سند به منظور حفظ محرمانگی اطلاعات
۶- امکان ارائه گزارشات متنوع به متقاضی

تعاریف عمومی

امضا کننده : فردی که باید سند را امضا کند.

متقاضی امضا : سامانه‌ای که می خواهد سندی توسط امضا کننده (کاربر آسان امضا) از طریق ابر آسان امضا، امضا کند.

مشتری : شخص حقوقی طرف قرارداد شرکت پندار کوشک ایمن که از سرویس‌های برای صدور گواهی به متقاضی استفاده می کند.

داده پیام : هر محتوای باینری که باید امضا شود.

رمز گواهی: منظور رمزی است که امضا کننده در زمان دریافت گواهی در سامانه آسان امضا برای گواهی خود گذاشته است.

رمز امضا: یک رمز یکبار مصرف است که برای امضای هر داده پیام به امضاکننده ارسال می‌شود.

امکانات سرویس

در سرویس‌های سامانه آسان امضا قابلیت های زیر وجود دارد:

۱- تشخیص دارا بودن گواهی امضای دیجیتال برای شخص
۲- امکان دریافت گواهی امضا کننده سند
۳- امضای محتوا الکترونیکی با استاندارد PKCS#1
۴- امضای محتوا الکترونیکی بدون اعمال الگوریتم درهمسازی (Hash)
۵- دریافت امضای یک داده پیام که قبلا امضا شده

چگونه از این سرویس می توان استفاده کرد ؟

برای استفاده از سرویس‌های صدور گواهی شرکت پندار کوشک ایمن باید مراحل زیر طی شود:

	اقدام	انجام دهنده
۱	امضای تفاهمنامه همکاری	طرفین
۲	ارائه مستندات سرویس و گواهی های محیط تست	پندار کوشک ایمن
۳	ایجاد کد مشتری و تخصیص دسترسی به سرویس مطابق با اطلاعات پیوست ۲	پندار کوشک ایمن
۴	پیاده سازی سرویس در سامانه مشتری	مشتری
۵	استفاده از سرویس و کسب درآمد از آن	مشتری

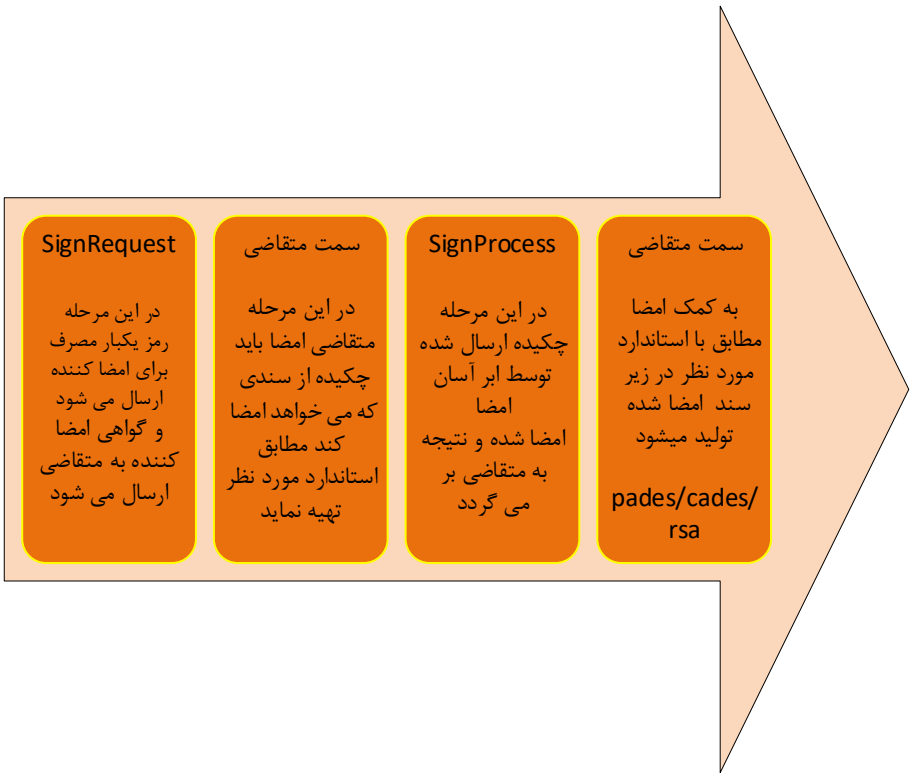
روال امضای داده پیام

برای امضای داده پیام مطابق توالی زیر باید ابتدا درخواست امضا ارسال و سپس داده پیام جهت امضا ارسال گردد:

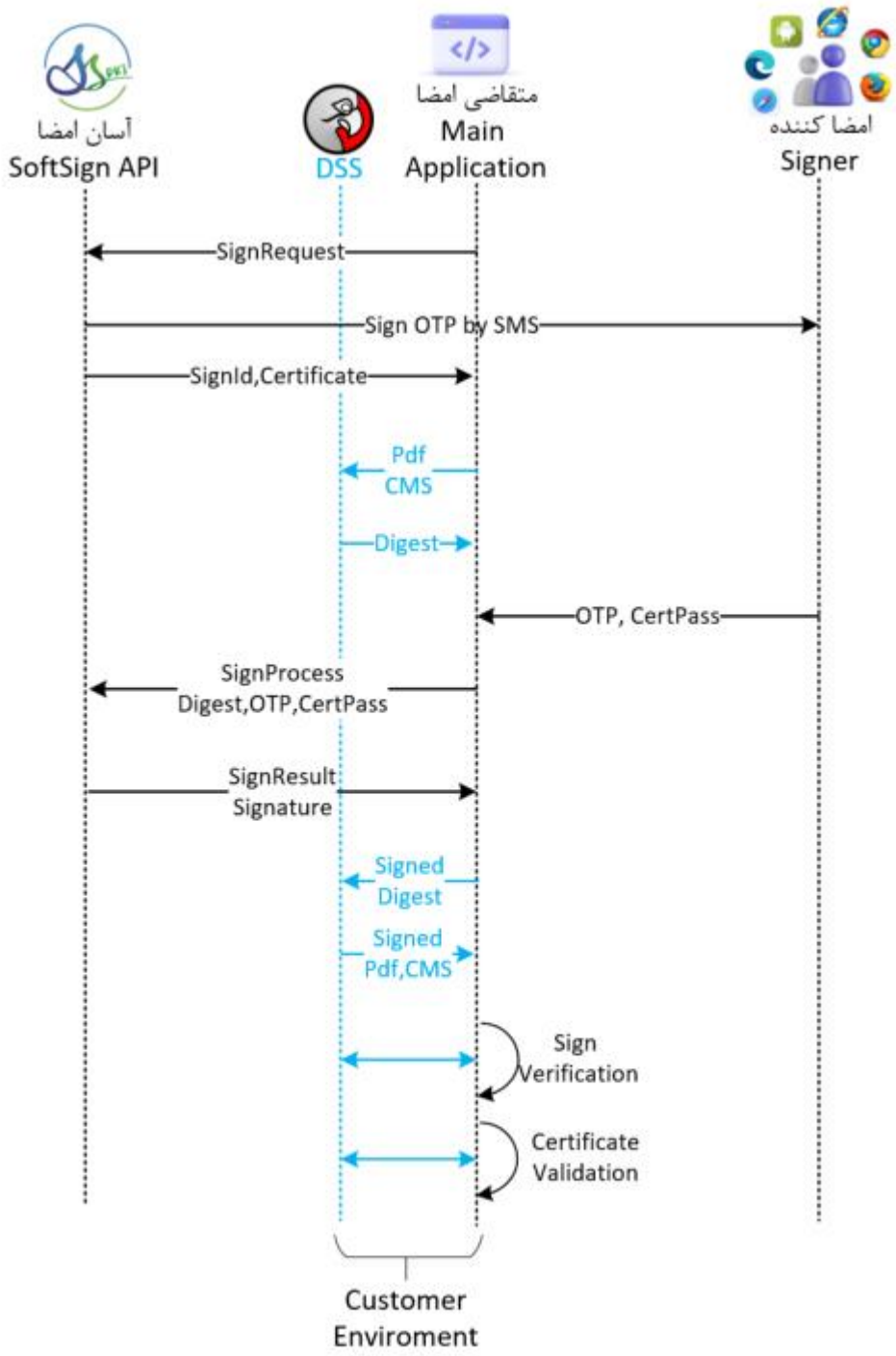
۱- ابتدا متقاضی امضا درخواست امضای خود را که حاوی عنوانی برای درخواستی که باید امضا شود و کد ملی امضا کننده است را صادر می کند. (SignRequest)

۲- یک رمز یکبار مصرف برای امضا کننده توسط سامانه آسان امضا ارسال می‌شود و امضا کننده از درخواست امضا و متقاضی و موضوع آن مطلع می‌شود.

۳- رمز گواهی و رمز امضای توسط متقاضی امضا از امضا کننده دریافت و برای امضا به همراه داده پیام به سامانه آسان امضا ارسال می شود. (SignProcess)



توالی امضا در ابر آسان امضا مطابق نمودار توالی زیر است :



موارد آبی فقط در امضای اسناد نوع PDF یا CMS وجود دارد.
متقاضی می تواند تولید Digest از طریق DSS یا هر سرویس یا ابزاری که صلاح می داند انجام دهد.

در ادامه هر عملیات و روش استفاده از آن بیان شده است.

سرویس‌های درگاه

سند پیش رو مربوط به سرویس‌های امضای دیجیتال سامانه آسان امضا شرکت پندار کوشک ایمن نگارش ۱ می‌باشد. جهت دریافت نگارش سرویس فعال می تواند از آدرس زیر استفاده کرد:

https://api.pki.co.ir/softsign/GetVersion

کلیه متدها از طریق آدرس زیر در دسترس می باشند:

https://api.pki.co.ir/softsign/query

جهت اخذ گواهی امضای دیجیتال و امضای داده پیام عمل‌های زیر در سرویس این شرکت وجود دارد:

- SignRequest
- SignProcess
- GetSignValue
- GetUserCertificate
- SignCancel
- CreateAccount
- CertificateRequest

نکات امنیتی :

۱- دسترسی به سرویس مبتنی بر امضای اختصاصی RSA Sign هر مشتری انجام شده و فاقد Username,Password است.

۲- کلیه متدها بر پایه تراکنش (Transaction Base) بوده و جلسه‌ای (Session) جهت انجام درخواست ایجاد نمی‌شود.

۳- برای هر تراکنش نیاز به تصدیق هویت متقاضی از طریق امضای اختصاصی وی (RSA Sign) در آن تراکنش است.

۴- هر تراکنش انجام شده با امضای متقاضی بعنوان سند انکارناپذیر درخواست انجام آن تراکنش از سوی متقاضی امضا تلقی شده و سندیت حقوقی دارد و کلیه مسولیت آن بعهده متقاضی می‌باشد.

۵- خروجی هر دستور در صورتی که کد خطا صفر باشد معتبر می‌باشد در غیر این صورت باید طبق پیوست ۱ کد خطا را بررسی نمایید.

سرویس امضای دیجیتال

این سرویس دارای شش عمل است ساختار سرویس مطابق زیر است:

جدول ۱ : کد هر عمل

عمل	کد	توضیح
SignRequest	1000	در خواست امضای یک داده پیام
SignProcess	1001	انجام عملیات امضای یک داده پیام
GetUserCertificate	1002	دریافتی گواهی امضای دیجیتال امضا کننده
GetSignValue	1003	دریافت نتیجه امضای یک داده پیام
SignCancel	1004	انصراف از امضای یک محتوا
CreateAccount	1005	ایجاد کاربر جدید در ابر آسان امضا
CertificateRequest	1006	صدور گواهی جدید برای کاربر در ابر آسان امضا

Request	
EndPoint	https://api.pki.co.ir/softsign/query
Method	Post
Body Content-Type	application/json
Encoding	UTF8
Parameters	appId: کد مشتری action: کد عمل مطابق جدول ۱ signature: data امضای بخش Base64(RSASing_PKCS#1_SHA1(UnicodeEncode(data))) data: اطلاعات در قالب json مطابق با هر عمل
Response	
Status Code	200 Success / 401 Unauthorized
	متناسب با هر عمل

۱- برای تولید مقدار پارامتر signature باید مقدار data را با کدینگ Unicode یا UTF16-LE به باینری تبدیل کرده و سپس با کلید خصوصی متقاضی امضا و الگوریتم RSA و استاندارد PKCS#1 و درهمسازی SHA1 آن را امضا کرده و نتیجه را با فرمت Base64 در پارامتر signature قرار داد.

۲- پارامتر data یک json است که در ادامه برای هر عمل محتوای آن مشخص شده است.

۳- الگوریتم درهمسازی برای امضا بخش data باید حتما SHA1 باشد.

SignRequest Action

کد عمل : ۱۰۰۰

این عمل جهت آغاز درخواست امضا توسط متقاضی می‌باشد در این گام متقاضی امضا اعلام می‌کند که می‌خواد یک داده پیام توسط یک فرد امضا شود.

در این عمل ۳ پارامتر به شکل زیر باید ارسال شود:

```
{
    "nationalcode":"کد ملی امضا کننده",
    "subject":"عنوان داده پیام",
    "validtime":مدت زمان مجاز امضا به دقیقه حداکثر می تواند ۱۴۴۰۰ باشد,
    "signimage":"true/false",
    "hashalg":"SHA1/SHA256/SHA384/SHA512"
}
```

- پارامتر signimage اختیاری است و در صورت true بودن عکس امضای فرد در قالب png ارسال می شود.
- پارامتر hashalg اختیاری است اگر این پارامتر نباشد یا مقدار آن SHA256 در نظر گرفته خواهد شد. طبق قوانین مرکز دولتی ریشه پیشنهاد می‌شود این پارامتر SHA256 باشد. براس سهولت پیشنهاد می‌شود این پارامتر را درج نکنید.

مثال برای در خواست امضا :

```
{
"appId":1,
"action":1000,
"signature":"",
"data":{"nationalcode":"0123456789","subject":"تسهیلات امضای قرارداد","validtime":60,"signimage":"false"}
}
```

خروجی :

```
{
"signId":شناسه درخواست امضا,
"certificate":"گواهی امضای امضا کننده در قالب بیس ۶۴",
"signimage":"",
"errorCode":کد خطا, (0,6901,6923,6918,6922,6933)
"errorMessage":"متن خطا"
}
```

از گواهینامه[۱] که در خروجی ارائه می شود می‌توان برای تولید چکیده[۲] در قالب امضای Pades(pdf) یا Cades(CMS) استفاده کرد. همچنین اگر به تصویر امضا برای درج در سند نیاز دارید می توانید از مقدار signimage استفاده کنید. دقت داشته باشید عکس امضا سندیت نداشته و صرفا بابت نمایش است و ممکن است با عکس امضای واقعی فرد انطباق نداشته باشد.

[۱] Sign Certificate

[۲] digest

SignProcess Action

کد عمل : ۱۰۰۱

این عمل جهت انجام عملیات امضای دیجیتال توسط امضا کننده می‌باشد در این گام داده پیام توسط امضا کننده امضا شده و نتیجه به متقاضی امضا ارسال می شود.

در این عمل ۴ پارامتر به شکل زیر باید ارسال شود:

```
{
    " signId":شناسه امضا,
    " dataforsign":"داده پیام در قالب بیس ۶۴",
    " password":"رمز گواهی امضا کننده", پیشنهاد می شود این رمز بصورت کد شده (مطابق توجه ۱ در ادامه) ارسال شود
    "otp":"رمز امضا همان رمزی که برای امضا کننده از طریق پیامک ارسال شده است",
    "pkcs1support":" true/false"
}
```

- پارامتر pkcs1support اختیاری است اگر این پارامتر نباشد یا مقدار آن true باشد امضا دقیقا مطابق استاندارد PKCS#1 انجام می‌شود و اگر مقدار آن false باشد سیستم بدون تولید جکیده[۳] SHA256 محتوا را امضا می‌کند. بدیهی است در این حالت داده ورودی حتما باید با الگوریتم SHA256 جکیده شده باشد در غیر این صورت امضای حاصل با سیستم‌های عمومی قابل اعتبارسنجی نخواهد بود. توجه در امضای pdf یا cms یا PKCS#7 این پارامتر یا نباید باشد و یا مقدار آن حتما true باشد.

نمونه ورودی در عملیات امضا :

```
{
"appId":1,
"action":1001,
"signature":"",
"data":{"signId":0,"dataforsign":"cABrAGkAYwBvAA==","password":"?","otp":"000000"}
}
```

خروجی :

```
{
"signature":"امضا در قالب بیس ۶۴",
"errorCode":کد خطا, (0,6920,6926,6927,6925,6913,6922)
"errorMessage":"متن خطا"
}
```

[۳] Digest or Hash

توجه :

۱- بهتر است به منظور حفظ امنیت رمز گواهی، مقدار رمز گواهی را با گواهی ابر آسان امضا با الگوریتم RSA و پدینگ PKCS1 رمز کرده و مقدار رمز شده ی رمز گواهی را در پارامتر password قرار دهید.

گواهی رمزنگاری ابر آسان امضا در آدرس زیر وجود دارد:

https://pki.co.ir/download/SS/SoftSign.cer

نمونه کد رمزنگاری پسورد به زبان #C

```
X509Certificate2 ssCert = new X509Certificate2(گواهی رمزنگاری ابر آسان امضا);
String encryptedPass =
Convert.ToBase64String(ssCert.GetRSAPublicKey().Encrypt(Encoding.Unicode.GetBytes(رمزگواهی), RSAEncryptionPadding.Pkcs1));
```

۲- مقدار dataforsign می‌تواند بصورت آرایه باشد. در این حالت کلیه داده‌های ارسالی امضا خواهد شد و پاسخ نیز بصورت یک آرایه دقیقا به ترتیب ورودی در خروجی قرار خواهد گرفت. این حالت برای امضا همزمان چند سند بسیار کاربردی می‌باشد.

نمونه ورودی در عملیات امضا انبوه :

```
3- {
4- "appId":1,
5- "action":1001,
6- "signature":"",
7- "data":{"signId":0,"dataforsign":["cABrAGkA","cABrAGkAYwBvAA=="],"password":"?","otp":"?"}
8- }
```

خروجی :

```
9- {
10- "signature":["امضا در قالب بیس ۶۴","امضا در قالب بیس ۶۴"],
11- "errorCode":کد خطا, (0,6920,6926,6927,6925,6913,6922)
12- "errorMessage":"متن خطا"
13- }
```

GetUserCertificate Action

کد عمل : ۱۰۰۲

این عمل جهت دریافت گواهی الکترونیکی یک امضا کننده است. در صورتی که امضا کننده در سامانه آسان امضا داری گواهینامه فعال و معتبر برای امضا باشد با کمک این عمل می توان گواهی فرد را دریافت کرد. از این گواهی برای تولید چکیده[۴] در قالب امضای Pades(pdf) یا Cades(CMS) می توان استفاده کرد.

در این عمل یک پارامتر به شکل زیر باید ارسال شود:

```
{
        "nationalcode":"کد ملی امضا کننده"
}
```

نمونه ورودی دریافت گواهی امضا :

```
{
"appId":1,
"action":1002,
"signature":"",
"data":{"nationalcode":"1234567890"}
}
```

خروجی :

```
{
"certificate":"گواهی امضای امضا کننده در قالب بیس ۶۴",
"errorCode":کد خطا, (0,6914)
"errorMessage":"متن خطا"
}
```

در صورتی که فرد فاقد گواهی فعال باشد خروجی این عمل خالی خواهد بود

[۴] digest

GetSignValue Action

کد عمل : ۱۰۰۳

چنانچه در عمل امضا به هر دلیلی نتیجه به متقاضی نرسد و امضا توسط امضاکننده انجام شده باشد با کمک این عمل می‌توان نتیجه امضا را مجدد دریافت کرد.

در این عمل ۲ پارامتر به شکل زیر باید ارسال شود:

```
{
    " signId":شناسه امضا,
    "nationalcode":"کد ملی امضا کننده"
}
```

نمونه ورودی دریافت امضای یک داده پیام :

```
{
"appId":1,
"action":1003,
"signature":"",
"data":{"signId":3,"nationalcode":"1234567890"}
}
```

خروجی :

```
{
"signature":"امضا در قالب بیس ۶۴",
"errorCode":کد خطا, (0,6914)
"errorMessage":"متن خطا"
}
```

SignCancel Action

کد عمل : ۱۰۰۴

چنانچه به هر دلیلی نتیجه به متقاضی امضا از انجام امضا منصرف شود با کمک این دستور می‌تواند درخواست خود را کنسل کند. بدیهی است درخواست امضا شده یا منقضی شده کنسل نخواهد شد.

در این عمل ۲ پارامتر به شکل زیر باید ارسال شود:

```
{
    " signId":شناسه امضا,
    "nationalcode":"کد ملی امضا کننده"
}
```

نمونه ورودی دریافت امضای یک داده پیام :

```
{
"appId":1,
"action":1003,
"signature":"",
"data":{"signId":3,"nationalcode":"1234567890"}
}
```

خروجی :

```
{
"errorCode":کد خطا, (0,6909,6920)
"errorMessage":"متن خطا"
}
```

CreateAccount Action

کد عمل : ۱۰۰۵

این عمل برای ایجاد کاربر جدید در ابر آسان امضا توسط متقاضی امضا می‌باشد. فقط متقاضی می‌تواند در ابر آسان امضا کاربر ایجاد کند که دسترسی و اعتبار مالی لازم را داشته باشد.

هدف از این دستور ایجاد یک کاربر جدید در ابر آسان امضا توسط متقاضی امضا بدون خروج کاربر از سایت متقاضی امضا می‌باشد تا به این ترتیب کاربر تجربه بهتری در استفاده از سایت متقاضی امضا داشته باشد.

برای درک بهتر پیوست ۳ را ببینید.

در این عمل ۲ پارامتر به شکل زیر باید ارسال شود:

```
{
    "nationalcode":"کد ملی امضا کننده",
    "mobile":"شماره موبایل"
}
```

نمونه ورودی دریافت امضای یک داده پیام :

```
{
"appId":1,
"action":1005,
"signature":"",
"data":{"nationalcode":"1234567890","mobile":"09121234567"}
}
```

خروجی :

```
{
"errorCode":کد خطا, (0,6909,6930,6919,6915,6918)
"errorMessage":"متن خطا"
}
```

چنانچه کاربر در گذشته توسط متقاضی امضا در ابر آسان امضا عضو شده باشد با اجرای این دستور در ابر آسان امضا لاگین خواهد شد.

CertificateRequest Action

کد عمل : ۱۰۰۶

این عمل برای صدور گواهی الکترونیکی برای کاربر یک متقاضی می‌باشد. یک متقاضی فقط برای کاربری که خود ایجاد کرده می‌تواند گواهی صادر کند. فقط متقاضی می‌تواند در ابر آسان امضا برای یک کاربر گواهی صادر کند که دسترسی و اعتبار مالی لازم را داشته باشد.

هدف از این دستور ایجاد صدور یک گواهی در ابر آسان امضا توسط متقاضی امضا برای امضا کننده فاقد گواهی فعال بدون خروج کاربر از سایت متقاضی امضا می‌باشد تا به این ترتیب کاربر تجربه بهتری در استفاده از سایت متقاضی امضا داشته باشد.

برای درک بهتر پیوست ۳ را ببینید.

در این عمل ۹ پارامتر به شکل زیر باید ارسال شود:

کلیه اطلاعات توسط متقاضی امضا (در سایت یا برنامه ایشان) دریافت می‌شود و برای ابر آسان امضا ارسال می‌گردد.

```
{
    "nationalcode":"کد ملی امضا کننده",
    "mobile":"شماره موبایل",
    "cardserialno":"شماره سریال کارت ملی امضا کننده ",
    "certpass":"رمز گواهی الکترونیکی",
    "birthdate":"yyyy/mm/dd", تاریخ تولد امضا کننده به میلادی
    "birthdateshamsi":"yyyy/mm/dd", تاریخ تولد امضا کننده به شمسی
    "postalcode":"کد پستی (اختیاری)",
    "callbackurl":"آدرس برگشت بعد از احراز هویت",
    "otp":"رمز یکبار مصرف"
}
```

- certpass رمز گواهی الکترونیکی : یک رمز است که باید توسط کاربر تعیین شود این رمز باید حداقل ۸ حرف باشد و پیچیدگی لازم را داشته باشد کنترل طول رمز و پیچیدگی آن بعهده متقاضی امضا است.
- otp رمز یکبار مصرف : رمزی است که در مرحله createaccount توسط ابر آسان امضا برای امضا کننده ارسال شد.
- callbackurl آدرس برگشت : یک آدرس از سایت متقاضی امضا که پس از احراز هویت کاربر نتیجه به آن آدرس ارسال می‌شود. در این آدرس دو متغییر @errorcode و @errormessage جایگزین می‌شود. نمونه آدرس در زیر آورده شده است:

https://yourdomain?errorcode=@errorcode&errormessage=@errormessage

نمونه ورودی دریافت امضای یک داده پیام :

```
{
"appId":1,
"action":1006,
```

"signature":"",
"data":{"nationalcode":"1234567890","mobile":"09121234567",...}
}

خروجی :

{
"RequestId":شناسه درخواست,
"FirstName":"نام کاربر",
"LastName":"نام خانوادگی کاربر",
"TrackingCode":"کد رهگیری درخواست",
"eKYCURL":"آدرس احراز هویت",
"errorCode":کد خطا, (0,1,6918,6909,6923)
"errorMessage":"متن خطا"
}

بعد از اجرای موفق این دستور سایت متقاضی باید کاربر را به آدرس مندرج در eKYCURL منتقل نماید تا کاربر احراز هویت شود. چنانچه عملیات احراز هویت به هر شکلی پایان یابد (موفق یا ناموفق) نتیجه آن به سایت متقاضی از طریق آدرس callbackurl اعلان خواهد شد.

پیوست شماره ۱

کد خطا	شرح خطا
0	اجرای موفق
1	خطا در اجرای برنامه
6900	خطای داخلی
6901	کد مشتری نا معتبر است
6902	لایسنس استفاده شده است
6903	Action is not valid
6904	License count is not valid
6905	مبدا درخواست معتبر نیست
6906	محصول وجود ندارد
6907	برنامه معتبر نیست
6908	این شناسه پرداخت استفاده شده است
6909	محدودیت دسترسی
6910	کلید عمومی تعریف نشده است
6911	توکن پیدا نشد
6912	امضا نا معتبر است
6913	رمز اشتباه است
6914	رکورد پیدا نشد
6915	شماره موبایل با کد ملی انطباق ندارد
6916	زمان استفاده از رمز سپری شد
6917	کالرکد نامعتبر است
6918	کاربر وجود ندارد
6919	سرویس شاهکار قطع است
6920	درخواست گواهی پیدا نشد
6921	این فرد گواهی فعال ندارد
6922	گواهی فعال وجود ندارد
6923	موجودی کافی نیست
6924	این درخواست قبلا احراز هویت شده است
6925	امکان امضای این درخواست وجود ندارد

6926	این درخواست قبلا امضا شده است
6927	زمان امضای درخواست سپری شده است
6928	کد درخواست با کد ملی انطباق ندارد
6929	کد تخفیف نامعتبر است
6930	کاربر تکراری است
6931	این فرد گواهی فعال دارد
6932	این درخواست احراز هویت نشده است
6933	الگوریتم هشینگ نامعتبر است

پیوست شماره ۲

اطلاعات مورد نیاز جهت تعریف کد مشتری به شرح زیر می باشد. این اطلاعات به همراه کلید عمومی مشتری باید به آدرس ایمیل sales@pki.co.ir با عنوان "درخواست ایجاد کد مشتری در سرویس صدور گواهی" ارسال شود.

۱- نام شرکت
۲- شناسه ملی شرکت
۳- کدپستی و آدرس شرکت
۴- آدرس سایت شرکت
۵- آدرس ایمیل شرکت
۶- تلفن شرکت
۷- نام و نام خانوادگی مدیرعامل
۸- شماره ملی مدیر عامل
۹- تلفن همراه مدیرعامل
۱۰- گواهی الکترونیک با طول کلید ۱۰۲۴ ترجیحا الگوریتم RSA اختصاصی شرکت

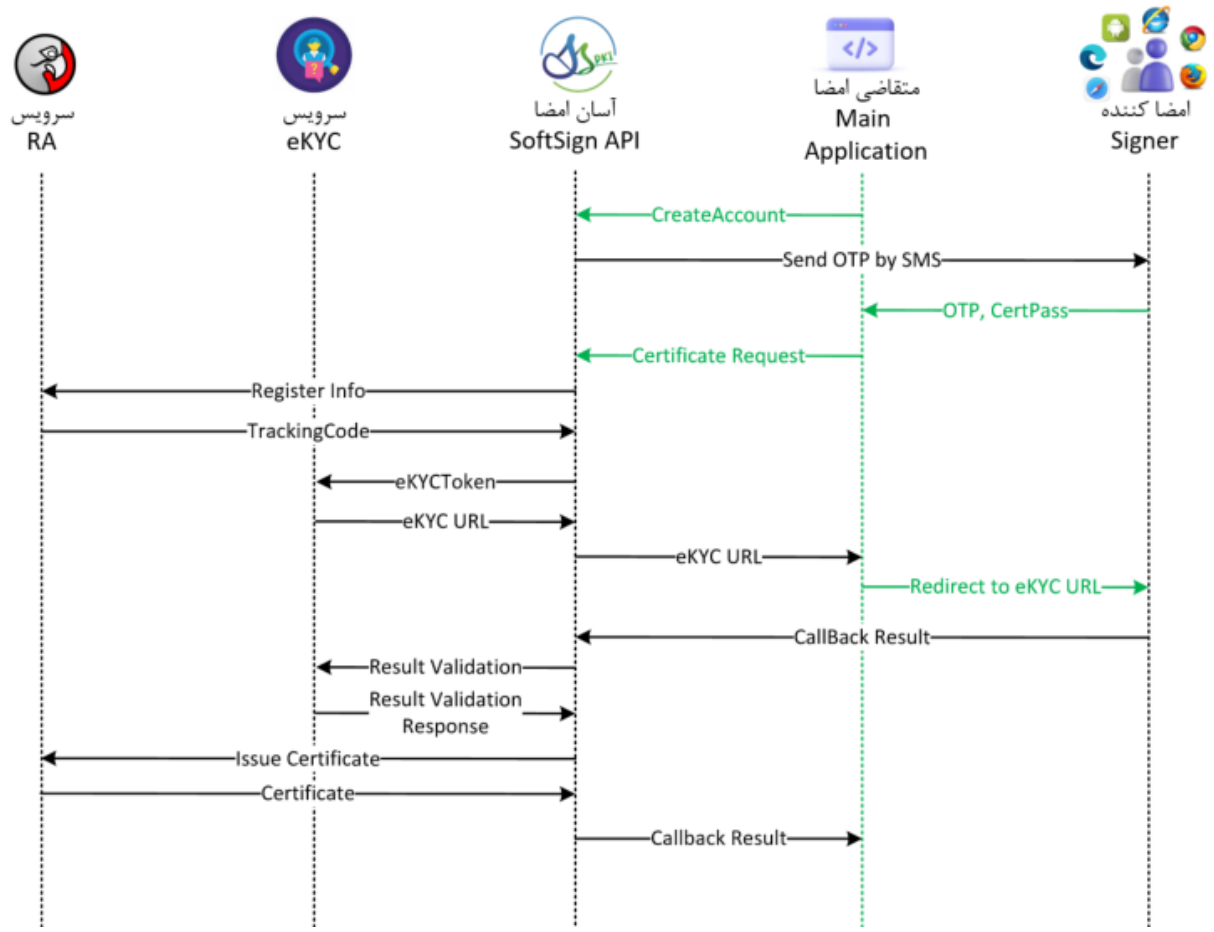
پیوست شماره ۳

توالی درخواست صدور گواهی توسط متقاضی (مشتری آسان امضا)

لازم به ذکر است :

۱- این فرآیند فقط باید توسط مشتریی انجام شود که نیاز دارد از طریق سایت خود در ابر آسان امضا گواهی صادر نماید. پیشنهاد می‌شود برای حفظ امنیت خود و کاربر، کاربر را برای صدور گواهی به سایت ابر آسان امضا (softsign.ir) منتقل نمایید.

۲- در این توالی فقط موارد سبز در برنامه متقاضی انجام می شود سایر موارد مربوط به زیر سیستم ها است و از دید متقاضی مخفی می باشد.



همانطور که در مستند بیان شده برای صدور گواهی به اطلاعات زیر از کاربر نیاز است. پیشنهاد می‌شود در سایت متقاضی ابتدا اطلاعات هویتی (موارد ۱ تا ۵) دریافت و سپس مرحله اول توالی (CreateAccount) انجام شده و سپس رمزهای مورد نیاز (موارد ۶ و ۷) از کاربر دریافت و سرویس دوم توالی (CertificateRequest) فراخوانی شود.

۱-کد ملی
۲-شماره موبایل
۳- تاریخ تولد
۴- شماره سریال کارت ملی
۵-کد پستی
۶- رمز گواهی
۷- رمز یکبار مصرف

پیوست شماره ۴

چگونگی تولید زوج کلید در تصدیق هویت متقاضی

همانگونکه در بخش سرویس های درگاه امضا بیان گردید, برای هر تراکنش نیاز به تصدیق هویت متقاضی از طریق امضای اختصاصی وی (RSA Sign) در آن تراکنش می باشد. جهت راهنمایی بهتر متقاضیان سعی شده که در این پیوست با استفاده از OpenSSL اقدام به ایجاد زوج کلید تصدیق هویت شود.

در ابتدا لازمست تا نرم افزار openssl نسخه ویندوز خود را دانلود کنید:

https://slproweb.com/products/Win32OpenSSL.html

در مسیر برنامه, دستور زیر را اجرا کنید. توجه کنید که بعد از O نام شرکت و بعد از CN نام خود و یا نام پروژه را قرار دهید:

```
openssl req -x509 -newkey rsa:1024 -keyout key.pem -out cert.pem -sha256 -days 3650 -nodes -subj "/C=IR/O=Company_Name/CN=Developer_or_Project_Name/OU=Department"
```

در صورت اجرای موفق دستور فوق, یک فایل بنام cert.pem حاوی کلید عمومی ایجاد می شود که باید آنرا برای شرکت ارسال کنید.

با دستور زیر و به منظور نگهداری امن کلیدها, می توانید فایل ها را به قالب pfx تبدیل کنید:

```
openssl pkcs12 -inkey key.pem -in cert.pem -export -out Cert.pfx
```

با این اقدام, می توان دو فایل pem را حذف نمود.

پیوست شماره ۵

نمونه کد برای امضای محتوا به زبان #C

```
private string Sign(string body, X509Certificate2 cert)
{
    byte[] data4Sign = Encoding.Unicode.GetBytes(body)
    if (cert.HasPrivateKey)
    {
        RSA rsa = cert.GetRSAPrivateKey();
        byte[] signed= rsa.SignData(data4Sign, HashAlgorithmName.SHA1, RSASignaturePadding.Pkcs1);
        return Convert.ToBase64String(signed);
    }
    return “”;
}
```