سرویس‌های درگاه اینترنتی امضا

راهنمای بهره‌برداران

واحد امنیت اطلاعات و زیرساخت کلید عمومی

نسخه	۲.۵.۴
تاریخ	فروردین ماه ۱۴۰۵
شناسه	PKI-ISG-API-DG
طبقه بندی	عمومی

تاریخچه

نسخه	تاریخ	تهیه کنندگان	مرور کنندگان	توضیحات
۲.۰.۰	۱۴۰۰/۱۱/۰۲	واحد امنیت	واحد کنترل کیفیت	تهیه سند
۲.۰.۱	۱۴۰۰/۱۱/۰۵	واحد امنیت	واحد کنترل کیفیت	تکمیل توضیحات سند
۲.۰.۲	۱۴۰۰/۱۱/۰۷	واحد امنیت	واحد کنترل کیفیت	تکمیل توضیحات API های: GetSigningToken ResendNotiication
۲.۲.۰	۱۴۰۰/۱۱/۲۵	واحد امنیت	واحد کنترل کیفیت	اضافه شدن ContentImage و HandwriteSignature
۲.۲.۱	۱۴۰۱/۰۱/۳۰	واحد امنیت	واحد کنترل کیفیت	تکمیل توضیحات ContentImage و HandwriteSignature
۲.۲.۲	۱۴۰۱/۰۳/۰۴	واحد امنیت	واحد کنترل کیفیت	نکته قابل توجه در چک کردن base64string
۲.۳.۰	۱۴۰۱/۰۴/۲۹	واحد امنیت	واحد کنترل کیفیت	به‌روزرسانی سند و امضای نوع CMS Attached
۲.۳.۱	۱۴۰۱/۰۵/۰۴	واحد امنیت	واحد کنترل کیفیت	تکمیل توضیحات ContentImage
۲.۴.۰	۱۴۰۱/۰۷/۳۰	واحد امنیت	واحد کنترل کیفیت	بهبود صف امضا
۲.۴.۲	۱۴۰۱/۰۸/۰۷	واحد امنیت	واحد کنترل کیفیت	تبدیل نسخه PDF
۲.۴.۳	۱۴۰۱/۰۸/۳۰	واحد امنیت	واحد کنترل کیفیت	محدودیت تاریخ انقضای محتوا به ۱۰ روز
۲.۴.۴	۱۴۰۱/۰۹/۰۵	واحد امنیت	واحد کنترل کیفیت	– افزودن قابلیت رمزگذاری و رمزگشایی محتوای امضا – کنترل حجم محتوای امضا با توجه به حجم تنظیم شده در Config – محدودیت تاریخ انقضای امضا با توجه زمان تنظیم شده در Config
۲.۴.۵	۱۴۰۱/۰۹/۲۹	واحد امنیت	واحد کنترل کیفیت	یکسان کردن Conversionهای محتوای امضا افزودن Certificate Validation به نمودار امضا
۲.۴.۶	۱۴۰۱/۱۰/۰۹	واحد امنیت	واحد کنترل کیفیت	توضیحات بیشتر در مورد برخی فیلدهای GetSigningToken
۲.۴.۷	۱۴۰۲/۰۶/۰۸	واحد امنیت	واحد کنترل کیفیت	افزودن قابلیت عدم اعتبارسنجی گواهینامه صادرشده از مراکز میانی که قابلیت اعتبارسنجی گواهی را ندارند.
۲.۴.۸	۱۴۰۲/۰۷/۰۱	واحد امنیت	واحد کنترل کیفیت	افزودن تایپ چکاد
۲.۴.۹	۱۴۰۲/۱۰/۰۵	واحد امنیت	واحد کنترل کیفیت	تغییر خروجی ارسال نوتیفیکیشن و افزودن توابع ذیل: GetSigningList RemoveFromQueue
۲.۴.۱۰	۱۴۰۳/۰۲/۲۲	واحد امنیت	واحد کنترل کیفیت	افزودن ErrorCode ها به جدول مربوطه، افزودن امکان امضا با SoftSign و امضای چکاد
۲.۵.۱	۱۴۰۳/۰۲/۲۲	واحد امنیت	واحد کنترل کیفیت	افزودن توضیحات مربوط به ساختار Html در امضای نوع محتوایJson
۲.۵.۲	۱۴۰۳/۱۰/۰۹	واحد امنیت	واحد کنترل کیفیت	افزودن امضای اشخاص حقوقی
۲.۵.۳	۱۴۰۳/۱۱/۲۷	واحد امنیت	واحد کنترل کیفیت	اصلاح خروجی سرویس نوتیفیکیشن برای حفظ Backward compatibility
۲.۵.۴	۱۴۰۵/۰۲/۰۷	واحد امنیت	واحد کنترل کیفیت	- تغییر نام سرویس امضای اشخاص حقیقی به سرویس ارسال درخواست امضا - تغییر نام سرویس امضای اشخاص حقوقی به سرویس ارسال درخواست برای دارندگان حق امضا - بروزرسانی خروجی CheckStatus در سرویس امضای چندگانه

پیوست‌ها

ردیف	نام	نسخه	توضیحات
۱	ISG.DTO.nupkg	1.6.14	Binary and source code
۲	ISG.DTO.jar	1.6.14	Binary and source code
۳	ISG.Demo	2.5.5	Sample Application with source code

فهرست مطالب

پیشگفتار

با توجه به شتاب توسعه فناوری اطلاعات در حوزه نرم افزارهای کاربردی و همچنین نیاز به تولید محصولات نرم افزاری در پلتفرم‌های متنوع، دیگر تولید SDK به تمام زبان‌های برنامه‌نویسی امری منطقی محسوب نمی‌گردد. از سوی دیگر ارائه نرم‌افزار به عنوان یک سرویس (SaaS) راهکاری است که در سالیان اخیر پاسخگوی بسیاری از نیازمندی‌های موسسات و سازمان‌های متوسط تا بزرگ در سطح جهان بوده است.

درگاه اینترنتی امضا ابزاری برای تعامل با حوزه زیرساخت کلید عمومی در تمامی پلتفرم‌ها است که می‌تواند نیازمندی‌های بهره‌برداران و دغدغه‌های برنامه‌نویسان را در این حوزه پوشش دهد.

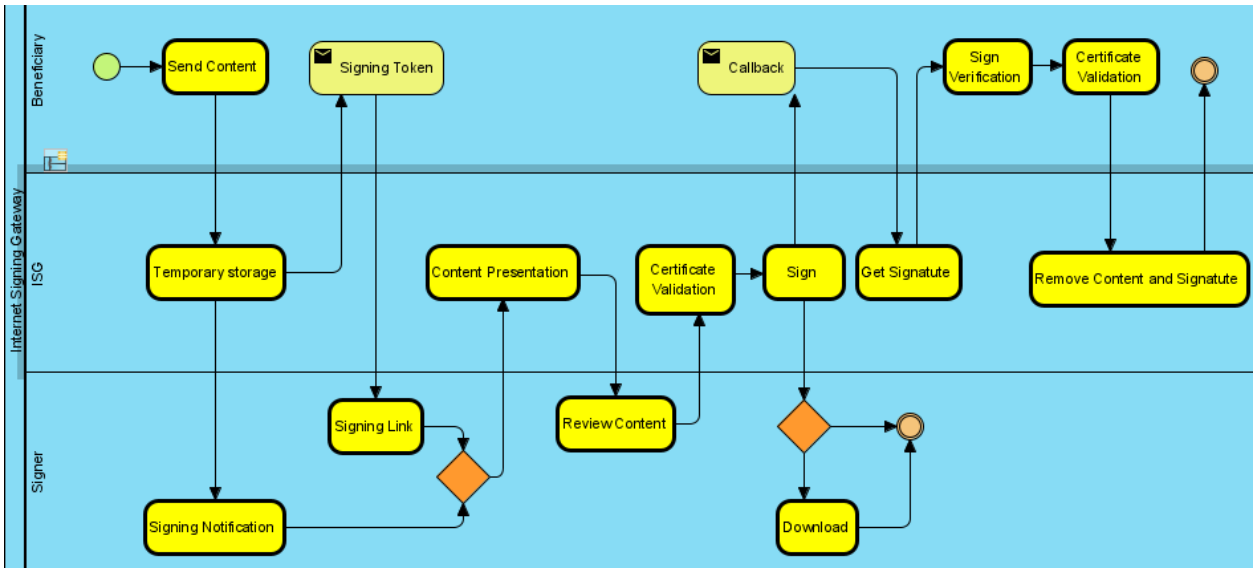
برای به دست آوردن دیدی روشن از درگاه امضا، می‌توان خریدهای اینترنتی را در نظر گرفت. بطور معمول هر گاه از یک فروشگاه آنلاین خریدی انجام شود، عملیات پرداخت وجه از درگاه پرداخت بانکی صورت می‌گیرد. در حقیقت پذیرنده یا همان فروشگاه آنلاین، به هیچ عنوان درگیر عملیات انتقال وجه نبوده و تنها پس از پایان تعاملات لازم بین مشتری و بانک، رسید پرداخت را از بانک دریافت می‌نماید. درگاه اینترنتی امضا نیز مشابه همین رویکرد را دارد. پس از آماده شدن محتوایی که باید امضا شود، فرد امضاکننده به درگاه هدایت می‌شود. در آنجا مطابق قوانین و استانداردهای لازم، ضمن رویت محتوا، آنرا خوانده، بررسی نموده و در صورت موافقت، امضا خواهد نمود. آنگاه نتیجه امضا برای تهیه‌کننده محتوا ارسال خواهد شد.

در این سند ابتدا فرآیند امضای آنلاین بررسی شده، سپس سرویس‌های درگاه که تعاملات لازم در فرآیند را ارائه می‌دهند، توصیف خواهند شد.

در درگاه امضا، به دو صورت می‌توان فرآیند امضا را انجام داد: مدل Plain و مدل امضای رمزگذاری شده.

مدل Plain به صورت ذیل می‌باشد:

فرآیند امضای Plain آنلاین



همانگونه که در نمودار فوق مشخص است، کلیات فرآیند Plain امضا چنین است:

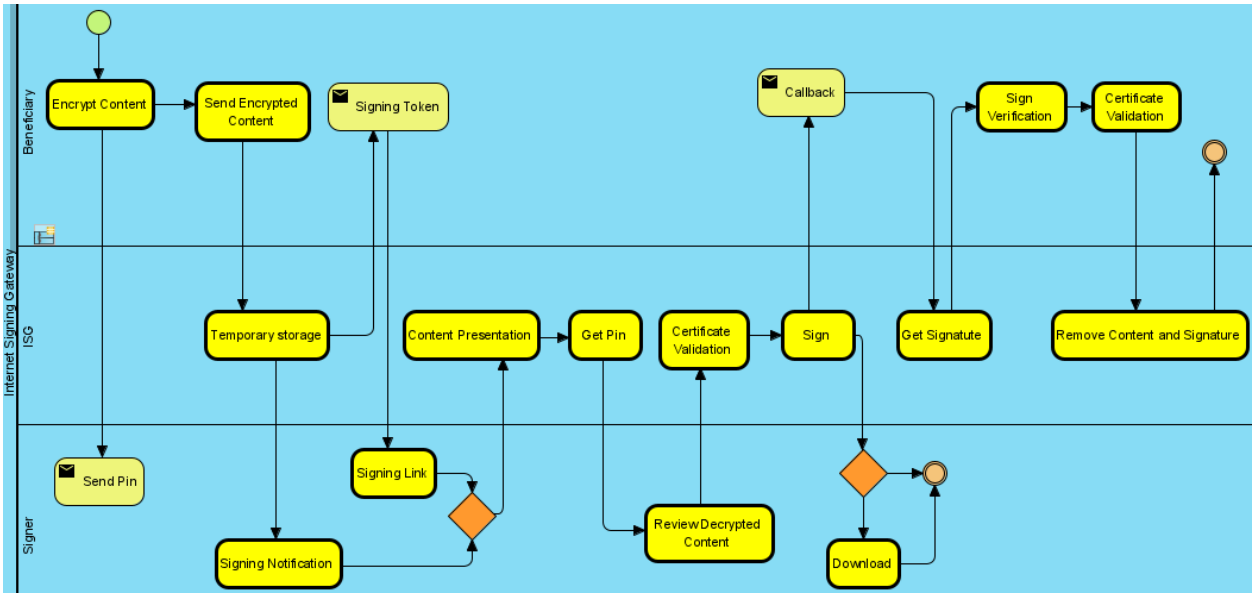
۱. بهره‌بردار محتوای مورد نظر را برای درگاه امضا ارسال می‌نماید.
۲. درگاه محتوای دریافتی را به صورت موقت ذخیره می‌کند.
۳. درگاه دریافت محتوا را اطلاع می‌دهد:
   ۳/۱. در پاسخ به ارسال محتوا، یک شناسه ارجاع به محتوا به بهره‌بردار ارایه می‌دهد.
   ۳/۲. در صورت درخواست بهره‌بردار یک پیام اطلاع‌رسانی به گوشی همراه امضاکننده ارسال می‌نماید.
۴. متناسب با درخواست بهره‌بردار، امضاکننده یا توسط خود بهره‌بردار یا توسط پیام دریافتی از درگاه، به درگاه امضا هدایت می‌شود.
۵. درگاه محتوا را به امضاکننده ارایه می‌نماید.
۶. امضاکننده ضمن رویت و بررسی محتوا، با استفاده از امضای دیجیتال خود اقدام به امضای آن می‌نماید.
۷. درگاه، پس از اعتبارسنجی گواهی کاربر و تکمیل فرآیند امضا، امضا شدن محتوا را به بهره‌بردار اطلاع می‌دهد.
   ۷/۱. در صورتی که بهره‌بردار هنگام ارسال محتوا، اجازه دانلود توسط امضاکننده را داده باشد، درگاه امکان دانلود را در اختیار امضاکننده قرار می‌دهد.
   ۷/۲. تعامل امضاکننده با یا بدون دانلود محتوای امضا شده به پایان می‌رسد.
۸. بهره‌بردار امضا را از درگاه دریافت می‌نماید.
۹. بهره‌بردار، امضا دریافتی را اعتبارسنجی می‌نماید.

۱۰. اگر مرکز میانی گواهی، امکان اعتبارسنجی گواهی را داشته باشد، بهره‌بردار، باید گواهی کاربر را با استفاده از سرویس DSS یا SDK PKTB، اعتبارسنجی نماید.(در نمونه برنامه پیوست این سند به نام ISG.Demo، فراخوانی‌های مربوط به اعتبارسنجی گواهی با استفاده از DSS و SDK PKTB پیاده‌سازی شده و قابل بررسی برای توسعه‌دهندگان می‌باشد.)

۱۱. بهره‌بردار، به درگاه اعلام می‌نماید که محتوا و امضای آن را حذف نماید.

۱۲. درگاه نتیجه حذف اطلاعات را به بهره‌بردار اعلام و فرآیند به پایان می‌رسد.

فرآیند امضای رمزگذاری شده آنلاین



همانگونه که در نمودار فوق مشخص است، فرآیند امضای رمزگذاری شده مانند امضای Plain است با این تفاوت که در مرحله اول، بهره‌بردار باید محتوای موردنظر را به صورت رمزگذاری شده برای درگاه امضا ارسال نماید.

سایر مراحل تا مرحله ۵ نیز مانند فرآیند Plain انجام می‌شود.

در مرحله ۵، برای نمایش محتوای امضا، ابتدا پنجره‌ای به امضاکننده نمایش داده می‌شود تا رمزی که برای ایشان از طرف بهره‌بردار ارسال شده است را وارد نموده و در صورت ورود رمز صحیح، محتوای امضا نمایش داده می‌شود. در غیر این‌صورت محتوای امضا قابل نمایش نخواهد بود.

و در صورت ورود صحیح رمز و پس از نمایش محتوا، سایر مراحل نیز مانند فرآیند امضای Plain انجام می‌گردد.

شایان ذکر است که Encrypt محتوای امضا به صورت ذیل انجام می‌گردد:

بهره‌بردار باید برای هر عملیات امضا، یک رمز ایجاد کرده و این رمز را به همراه محتوایی که باید برای امضا ارسال شود، به تابع Encrypt که طی یک الگوریتم توافقی، عملیات رمزگذاری محتوا را انجام و نتیجه را به صورت یک رشته‌ی Base64 برمیگرداند،

ارسال نماید. پس از انجام عملیات Encrypt امضا، بهره‌بردار باید محتوای Encrypt شده را به درگاه امضا و رمز را نیز برای امضاکننده ارسال نماید و امضاکننده از این رمز، زمان نمایش محتوا برای رمزگشایی محتوا استفاده می‌نماید.

فرآیند امضای چندگانه در درگاه امضای ISG

در این فرآیند، باید اطلاعات صاحبین امضا در سازمان ثبت اسناد، ثبت شده باشد.

در این مدل امضا، بهره‌بردار زمان ارسال محتوای امضا در تابع Legal/GetSigningToken باید علاوه بر پارامترهای دیگر که مشابه پارامترهای درخواست امضای شخص حقیقی می‌باشد، شناسه شخص حقوقی(سازمان/شرکت) را نیز به همراه نوع ترتیب‌دهی، مشخص نماید.

نوع ترتیب‌دهی، می‌تواند به دو صورت ذیل تنظیم شود:

- Manual
- Auto

اگر نوع ترتیب‌دهی به صورت Auto تعیین شود، ابتدا ترتیب امضاکنندگان در فهرست صاحبین امضا به صورت سیستمی تنظیم می‌شود و سپس درخواست امضا برای اولین امضاکننده در فهرست، ارسال می‌گردد و پس از امضای موفقیت آمیز هر امضاکننده، درخواست امضا به ترتیب برای نفر بعدی ارسال می‌شود.

اگر نوع ترتیب‌دهی، به صورت Manual تعیین شده باشد، ابتدا بهره‌بردار باید فهرست امضاکنندگان را از سرویس Legal/GetSignerList دریافت کند و سپس برای هر یک از امضاکنندگان، با استفاده از سرویس SetSignerOrder، ترتیب امضا را مشخص کند. پس از تعیین ترتیب، درخواست امضا به ترتیب مشخص‌شده برای صاحبین امضا ارسال می‌گردد.

سرویس‌های درگاه

جهت آغاز فرآیند امضای آنلاین و ادامه تعامل با درگاه سرویس‌هایی به شرح زیر ارایه شده‌اند:

Auth
- Login
- Logout

Token
- Refresh
- Revoke

Signing
- ChangePassword
- AllTemplates
- TemplateInfo
- TemplateStructure
- GetSigningToken
- CheckStatus
- GetData
- RemoveData
- ResendNotification

Notification
- GetSigningQueueList
- RemoveFromQueue

Tools
- EncryptData

LegalSigning
- GetSigningToken
- GetSignerList
- SetSignerOrder
- CheckStatus
- GetData
- RemoveData

نکته: *در این نسخه احراز هویت (Authentication) فراخواننده سرویس‌ها بر اساس نام کاربری و گذرواژه است. در نسخه آتی احراز هویت مبتنی بر challenge – sign خواهد بود. کنترل دسترسی (Authorization) نیز بر اساس JWT انجام می‌شود.*
در تمامی فراخوانی‌ها، مجاز بودن نشانی IP فراخواننده API متناسب با IPاهای معرفی شده در حساب کاربری بررسی می‌شود.

سرویس احراز هویت (Auth Service)

<Protocol>://<ISGApi URL>/api/v2/Auth

Login API

<Protocol>://<ISGApi URL>/api/v2/Auth/login

در گام اول تعامل با درگاه، لازم است احراز هویت انجام شود. با فراخوانی Login API و ارسال نام کاربری و گذرواژه، یک JWT دریافت می‌شود که در تعاملات آتی مورد استفاده قرار می‌گیرد. پارامترهای ارسالی به این API در کلاس AuthRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس AuthResponse است. پس از احراز هویت موفقیت‌آمیز و دریافت accessToken در فراخوانی‌های بعدی سایر APIها باید توکن دریافتی را در بخش authorize مربوط به header درخواست خود قرار دهید:

Authorization value: Bearer accessToken

تنظیمات پیش‌فرض APIها طول عمری معادل ۱۰ دقیقه برای accessToken در نظر می‌گیرد. پس از طی این مدت accessToken منقضی خواهد شد. پس از انقضا می‌توانید در طول روز جاری با فراخوانی Refresh API و ارسال همزمان accessToken و refreshToken توکن جدیدی دریافت نمایید که به طور مجدد، برای ۱۰ دقیقه قابل استفاده خواهد بود. طول عمر accessToken و refreshToken توسط مدیر سیستم قابل تنظیم است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

Login API	
Request	
Method	Post
Body Content-Type	application/json
Schema	AuthRequest

	{
	"userName": "string",
	"password": "string"
	}

Response	
Status Code	200 Success / 401 Unauthorized
200 Schema	AuthResponse

	{
	"accessToken": "string",
	"refreshToken": {
	"username": "string",
	"token": "string",
	"expireAt": "date time"
	}
	}

401 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

Logout API

<Protocol>://<ISGApi URL>/api/v2/Auth/logout

پس از اتمام تعامل با درگاه می‌توان با فراخوانی Logout API و ارسال refreshToken دریافتی در هنگام login، به درگاه اعلام نمود که JWT مرتبط را معدوم نماید. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

Logout API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Parameters	refreshToken: "string"
Response	
Status Code	204 NoContent

سرویس توکن احراز هویت (Token Service)

<Protocol>://<ISGApi URL>/api/v2/Token

Refresh

<Protocol>://<ISGApi URL>/api/v2/Token/refresh

با استفاده از این API توکن احراز هویت JWT فعلی نوسازی شده و یک JWT جدید ارایه می‌شود. پارامترهای ارسالی به این API در کلاس JwtRefreshRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس AuthResponse است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

Refresh API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	JwtRefreshRequest

	{
	"accessToken": "string",
	"refreshToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	AuthResponse

	{
	"accessToken": "string",
	"refreshToken": {
	"username": "string",
	"token": "string",
	"expireAt": "date time"

	} }

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

Revoke

<Protocol>://<ISGApi URL>/api/v2/Token/revoke

عملکرد و ساختار این API همانند Logout API بوده و با فراخوانی آن JWT مرتبط ابطال خواهد شد. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

Revoke API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Parameters	refreshToken: "string"
Response	
Status Code	204 NoContent

در نمونه برنامه پیوست این سند به نام ISG.Demo، فراخوانی‌های مربوط به احراز هویت و نوسازی JWT پیاده‌سازی شده و قابل بررسی برای توسعه‌دهندگان می‌باشد.

سرویس ارسال درخواست امضا (Signing Service)

<Protocol>://<ISGApi URL>/api/v2/Signing

ChangePassword

<Protocol>://<ISGApi URL>/api/v2/Signing/ChangePassword

تغییر گذرواژه از دو طریق امکان‌پذیر است. توسط مدیر سیستم در بخش مدیریت کاربران گذرواژه تغییر داده شود یا با فراخوانی ChangePassword API این کار صورت پذیرد. پارامترهای ارسالی به این API در کلاس ChangePasswordRequest توصیف شده‌اند. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

ChangePassword API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	ChangePasswordRequest

	{
	"userName": "string",
	"password": "string"
	}

Response	
Status Code	200 Success
200 Schema	Boolean

AllTemplates

<Protocol>://<ISGApi URL>/api/v2/Signing/AllTemplates

این API فهرست الگوهای تعریف شده در حساب کاربری را ارائه می‌دهد. جهت بررسی اقلام الگو و کاربرد آن به بخش Template مراجعه نمایید. در فراخوانی این API نیازی به ارسال پارامتر وجود نداشته و داشتن JWT معتبر کفایت می‌نماید. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

AllTemplates API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	No Parameters
Response	
Status Code	200 Success / 400 BadRequest
200 Schema	Template[]

	[
	{
	"name": "string",
	"direction": "string",
	"contentType": number,
	"signType": number,
	"callbackUrl": "string",
	"callbackType": number,
	"hint": "string",
	"title": "string",
	"letSignerDownload": boolean,
	"validityDuration": number,
	"handwriteSignature": number,
	"issuersDn": "string",
	"version": "string" *
	}
	]

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

* در نسخه جاری مقدار version برای الگوها همیشه ۱.۰ است.

TemplateInfo

<Protocol>://<ISGApi URL>/api/v2/Signing/TemplateInfo

این API با دریافت نام الگو مشخصات الگوی موردنظر را ارائه می‌دهد. پارامترهای ارسالی به این API در کلاس TemplateRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس Template است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

TemplateInfo API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	TemplateRequest

	{
	"templateName": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	Template

	{
	"name": "string",
	"direction": "string",
	"contentType": number,
	"signType": number,
	"callbackUrl": "string",
	"callbackType": number,
	"hint": "string",
	"title": "string",
	"letSignerDownload": boolean,
	"validityDuration": number,
	"handwriteSignature": number,
	"issuersDn": "string",
	"version": "string" *
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

TemplateStructure

<Protocol>://<ISGApi URL>/api/v2/Signing/TemplateStructure

در صورتی‌که محتوای ارسالی برای امضا از نوع json باشد، لازم است ساختار نمایش اطلاعات آن به صورت html در الگوی مرتبط تعریف شود تا درگاه، اطلاعات json را مطابق با ساختار تعریف شده، نمایش دهد. TemplateStructure API ساختار الگوی json را ارائه می‌دهد.

نکته: *ساختار نمایش اطلاعات که به صورت html تعریف می‌شود با استفاده از قابلیت data-binding موجود در HTML5 عمل خواهد نمود. به دلایل حقوقی- امنیتی باید المنت‌های html موجود در ساختار ارایه شده، نظیر به نظیر با المنت‌های اطلاعات ارسالی در محتوای Json یکسان باشند تا امضاکننده، تمامی اطلاعات را پیش از امضا مشاهده نماید. در صورت سازگار نبودن ساختار نمایش html با اطلاعات json و عدم نمایش حتی یک المنت از محتوای ارسالی، درگاه امضا، محتوا را معتبر ندانسته، آن را نمایش نداده و اجازه امضای محتوا را نخواهد داد.*

پارامترهای ارسالی به این API در کلاس TemplateRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس TemplateBlob است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

TemplateStructure API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	TemplateRequest

	{
	"templateName": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	TemplateBlob

	{
	"structure": "html string",
	"logo": {
	"name": "string",
	"mediaType": "string",
	"content": "base64 string"
	}
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

GetSigningToken

<Protocol>://<ISGApi URL>/api/v2/Signing/GetSigningToken

با استفاده از این API محتوایی که باید امضا شود برای درگاه ارسال شده و در مقابل، یک شناسه ارجاع به نام SigningToken دریافت می‌گردد. هنگام فراخوانی این API اگر نام الگو ارسال شود، تمامی پارامترهای لازم به صورت خودکار از الگو برداشت خواهد شد و فقط ارسال اطلاعات محتوا به همراه نام الگو الزامی خواهد بود. در صورتی‌که هر پارامتر دیگری در این API مقداردهی و ارسال شود، اولویت با پارامتر ارسالی بوده و از مقدار مشابه آن در الگو صرف‌نظر خواهد شد.

برای محتوای نوع Json ارسال نام الگو و وجود ساختار html جهت نمایش اطلاعات در آن، الگو الزامی است. برای سایر انواع محتوا، ارسال نام الگو اختیاری می‌باشد. الگوی دارای ساختار، می‌تواند آرم سازمانی متفاوتی از آنچه در حساب کاربری معرفی شده است داشته باشد، هنگام ایجاد یک ساختار جدید، به صورت پیش‌فرض آرم سازمانی در ساختار الگو قرار می‌گیرد. در صورت تمایل می‌توانید برای الگوهای مربوط به json آرم سازمانی جداگانه‌ای معرفی نمایید.

هنگام ارسال محتوا به این نکته توجه داشته باشید که انواع مختلف data در پروتکل html رفتار متفاوتی دارند. به عنوان مثال کاراکترهای / و + که در بعضی از base64 string ها وجود دارند، در URL جزو کاراکترهای غیرمجاز و دارای ریسک شناسایی می‌شوند. یا ارسال اطلاعات html به دلیل کاراکترهای > و < ... به صورت عادی پذیرفته نمی‌شود.

در صورتی‌که محتوای ارسالی Json باشد، باید ساختار Html متناظر با آن تعریف شده باشد.

برای تعریف ساختار Html، تک تک پارامترها باید نظیر به نظیر در فایل json ارسالی نیز وجود داشته باشد.

در فایل Html پارامترهای داینامیک را باید به شکل ذیل، تعریف کرد: بطورمثال:

Json File: {StudentFullName: 'Test'}

Html File: {{StudentFullName}}

- اگر در فایل Json، پارامتری با ساختار Nested-Json یا Sub-Json وجود دارد در ساختار Html باید به شکل ذیل قرار داده شود:

Json File:
```
{
    "PersonalInformation": {
                    "IdentificationNumber":"1051002052"
                    }
}
```

Html File: {{PersonalInformation.IdentificationNumber}}

- اگر در فایل Json، پارامتری با ساختار **آرایه** وجود دارد، در ساختار Html باید در تگ **<btable> </btable>** به شکل ذیل قرار داده شود:

Json File:
```
{
    "accountOwners": [
        {"idCode": "0064689565",
```

```
                "name": "زهرا رسولیان"}
                    ],
            "serialNo": "1000021",
        }
```

Html File:
```
<div>
    <b><label>لیست صاحبان حساب چک:</label></b>
    <btable {{accountOwners[]}}>
        <br/><label>نام: {{accountOwners[].name}} </label>
        <label>کد ملی/شناسه ملی: {{accountOwners[].idCode}} </label>
    </btable>
</div>
```

در صورتی‌که مقدار آرایه در فایل Json خالی باشد، باید به شکل ذیل در فایل json ارسال گردد، اما در فایل Html تک تک پارامترها مثل بالا تعریف گردد:

Json File:
```
{
    "accountOwners": []
}
```

که در این صورت زمان نمایش با مقدار خالی آنها را نمایش می دهد.

در هنگام ارسال محتوا به نکات زیر توجه داشته باشید:

با توجه به اینکه هم امکان امضای رمزگذاری‌شده هست، هم مدل امضای Plain، اگر از مدل امضای رمزگذاری‌شده استفاده می‌گردد، باید محتوای ارسالی برای امضا، با توجه به توضیحاتی که برای EncryptData آورده شده است، یک رشته Base64 رمزگذاری‌شده باشد و همینطور مقدار پارامتر IsEncrypted، در این Api، true ارسال گردد.

اما اگر از مدل امضای Plain استفاده می‌شود، مقدار پارامتر IsEncrypted باید false ارسال شود و توجه به نکات ذیل نیز حائز اهمیت است:

- اطلاعات فایل pdf را ابتدا به base64 string تبدیل نموده سپس URL Encode نمایید.
- اطلاعات json را ابتدا اطمینان حاصل کنید که UTF-8 است، سپس URL Encode نموده و ارسال نمایید.
- اطلاعات نوع html را به صورت Html Encode ارسال کنید.
- اطلاعات text را به صورت URL Encode ارسال نمایید.
- اطلاعات Chakad را به صورت json که UTF-8 است، سپس URL Encode نموده و ارسال نمایید.

متناسب با نوع محتوای ارسالی، محدودیتی در نوع امضا وجود دارد:

- محتوای از نوع Html و Text را می‌توان با انواع مختلف امضا، امضا نمود.
- محتوای از نوع pdf فقط قابلیت امضای pdf دارد.
- محتوای از نوع json را باید CMS (detached) امضا کرد.
- محتوای از نوع Chakad فقط قابلیت امضای CMS(attach) دارد.

توجه به این نکته الزامی ست که محتوا از نوع چکاد، با utf8 encoding امضا می شود و صرفا برای امضای چک بانک مرکزی، کاربرد دارد.

تمامی این موارد را می‌توانید در برنامه نمونه‌ای به نام ISG.Demo که در پیوست این سند قرار دارد ملاحظه فرمایید.

در صورتی که محتوای ارسالی از نوع text یا html بوده و امضای درخواستی از نوع pdf باشد، درگاه به صورت خودکار محتوا را به pdf نسخه ۱.۶ (acrobat 7) تبدیل می‌نماید. اما اگر محتوای ارسالی از نوع pdf باشد، درگاه می‌تواند نسخه‌های پایین‌تر از ۱.۶ را به نسخه ۱.۶ تبدیل نموده و بقیه نسخه‌ها را بدون تغییر ارایه دهد. تبدیل نسخه pdf پایینتر از ۱.۶ به نسخه ۱.۶ به صورت پیش‌فرض غیرفعال بوده و توسط مدیر وب سرور درگاه قابل فعال شدن است. بنابراین توصیه می‌گردد اگر محتوای ارسالی به درگاه امضا، از نوع pdf می‌باشد، نسخه آن ۱.۶ به بالا باشد.

نسخه ۱.۶ کمترین شماره نسخه برای برخی از فعالیت‌های خاص بر روی سند، از جمله افزودن مهر سازمانی است.

نکته: این‌گونه فعالیت‌های خاص، خارج از درگاه امضا انجام می‌شود و به عهده سیستم مربوطه است. درگاه امضا فقط به منظور ایجاد تسهیلات برای فعالیت‌های احتمالی آتی، این قابلیت را فراهم نموده است.

نکته: حجم محتوای ارسال شده برای امضا، با توجه به مقداری که توسط مدیر وب سرور تنظیم شده، قابل کنترل است.

در GetSigningToken API به صورت اختیاری می‌توان با تعیین مشخصات امضاکننده و نوع پیام‌رسانی، همزمان با ارسال محتوا، پیامی هم برای گوشی همراه امضاکننده ارسال نمود تا جهت رویت و امضای محتوا به درگاه مراجعه نماید. در صورتی‌که تمایل به ارسال پیام اطلاع‌رسانی به گوشی همراه امضاکننده را ندارید، می‌توانید Notification -> Type را معادل QueueOnly قرار دهید تا بدون ارسال پیام، صف امضا تشکیل شود. برای ارجاع آن دسته از کاربران که فاقد نرم‌افزار لازم روی تلفن همراه خود هستند یا از طریق مرورگر قصد امضای محتوا را دارند، پس از دریافت SigningToken می‌توان مرورگر کاربر را با فرمت زیر به درگاه امضا هدایت نمود:

<Protocol>://<ISG URL>/SigningToken

مثال: https://isg.pki.co.ir/1DF6A769-4E82-49AB-8835-3CB146E4F00

همچنین با تعیین مشخصات گواهینامه امضاکننده، امکان امضای محتوا، محدود به دارنده آن گواهینامه خواهد بود. بدین منظور در پارامتر issuersDn می‌توان یک یا فهرستی از اسامی صادرکنندگان مجاز گواهینامه را قرار داد که در این روش، امکان امضای محتوا، فقط به دارندگان گواهینامه از این مراکز، محدود می‌گردد.

برای محدود کردن امضاکننده، روش دیگری هم وجود دارد که می‌توان در پارامتر signerSubjectDn، کدملی یک شخص را به صورت "SERIALNUMBER=NationalID" قرار داد که در این روش، امکان امضا به یک امضاکننده، محدود می‌گردد.

ارسال پارامتر metadata به منظور دریافت آن در GetData API از دیگر اقلام اختیاری در فراخوانی این API است. این پارامتر در GetSigningToken API توسط بهره‌بردار ارسال و بدون هیچ تغییری در GetData API دریافت می‌شود. از این پارامتر به عنوان برقراری mapping اطلاعات درگاه امضا با برنامه کاربردی بهره‌بردار استفاده می‌گردد. بدین‌صورت که بهره‌بردار در زمان ارسال محتوا به درگاه امضا، در GetSigningToken API در این پارامتر، مقدار کلیدی خود را قرار می‌دهد و پس از دریافت محتوای امضا در GetData API می‌تواند با دریافت آن، ارتباط برنامه کاربردی خود با محتوای امضای دریافت‌شده را برقرار نماید.

هرگاه امضاکننده اقدام به امضای محتوا نماید، در صورتی که امضا و گواهینامه امضاکننده معتبر باشند، درگاه امضا شدن محتوا را به نشانی callback اعلام می‌نماید.

در پارامتر callbackType نوع ارسال نتیجه‌ی امضا به بهره‌بردار مشخص می‌گردد. دو نوع callback وجود دارد: client-side و Server-side.

پارامتر callbackurl نیز نشانی فراخواننده سرویس می‌باشد و با توجه به نوع callback مقداردهی می‌گردد.

بدین صورت که اگر نوع callback به صورت clent-side باشد، بهره‌بردار باید متدی پیاده‌سازی کرده باشد که از جنس HttpGet بوده و پارامتر callbackurl را نیز به شکل ذیل مقداردهی نماید:

<Protocol>://<Beneficiary URL>**/getResult**
https://isg.pki.co.ir/demo/home/getresult

و اگر نوع callback از نوع server-side باشد، بهره‌بردار باید متدی از جنس HttpPost پیاده‌سازی کرده باشد و مقدار پارامتر callbackurl نیز به شکل ذیل مقداردهی شود:

<Protocol>://<Beneficiary URL>**/postResult**
https://isg.pki.co.ir/demo/home/postresult

نکته: در صورتی که نشانی callback از نوع server-side باشد، درگاه یک http request به نشانی callback ارسال نموده و در انتظار پاسخ http status باقی می‌ماند تا نتیجه را به امضاکننده اعلام نماید. توضیحات بیشتر در GetData API آورده شده است.

نکته: پارامتر notificationResponse برای حفظ Backward compatibility در خروجی این سرویس وجود دارد و پیشنهاد می‌گردد در پیاده‌سازی سرویس‌ها، از این پارامتر استفاده نشود و به جای آن از پارامتر notificationOutput استفاده گردد.

نکته: با توجه به اینکه اطلاعات مربوط به پیام‌رسانی به گوشی همراه (Push Notification)، تعیین امضاکننده خاص (Signer Subject DN) و metadata در الگوها تعریف نمی‌شوند، در صورت نیاز، باید در هر درخواست ارسال شوند. پارامترهای ارسالی به این API در کلاس SigningRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس SigningResponse است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

GetSigningToken API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	SigningRequest
	{ "dataToSign": "string", "templateName": "string", "direction": "string", "contentType": number, "signType": number, "callbackType": number, "callbackUrl": "string", "hint": "string",

	"title": "string",
	"letSignerDownload": boolean,
	"validityDuration": number,
	"issuersDn": [
	"string"
	],
	"signerSubjectDn": "string",
	"handwriteSignature": number,
	"contentImage": {
	"image": "string",
	"imageType": "string",
	"page": number,
	"lowerLeftX": number,
	"lowerLeftY": number,
	"upperRightX number,
	"upperRightY": number
	},
	"metadata": "string",
	"notification": {
	"token": "string",
	"type": number
	},
	"IsEncrypted": boolean
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	SigningResponse

	{
	"signingToken": "string",
	"notificationOutput": {
	"notifSucceedCount": number,
	"notificationResponse": notificationResponse,
	"issuccess": Boolean,
	"description": "string",
	"errorCode": number,
	"errorMessage": "string",
	},
	"notificationResponse": notificationResponse,
	"succeeded": boolean,
	"errorCode": number,
	"errorMessage": "string"
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

CheckStatus

<Protocol>://<ISGApi URL>/api/v2/Signing/CheckStatus

پس از ارسال محتوا و دریافت شناسه ارجاع (Signing Token) هر زمان که لازم باشد می‌توان با فراخوانی این API وضعیت جاری محتوا را بررسی نمود. پارامترهای ارسالی به این API در کلاس GeneralRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق ContentStatus است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

CheckStatus API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	GeneralRequest

	{
	"signingToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	"contentStatus": number

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

GetData

<Protocol>://<ISGApi URL>/api/v2/Signing/GetData

هرگاه امضاکننده اقدام به امضای محتوا نماید، در صورتی که امضا و گواهینامه امضاکننده معتبر باشند، درگاه امضا شدن محتوا را به نشانی callback اعلام می‌نماید. در این زمان با فراخوانی این API تمامی اطلاعات مرتبط با امضا ارائه خواهد شد.

نکته: در صورتی که نشانی callback از نوع server-side باشد، درگاه یک http request به نشانی callback ارسال نموده و در انتظار پاسخ http status باقی می‌ماند تا نتیجه را به امضاکننده اعلام نماید. بنابراین توصیه می‌شود بهره‌برداران پس از دریافت اعلام امضا شدن محتوا، ابتدا پاسخ http status 200 را به درگاه بازگردانده، سپس با یک وقفه کوتاه GetData API را فراخوانی نمایند تا سرور درگاه با کارایی مناسب وظایف خود را انجام دهد. ایجاد یک queue با وقفه‌ای در حدود ۵۰۰ میلی ثانیه فرصت لازم را در اختیار سرور درگاه قرار می‌دهد تا همزمانی امضاکنندگان را مدیریت نموده و پایان فرآیند را به امضاکننده اعلام نماید. این نکته نیز در ISG.Demo در QueueService پیاده‌سازی شده و قابل بررسی می‌باشد. پارامترهای ارسالی به این API در کلاس GeneralRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس SignatureResponse است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

GetData API	
Request	
Method	post

Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	GeneralRequest

	{
	"signingToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	SignatureResponse

	{
	"certificate": "base64 string",
	"serialNumber": "string", *
	"digest": "string", **
	"signature": "url-encoded base64String",
	"metadata": "string",
	"signType": number,
	"sendTime": "string",
	"senderIP": "string",
	"signTime": "string",
	"signerIP": "string",
	"issuerDn": "string",
	"subjectDn": "string",
	"signerCn": "string", ***
	"thumbprint": "string"
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

* مقدار serialNumber شماره سریال گواهینامه امضاکننده بر اساس CA صادر کننده گواهینامه است.

** در صورتی‌که امضا از نوع PDFSign باشد Digest امضا شده دریافت می‌شود. در سایر انواع امضا، Digest امضا نشده دریافت می‌گردد.

*** مقدار signerCn اشاره به common name گواهینامه امضاکننده دارد. مقدار این آیتم در گواهینامه‌های صادره از سوی مراکز میانی صدور گواهینامه‌های کشوری متفاوت بوده اما در هر صورت مشخص‌کننده شناسه یکتای صاحب گواهینامه می‌باشد. به عنوان مثال در گواهینامه‌های کارت ملی، شناسه ملی هر فرد در آیتم common name گواهینامه ذخیره شده است.

نکته: وجود کاراکترهای / و + در بعضی از base64 string ها موجب می‌شود تا مقادیر base64string از قبیل certificate و signature به صورت خودکار URL Encode شوند. بنابراین هنگام دریافت مقادیر base64string با استفاده از متد IsBase64String که در مولفه DTO فراهم شده است، مقدار دریافتی را بررسی نمایید. در صورتی‌که IsBase64String مقدار false برمی‌گرداند، لازم است base64string دریافتی را URL Decode نموده، سپس در توابعی مانند Verify استفاده کنید. در استفاده از توابع PKTB لازم است رشته‌های ارسالی از نوع Unicode (UTF-16) باشند. هم در PKTB توابع لازم برای تبدیل رشته‌های متنی به Unicode وجود دارد و هم می‌توان با داشتن متن اصلی (message)، آرایه بایتی Unicode آن را استخراج و در توابعی نظیر Verify استفاده نمود.

اطلاعات html و xml نیز به دلیل وجود کاراکترهای > و < ... به صورت عادی پذیرفته نمی‌شود و باید تمهیدات لازم در نقل و انتقال و دریافت این‌گونه اطلاعات در نظر گرفته شود. جهت بررسی دقیق موضوع به کلاس SignatureExtensions در پروژه ISG.Demo مراجعه نمایید.

RemoveData

<Protocol>://<ISGApi URL>/api/v2/Signing/RemoveData

با فراخوانی این API و ارسال شناسه ارجاع (Signing Token)، تمامی اطلاعات مربوط به محتوا و امضای آن حذف خواهند شد. متناسب با پارامترهای ارسالی در SigningRequest مقدار بازگشتی این API عددی بین ۳ تا ۵ خواهد بود که بیانگر حذف موفقیت‌آمیز تعداد رکوردهای مرتبط با محتوای ارسالی و امضای آن است. پارامترهای ارسالی به این API در کلاس GeneralRequest توصیف شده‌اند. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

RemoveData API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	GeneralRequest

	{
	"signingToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	RowsAffected": number

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

ResendNotification

<Protocol>://<ISGApi URL>/api/v2/Signing/ResendNotification

در صورتی‌که پیام اطلاع‌رسانی با موفقیت ارسال شده ولی به گوشی همراه کاربر نرسیده باشد، با استفاده از این API می‌توان اقدام به ارسال مجدد همان پیام قبلی نمود. پیش‌نیاز استفاده از این API ارسال اطلاعات پیام‌رسانی در هنگام ارسال محتوا با استفاده از GetSigningToken API است. این API فقط برای محتوایی اقدام به ارسال مجدد پیام اطلاع‌رسانی می‌کند که آن محتوا هنوز امضا نشده و منقضی هم نشده باشد. در سایر حالت‌ها، کد خطای ۲۴ (ContentExpired) را دریافت می‌نمایید. پارامترهای ارسالی به این API در کلاس GeneralRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق کلاس NotificationOutput است. لازم به توضیح است که پارامتر notifSucceedCount تعداد نوتیفیکیشن ارسال شده‌ی موفق به کاربر(بر حسب تعداد برنامه‌های امضای کاربر) را نمایش می‌دهد. پارامتر notificationResponseList فهرست نوتیفیکیشن‌های ارسال شده را نمایش می‌دهد که در این فهرست، علاوه بر ارسال های موفق، ارسال های ناموفق هم قرار دارد.

مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

ResendNotification API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	GeneralRequest

	{
	"signingToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	NotificationOutput

	{
	"notifSucceedCount": number,
	"notificationResponseList": IEnumerable<notificationResponse>,
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

سرویس Notification (Notification Service)

<Protocol>://<ISGApi URL>/api/v2/Notification

این سرویس دارای دو API است:

GetSigningQueueList

<Protocol>://<ISGApi URL>/api/v2/Notification/GetSigningQueueList

این API برای نمایش فهرست درخواست‌های امضای ارسال شده به کاربر می‌باشد. به این صورت که بهربردار، شناسه‌ملی یا شماره موبایل امضاکننده را ارسال کرده و فهرست درخواست‌های در انتظار امضای آن شخص را دریافت می‌نماید.

برای ارسال نوتیفیکیشن در GetSigningToken، می‌توان کدملی یا شماره موبایل شخص امضاکننده را ارسال کرد، هرکدام که هنگام ارسال درخواست ارسال شده، زمان دریافت فهرست صف امضا نیز در پارامتر NotifToken باید همان پارامتر قرار داده شود.

مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

GetSigningQueueList API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	SigningQueueRequest
	{ " notifToken": "string" }
Response	
Status Code	200 Success / 400 BadRequest
200 Schema	SigningQueueResponse
	{ "notificationQueueList": [{ "signingToken":"", "owner": "", "subject": "", "gateway": "", "imageURL": "", "date": "" }], "issuccess": true, "description": null, "errorCode": 0, "errorMessage": null }
400 Schema	{

	"errorCode": number, "errorMessage": "string" }

RemoveFromQueue

<Protocol>://<ISGApi URL>/api/v2/Notification/RemoveFromQueue

این API برای حذف درخواست امضای ارسال شده به امضاکننده می‌باشد.

مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

RemoveFromQueue API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	GeneralRequest

	{
	"signingToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	GeneralResponse

	{
	"issuccess": Boolean
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

سرویس Tools (Tools Service)

<Protocol>://<ISGApi URL>/api/v2/Tools

این سرویس دارای یک API است:

EncryptData

<Protocol>://<ISGApi URL>/api/v2/Tools/EncryptData

این API اقدام به Encrypt کردن محتوای امضا می‌نماید. به این صورت که بهره‌بردار باید محتوایی که می‌خواهد برای درگاه امضا ارسال کند، به همراه یک رمز که باید برای امضاکننده ایجاد و ارسال شود را دریافت و این اطلاعات را رمزگذاری کرده و محتوای رمزگذاری شده را به صورت یک رشته base64 برمی‌گرداند.

در هنگام ارسال محتوا به نکات زیر توجه داشته باشید:

- اطلاعات فایل pdf را ابتدا به base64 string تبدیل نموده سپس URL Encode نمایید.
- اطلاعات json را ابتدا اطمینان حاصل کنید که UTF-8 است، سپس URL Encode نموده و ارسال نمایید.
- اطلاعات نوع html را به صورت Html Encode ارسال کنید.
- اطلاعات text را به صورت URL Encode ارسال نمایید.

Encrypt API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	GeneralRequest

	{
	"data": "string",
	"Pin": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	{"EncryptedData": "string",

	“IsEncrypted” : “bool” }

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

سرویس ارسال درخواست امضا برای دارندگان حق امضا (LegalSigning Service)

GetSigningToken

<Protocol>://<ISGApi URL>/api/v2/LegalSigning/GetSigningToken

از این Api برای ارسال درخواست امضا به صاحبین امضا مندرج در روزنامه رسمی، استفاده می‌گردد. در واقع محتوایی که باید امضا شود برای درگاه ارسال شده و در مقابل، یک شناسه ارجاع چندگانه به نام LegalSigningToken دریافت می‌گردد.

نکته‌ی قابل توجه اینست که در سرویس امضای چندگانه، فقط انواع امضای PDF یا CMS کاربرد دارد و امکان استفاده از Template نیز در این مدل، فراهم نمی‌باشد.

GetSigningToken API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	LegalSigningRequest
	{ "dataToSign": "string", "templateName": "string", "direction": "string", "contentType": number, "signType": number, "callbackType": number, "callbackUrl": "string", "hint": "string", "title": "string", "letSignerDownload": boolean, "validityDuration": number, "issuersDn": ["string"], "signerSubjectDn": "string", "handwriteSignature": number, "contentImage": { "image": "string", "imageType": "string", "page": number, "lowerLeftX": number, "lowerLeftY": number, "upperRightX number, "upperRightY": number }, "metadata": "string", "notification": null, "IsEncrypted" : Boolean,

	"nationalID": "string", "priorityType": number }
Response	
Status Code	200 Success / 400 BadRequest
200 Schema	LegalSigningResponse

	{
	"legalSigningToken": "string",
	"succeeded": Boolean
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

اطلاعات ورودی این Api مشابه پارامترهای ورودی GetSigningToken شخص حقیقی است، با این تفاوت که دو پارامتر نیز اضافه شده که از پارامتر nationalId برای ارسال شناسه شخص حقوقی(شرکت/سازمان) استفاده می‌گردد.

از پارامتر Proritytype نیز برای مشخص نمودن نوع ترتیب‌دهی استفاده می شود. به این شکل که اگر مقدار این پارامتر Auto باشد، ابتدا ترتیب امضاکنندگان در فهرست صاحبین امضا به صورت سیستمی تنظیم می‌شود و سپس درخواست امضا برای اولین امضاکننده در فهرست، ارسال می‌گردد و پس از امضای موفقیت آمیز هر امضاکننده، درخواست امضا به ترتیب برای نفر بعدی ارسال می‌شود.

در مورد مدت زمان درخواست امضا(ValidityDuration) که بر حسب دقیقه می‌باشد، توجه به این نکته الزامی است که مدت زمانی که در درخواست امضا ارسال می‌گردد مجموع زمانی است که همه امضاکنندگان باید در آن مدت زمان، درخواست خود را امضا نمایند، در غیر این صورت اگر مدت زمان امضا به پایان برسد و همه‌ی امضاکنندگان، امضا را انجام نداده باشند، آن درخواست منقضی شده و باید دوباره درخواست امضا توسط بهره‌بردار ایجاد شده و برای تمام امضاکنندگان ارسال گردد.

GetSignerList

<Protocol>://<ISGApi URL>/api/v2/LegalSigning/GetSignerList

اگر نوع ترتیب‌دهی، به صورت Manual تعیین شده باشد، ابتدا بهره‌بردار باید با استفاده از این سرویس، فهرست امضاکنندگان را از سرویس دریافت کند.

GetSignerList API	
Request	
Method	Post
Header	Authorization: Bearer accessToken

Body Content-Type	application/json
Schema	LegalGeneralRequest

	{
	"legalSigningToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest

200 Schema	"CompanySigner": [
	{
	"nationalCode" : "string",
	"name": "string"
	}
	]

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

SetSignerOrder

<Protocol>://<ISGApi URL>/api/v2/LegalSigning/SetSignerOrder

اگر نوع ترتیب‌دهی، به صورت Manual باشد، بهره‌بردار باید برای هر یک از امضاکنندگان، با استفاده از این سرویس، ترتیب امضا را مشخص کند. پس از تعیین ترتیب، درخواست امضا به ترتیب مشخص‌شده برای صاحبین امضا ارسال می‌گردد و نتیجه‌ی این Api در صورت موفقیت، مقدار true در غیر این صورت، مقدار false می‌باشد.

با توجه به اینکه پس از تعیین ترتیب، درخواست امضا برای امضاکنندگان، ارسال می‌گردد، امکان تغییر ترتیب امضاکنندگان وجود ندارد و تنها یکبار امکان تعیین ترتیب توسط این Api را خواهید داشت.

SetSignerOrder API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	CompanySignerOrderRequest

	{
	"legalSigningToken": "string",
	“CompanySignerOrderList”: [
	“nationalCode”: “string”,
	“orderId”: number
	]
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	bool

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

CheckStatus

<Protocol>://<ISGApi URL>/api/v2/LegalSigning/CheckStatus

از این Api برای بررسی وضعیت درخواست امضا، استفاده می‌گردد.

پس از ارسال محتوا و دریافت شناسه ارجاع چندگانه (LegalSigningToken) هر زمان که لازم باشد می‌توان با فراخوانی این API ، اطلاعات زیر را دریافت نمود:

- وضعیت جاری درخواست
- شناسه ارجاع (SigningToken) درخواست ارسال شده برای هر یک از امضاکنندگان به همراه وضعیت امضای آن

پارامترهای ارسالی به این API در کلاس LegalGeneralRequest توصیف شده‌اند. پاسخ دریافتی نیز مطابق LegalSigningStatusResponse است. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

CheckStatus API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	LegalGeneralRequest

	{
	"legalSigningToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest

200 Schema	"LegalSigningStatusResponse":
	{
	“LegalRequestStatus”: LegalSignatureStatus,
	“LegalSignerRequsetStatus”:
	[
	{“nationalCode” : “string”,
	“requestStatus”: LegalSignatureStatus
	“signinToken”: "string"
	}
	]
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

GetData

<Protocol>://<ISGApi URL>/api/v2/LegalSigning/GetData

هرگاه درخواست امضا توسط تمامی امضاکنندگان با موفقیت انجام شده باشد، درگاه، امضا شدن محتوا را به نشانی callback اعلام می‌نماید. در این زمان با فراخوانی این API و ارسال شناسه ارجاع چندگانه (LegalSigningToken)، تمامی اطلاعات مرتبط با امضا ارائه خواهد شد.

نکته: با توجه به اینکه نشانی callback از نوع server-side می‌باشد، درگاه یک http request به نشانی callback ارسال نموده و در انتظار پاسخ http status باقی می‌ماند تا نتیجه را به امضاکننده اعلام نماید. بنابراین توصیه می‌شود بهره‌برداران پس از دریافت اعلام امضا شدن محتوا، ابتدا پاسخ http status 200 را به درگاه بازگردانده، سپس با یک وقفه کوتاه GetData API را فراخوانی نمایند تا سرور درگاه با کارایی مناسب وظایف خود را انجام دهد. ایجاد یک queue با وقفه‌ای در حدود ۵۰۰ میلی ثانیه فرصت لازم را در اختیار سرور درگاه قرار می‌دهد تا همزمانی امضاکنندگان را مدیریت نموده و پایان فرآیند را به امضاکننده اعلام نماید.

پارامترهای ارسالی به این API در کلاس LegalGeneralRequest توصیف شده‌اند. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

GetData API	
Request	
Method	post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	LegalGeneralRequest

	{
	"legalSigningToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	

	{
	"LegalRequestStatus": LegalSignatureStatus,
	"LegalSignature": "url-encoded base64String",
	"LegalSignDate": Datetime
	}

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

RemoveData

<Protocol>://<ISGApi URL>/api/v2/LegalSigning/RemoveData

با فراخوانی این API و ارسال شناسه ارجاع چندگانه (LegalSigningToken)، تمامی اطلاعات مربوط به محتوای درخواست و امضای آن حذف خواهند شد. متناسب با پارامترهای ارسالی در LegalSigningRequest مقدار بازگشتی این API عددی بین ۳ تا ۵ خواهد بود که بیانگر حذف موفقیت‌آمیز تعداد رکوردهای مرتبط با محتوای ارسالی و امضای آن است. پارامترهای ارسالی به این API در کلاس LegalGeneralRequest توصیف شده‌اند. مشخصات request ارسالی به این API و response دریافتی به شرح جدول زیر است:

RemoveData API	
Request	
Method	Post
Header	Authorization: Bearer accessToken
Body Content-Type	application/json
Schema	LegalGeneralRequest

	{
	"legalSigningToken": "string"
	}

Response	
Status Code	200 Success / 400 BadRequest
200 Schema	RowsAffected": number

400 Schema	{
	"errorCode": number,
	"errorMessage": "string"
	}

شمای تبادل اطلاعات (Schemas)

در این بخش schema های مورد استفاده در API های سرویس‌ها توصیف شده‌اند. تمامی schema های استفاده شده در مؤلفه ISG.DTO وجود داشته و توسعه دهندگان .NET و Java می‌توانند از نسخه سازگار با پلتفرم خود استفاده نمایند.

AuthRequest		
UserName	String	Not null
Password	String	Not null

- در نسخه آتی مکانیزم احرازهویت تغییر خواهد نمود و به جای ارسال نام کاربری و گذرواژه یک challenge از سرویس مربوط دریافت و Sign آن به منظور احراز هویت ارسال خواهد شد.

AuthResponse		
AccessToken	String	Not null
RefreshToken	JwtRefreshToken	Not null

- طول عمر access-token و refresh-token متناسب با نیازمندی بهره‌بردار قابل تنظیم می‌باشد.

CertificateValidationRequest		
Certificate	Base64 string	Not null

ChangePasswordRequest		
UserName	String	Not null
CurrentPassword	String	Not null
NewPassword	String	Not null

ContentImage		
Image	Base64 String	Not null
ImageType	ImageType (String)	Not null
Page	Number (Int 1 to 32767)	Not null
LowerLeftX	Number (float)	Not null
LowerLeftY	Number (float)	Not null
UpperRightX	Number (float)	Not null
UpperRightY	Number (float)	Not null

- ContentImage فقط برای ارسال تصویر برای محتوای نوع PDF کاربرد دارد. این تصویر پیش از امضا نمایش داده نمی‌شود ولی در صورتی‌که امضاکننده اجازه دانلود PDF امضا شده را داشته باشد و پس از امضا آن را دانلود نماید، تصویر ارسالی به عنوان تگ امضا در موقعیت مشخص شده قرار خواهد داشت. همچنین با فراخوانی GetData API سند PDF دریافتی شامل تصویر ارسالی نیز خواهد بود.
- مقدار Image می‌بایست Base64String و URL Encode شده باشد.
- موقعیت تصویر بر اساس X+ و Y+ در محور مختصات محاسبه می‌گردد. به بیان دیگر نقطه (0, 0) در سمت چپ و پایین صفحه PDF قرار دارد.

ErrorResult		
ErrorCode	Int (4)	Default
ErrorMessage	String	Allow null

- فهرست کدهای خطا و شرح آن در جدول Error Codes در انتهای بخش ضمیمه آورده شده است.

GeneralRequest		
SigningToken	String	Not null

GeneralResponse		
IsSuccess	boolean	Not null
Description	String	Not null
ErrorCode	int	Not null
ErrorMessag	String	Not null

JwtRefreshRequest		
AccessToken	String	Not null
RefreshToken	Base64 String	Not null

JwtRefreshToken		
UserName	String	Not null
Token	Base64 String	Not null
ExpireAt	DateTime	Not null

JwtToken		
AccessToken	String	Not null
RefreshToken	JwtRefreshToken	Not null

Logo		
Name	String	Not null
MediaType	String	Not null
Content	Base64 String	Not null

Notification		
Token	String	Not null (App Token, National ID, Mobile Number, Email Address)
Type	NotificationType	Not null

- در صورتی‌که مقدار App Token نرم افزار گوشی همراه (mkeyone) را در اختیار داشته باشید، می‌توانید با مشخص نمودن Type (Apple Token or Google Token) توکن مذکور را جهت مشخص نمودن تلفن همراه گیرنده پیام، ارسال نمایید. در غیر این صورت با ارسال یکی از پارامترهای کد ملی، شماره تلفن همراه یا نشانی پست الکترونیکی که موقع ثبت‌نام در

mkeyone توسط کاربر ارائه شده است، تلفن همراه گیرنده پیام به صورت خودکار توسط API مربوط شناسایی خواهد شد. در این شرایط باید مقدار Type را برابر با Local قرار دهید.

NotificationRequest		
Notification	Notification	Not null
SigningToken	String	Not null
Owner	String	Not null
Subject	String	Not null

- به صورت پیش‌فرض نام فارسی فراخواننده API به عنوان Owner و عنوان معرفی شده در پارامتر Title (در هنگام ارسال محتوا توسط GetSigningToken API) به عنوان Subject در نظر گرفته شده و در پیام دعوت به امضا در تلفن همراه، به کاربر نمایش داده می‌شوند.

NotificationResponse		
Result	String	Not null
NotificationStatus	NotificationStatus	Not null
Success	Boolean	Not null

NotificationOutput		
NotifSucceedCount	Int	Not null
NotificationResponse	NotificationResponse	Not null
Issuccess	Boolean	Not null
Description	String	Not null
ErrorCode	Int	Not null
ErrorMessage	String	Not null

SignatureResponse		
Succeeded	Boolean	Not null
Certificate	Base64 String	Not null
SerialNumber	String	Not null
Digest	String	Not null
Signature	Base64 String	Not null
SignType	SignType	Not null
Metadata	String	Allow null
SendTime	DateTime	Not null
SenderIP	String	Not null
SignTime	DateTime	Not null
SignerIP	String	Not null
IssuerDn	String	Not null
SubjectDn	String	Not null
SignerCn	String	Not null
Thumbprint	String	Not null

- مقدار Serial Number شماره سریالی است که از سوی CA صادر کننده به گواهینامه اختصاص داده می‌شود.
- در صورتی‌که امضا از نوع PDFSign باشد Digest امضا شده دریافت می‌شود. در سایر انواع امضا، Digest امضا نشده دریافت می‌گردد.
- مقدار SignerCn مقدار Common Name درج شده در SubjectDn گواهینامه امضاکننده است.

SigningRequest		
DataToSign	Base64 string/Encoded Html/String	Not null
TemplateName	String	Allow null
Direction *	String	Allow null
ContentType	ContentType	Not null
SignType	SignType	Not null
CallbackType	CallbackType	Not null
CallbackUrl	String	Not null
Hint	String	Allow null
Title	String	Not null
LetSignerDownload	Boolean	Allow null
ValidityDuration **	Number	Allow null (default value: 14400)
IssuersDn ***	String Array	Allow null
SignerSubjectDn ****	String	Allow null
HandwriteSignature *****	HandwriteSignature	Allow null
ContentImage	ContentImage	Allow null
Metadata	String	Allow null
Notification	Notification	Allow null
IsEncrypted	Boolean	Not null

نوع DataToSign وابسته به نوع محتوای ارسالی (Content Type) می‌باشد.

- برای PDF ابتدا باید Base64 string، سپس Url Encode ارسال شود.
- برای HTML باید Encoded Html string ارسال شود.
- برای Text ارسال Text با توجه به URL Encode شدن آن کفایت می‌نماید.
- برای Json ابتدا باید UTF-8، سپس Url Encode ارسال گردد.
- برای Chakad، محتوا ابتدا باید UTF-8، سپس Url Encode ارسال گردد.

* پارامتر Direction فقط هنگام ارسال Text کاربرد دارد.

** از نسخه ۲.۴.۳ (ISG.Infrastructure) حداکثر زمان اعتبار محتوای امضا نشده، ۱۰ روز یا معادل ۱۴،۴۰۰ دقیقه(با توجه به مقدار تنظیم شده در فایل Setting) می‌باشد. مقادیر صفر و بزرگتر از ۱۴،۴۰۰ به صورت خودکار به ۱۴،۴۰۰(با توجه به مقدار تنظیم شده در فایل Setting) تغییر می‌کنند.

*** با ارسال پارامتر IssuersDn به صورت آرایه‌ای از string، گواهینامه‌های مجاز به امضا محدود به گواهینامه‌هایی می‌شوند که توسط یکی از CA های میانی که در این پارامتر ذکر شده‌اند صادر گردیده باشند. این مقدار برابر با مقدار SubjectDn ذکر شده در گواهینامه CA یا همان مقدار فیلد Issuer در گواهینامه امضاکننده است.

**** با ارسال پارامتر SignerSubjectDn گواهینامه مجاز به امضا، محدود می‌شود به یک گواهینامه خاص که فیلد SubjectDn آن دارای مقداری برابر با این پارامتر است. به بیان دیگر، فقط یک امضاکننده مشخص، مجاز به امضای محتوا خواهد بود.

***** در صورتی‌که قصد ارسال تصویر امضای فرد امضاکننده را با استفاده از پارامتر ContentImage برای امضای نوع PDF داشته باشید، باید مقدار پارامتر HandwriteSignature را معادل Content قرار دهید. ارسال هر مقدار دیگر برای پارامتر HandwriteSignature موجب صرف‌نظر از مقدار ContentImage خواهد شد.

امضای چک(Chakad):

امضای چک، تقریبا مشابه با ساختار امضای Json است به این صورت که حتما برای استفاده از آن باید محتوایی که قرار است نمایش داده شود در قالب فایل Json ارسال گردد و متناظر با آن باید یک Template هم در پنل مدیریت ISG تعریف شود که شامل ساختار Html باشد.

برای نمایش آیتم‌های Json ارسالی از ساختار Html که تعریف شده، استفاده می‌گردد و در نهایت فقط مقدار TBS که در فایل json ارسال شده، در درگاه اینترنتی امضا، امضا می‌گردد.

بنابراین نوع محتوای ارسالی(ContentType) باید Chakad قرار داده شود و نوع امضا(SignType) باید CMAttached باشد.

SigningResponse		
SigningToken	String	Not null
NotificationResponse	NotificationOutput	Not null
Succeeded	Boolean	Not null

SigningResult		
SigningToken	String	Not null

SigningQueueRequest		
NationalCode	String	Not null

SigningQueueResponse		
NotificationQueueList	IEnumerable<NotificationQueue>	Not null
Issuccess	boolean	Not null
Description	String	Not null
ErrorCode	int	Not null
ErrorMessage	String	Not null

NotificationQueue		
SigningToken	String	Not null
Owner	String	Not null
Subject	String	Not null

Gateway	String	Not null
ImageURL	String	Not null
Date	DateTime	Not null

Template		
Name	String	Not null
Direction	String	Not null
ContentType	ContentType	Not null
SignType	SignType	Not null
CallbackUrl	String	Not null
CallbackType	CallbackType	Not null
Hint	String	Allow null
Title	String	Not null
LetSignerDownload	Boolean	Not null
ValidityDuration	Int (2)	Min: 1, Max:14400
HandwriteSignature	HandwriteSignature	Not null
IssuersDn	String	Allow null
Version	String	Always 1.0

ایجاد Template، تعامل با سرویس Signing و به طور مشخص یکی از پرکاربردترین API آن یعنی GetSigningToken را تسهیل می‌نماید. در صورتی‌که برای تیپ‌های مختلف امضا Template های مجزا تعریف شود، هنگام فراخوانی GetSigningToken ارسال پارامترهای DataToSign و TemplateName کفایت نموده و باقی پارامترها به صورت خودکار از Template معرفی شده مقداردهی خواهند شد.

TemplateBlob		
Structure	String	Allow null
Logo	Logo	Not null

- به ازای هر Template این امکان وجود دارد که امضاکننده تصویر logo متفاوتی را مشاهده نماید.
- در صورتی‌که نوع محتوای ارسالی JSON یا Chakad باشد، لازم است ساختار نمایش اطلاعات JSON یا Chakad در قالب HTML تعریف و در آیتم Structure ذخیره گردد.

TemplateRequest		
TemplateName	String	Not null

LegalSigningRequest		
DataToSign	Base64 string/Encoded Html/String	Not null
TemplateName	String	Allow null
Direction *	String	Allow null
ContentType	ContentType	Not null
SignType	SignType	Not null
CallbackType	CallbackType	Not null
CallbackUrl	String	Not null

Hint	String	Allow null
Title	String	Not null
LetSignerDownload	Boolean	Allow null
ValidityDuration **	Number	Allow null (default value: 14400)
IssuersDn ***	String Array	Allow null
SignerSubjectDn ****	String	Allow null
HandwriteSignature *****	HandwriteSignature	Allow null
ContentImage	ContentImage	Allow null
Metadata	String	Allow null
Notification	Notification	Allow null
IsEncrypted	Boolean	Not null
NationalID	String	Not null
PriorityType	Number	Not null

LegalSigningResponse		
LegalSigningToken	String	Not null
Succeeded	Boolean	Not null

LegalGeneralRequest		
LegalSigningToken	String	Not null

LegalSigningResponse		
LegalSigningToken	String	Not null
Succeeded	Boolean	Not null

LegalSigningStatusResponse		
LegalRequestStatus	LegalSignatureStatus	Not null
LegalSignerListStatus	IEnumerable<LegalSignerRequsetStatus>	Not null

LegalSignerRequsetStatus		
NationalCode	String	Not null
RequestStatus	LegalSignatureStatus	Not null

ضمیمه

CallbackType		
ClientSide	1	Client-side callback
ServerSide	2	Server-side callback

CertificateStatus		
Valid	0	The certificate is valid.
InvalidPeriod	1	The certificate has expired.
InvalidChain	2	The certificate chain is invalid.
InvalidIntegrity	3	Certificate integrity is invalid.
InvalidKeyUsage	4	The certificate key usage is invalid.
OCSPRevoked	5	The certificate has been revoked (OCSP).
OCSPValidationUnKnown	6	Certificate validity status is unknown (OCSP).
CRLRevoked	7	The certificate has been revoked (CRL).
CRLOrOCSPValidationError	8	Certificate validation encountered a problem. One of the CRL or OCSP servers is not available.
CRLValidationUnKnown	9	Certificate validity status is unknown (CRL).
CRLAndOCSPValidationUnknown	10	Certificate validation encountered a problem. Both CRL or OCSP servers is not available.

ContentStatus		
Unknown	0	Unknown
NotExist	1	Not exist
NotSigned	2	Not signed yet
Expired	3	Expired
Signed	4	Signed
Delivered	5	Delivered

ContentType		
Html	1	Hyper Text Markup Language
Pdf	2	Portable Document Format
Text	3	Plain Text
Json	4	Java Script Object Notation
Chakad	5	Java Script Object Notation

- محتوای نوع Html و Text توسط تمامی انواع امضا (PDF, CMS attach, CMS detach, ESF, RSA)، قابل امضا کردن هستند.
- محتوای نوع Pdf فقط قابلیت امضای PDF دارد.
- محتوای نوع Json فقط قابلیت امضای CMS به روش detach دارد.
- محتوای نوع Chakad فقط قابلیت امضای CMS به روش attach دارد.

Directions		
LTR	ltr	Left to right
RTL	rtl	Right to left
Unset	unset	Unset

HandwriteSignature		
None	0	No signature image
Content	1	The signature image is received from ContentImage
Optional	2	The signer can add the signature image
Required	3	The signer must add the signature image

ImageType	
APNG	image/apng
GIF	image/gif
JPEG	image/jpeg
JPG	image/jpg
PNG	image/png
TIF	image/tif
TIFF	image/tiff

NotificationStatus		
Ready	0	Ready to send
Sent	1	Sent
Success	2	Accepted by delivery service
Failed	3	Not accepted by delivery service

NotificationType		
Google	0	Google token
Apple	1	Apple token
Local	2	Mobile number, National ID, Email
QueueOnly	3	Create a signature queue without push-notification.

SignType		
Unknown	0	Unknown
PDFSign	1	PDF Sign PKCS#1
CMSSign	2	CMS Detached Sign PKCS#7
ESFSign	3	ESF Sign PKCS#1
RSASign	4	RSA Sign PKCS#1
CMSAttachedSign	5	CMS Attached Sign PKCS#7

PriorityType		
Auto	1	System Set order.
Manual	2	The beneficial set order.

LegalSignatureStatus		
NotExist	0	Not exist.
WattingForOrder	1	Wait for order.
WattingForSign	2	Wait for sign.
Signed	3	Signed.
Failed	4	Failed.
Expired	5	Expired.

Error Codes		
InvalidCallerIP	1	نشانی فراخواننده سرویس نامعتبر است
InvalidUsernameOrPassword	2	نام کاربری یا گذرواژه نامعتبر است
UserIsNotEnable	3	کاربر غیر فعال است
TokenTimeOut	4	توکن ارسالی منقضی شده است
Exception	5	خطای سیستمی
TokenIsUsed	6	محتوای درخواستی پیش از این امضا شده است
TokenNotFound	7	توکن ارسالی پیدا نشد
InvalidRequestOutputType	8	نوع تعریف شده برای خروجی امضا نامعتبر است
DastineError	9	خطای دستینه
NotImplementedException	10	این قابلیت هنوز پیاده سازی نشده است
SignatureNotVerified	11	امضا معتبر نیست
InvalidArchiveTokenId	12	توکن درخواست شده نامعتبر است
ValidationFailed	13	اطلاعات توکن نامعتبر است
InvalidSignType	14	نوع امضا نامعتبر است
InvalidFileType	15	فایل‌های {۰} پشتیبانی نمی‌شوند
InvalidPosition	16	موقعیت تصویر امضا نادرست است
NotSignedYet	17	محتوای مرتبط هنوز توسط کاربر امضا نشده است.
InvalidCallbackType	18	نوع فراخوانی نشانی callback معتبر نیست.
InvalidCallbackUrl	19	نشانی callback معتبر نیست.
NotificationTokenNotFound	20	توکن پیام رسانی پیدا نشد.
UserIsTampered	21	حساب کاربری دستکاری شده است.
CallbackServerNotAvailable	22	سرور دریافت کننده‌ی نتیجه امضا در دسترس نیست.
ValidityDurationOutOfRange	23	مدت اعتبار توکن بیشتر از ۱۰ روز (۱۴۴۰۰ دقیقه) پذیرفته نمی‌شود. (این مدت، توسط مدیر وب‌سرور قابل تغییر است)
ContentExpired	24	مدت اعتبار محتوای قابل امضا منقضی شده است.
AndroidNotificationAccountNotExist	25	حساب پیام رسانی Android وجود ندارد.
IOSNotificationAccountNotExist	26	حساب پیام رسانی iOS وجود ندارد.
InvalidTemplateName	27	قالب نمایش تحت وب، با نام درخواستی وجود ندارد.
AccessToSignedContentIsNotAllowed	28	دسترسی به محتوای امضا شده مجاز نیست.
NullContent	29	محتوای ارسالی تهی است.
TemplateNotFound	30	قالب محتوا پیدا نشد.
NotificationAccountNotFound	31	حساب پیام رسانی پیدا نشد.
NotificationOwnerNotFound	32	نام نمایشی ارسال کننده پیام پیدا نشد.

NotificationTokenNull	33	توکن پیام رسانی مقدار ندارد .
SignerNotFound	34	مشخصات امضاکننده پیدا نشد .
SignatureNotFound	35	اطلاعات امضا پیدا نشد .
InvalidContentType	36	نوع محتوا معتبر نیست .
NotificationRequestNotFound	37	اطلاعات درخواست پیام رسانی پیدا نشد .
InvalidRequestedContent	38	محتوای درخواستی نامعتبر است .
ContentNotReceived	39	محتوایی جهت امضا ارسال نشده است .
JsonToPdfNotSupported	40	از تبدیل json به pdf پشتیبانی نمی‌شود .
ISGNotificationNotConfigured	41	درگاه امضا برای ارسال پیام پیکربندی نشده است .
InvalidOwner	42	توکن و محتوای مرتبط با آن متعلق به کاربر دیگری است .
RequiredTitle	43	عنوان محتوا الزامی است.
RequiredJsonTemplateStructure	44	برای محتوای نوع json/Chakad ساختار الگو الزامی است.
InvalidHandwriteSignature	45	نوع تصویر امضا معتبر نیست.
InvalidMaxLenDataToSign	46	حجم محتوای ارسال شده برای امضا معتبر نیست.
InvalidDataToSign	47	محتوای ارسال شده برای امضا معتبر نیست.
InvalidOCSP	48	گواهینامه امضای شما از سوی سامانه OCSP نامعتبر اعلام شده است.
NoCertificate	49	گواهینامه امضا کننده وجود ندارد.
UnavailableDSSService	50	سرویس DSS در دسترس نیست .
InvalidKeyUsages	51	کاربرد گواهی نامعتبر است.
InvalidAuthenticationAttempt	70	تلاش برای احراز هویت نامعتبر است .
AuthenticationLastChance	71	احراز هویت نامعتبر است. فقط یک بار دیگر می‌توانید تلاش کنید .
AccountLockedOut	72	حساب کاربری قفل شده است. با مدیر سیستم تماس بگیرید .
NotAllowedToLogin	73	شما مجاز به ورود به سیستم نیستید .
ChangePasswordError	74	تغییر گذرواژه با خطا مواجه شد .
DisableUserError	75	غیر فعال کردن کاربر با خطا مواجه شد .
RequiredUserName	76	نام کاربری الزامی است .
RequiredDisplayName	77	نام نمایشی الزامی است .
RequiredValidIP	78	دست کم یک نشانی آی پی الزامی است .
RequiredPassword	79	گذرواژه الزامی است .
MatchConfirmPassword	80	گذرواژه و تایید گذرواژه یکسان نیستند .
UserRegistrationException	81	خطا در هنگام ایجاد حساب کاربری
UserNotFound	82	کاربر مورد نظر پیدا نشد .

AccessTokenExpired	83	توکن دسترسی منقضی شده است .
RefreshTokenExpired	84	توکن بازسازی دسترسی منقضی شده است .
InvalidAccessToken	85	توکن دسترسی معتبر نیست .
InvalidRefreshToken	86	توکن بازسازی معتبر نیست .
InvalidRequestedToken	87	توکن درخواستی معتبر نیست .
InvalidPayload	88	اطلاعات احراز هویت معتبر نیست .
AccessViolation	89	دسترسی مجاز نیست.
WrongPassword	90	رمز عبور معتبر نیست.
RequiredLogo	91	آرم سازمانی الزامی است.
NoSignTools	92	ابزار امضا انتخاب نشده است.
NoRecord	93	رکورد موردنظر یافت نشد
OperationFailed	94	عملیات انجام نشد.
NoExistTemplate	95	این الگو وجود ندارد.
SendNotifError	96	ارسال نوتیفیکیشن با خطا مواجه شد، اما درخواست در صف امضا قرار داده شد.
UnhandledNotificationException	97	ارسال نوتیفیکیشن با خطا مواجه شد، نوتیفیکیشن مجددا ارسال گردد.
UnhandledException	100	یک خطای مدیریت نشده رخ داده است. لطفا با مدیر سیستم تماس بگیرید .
NoSubjectDN	101	کد ملی شخص ارسال نشده است.
ConfilictJsonHtml	102	اطلاعات ارسال شده برای امضا معتبر نیست.
JsonHtmlParameter	103	پارامتر
JsonHtmlParameterNotExist	104	در ساختار ارسال شده وجود ندارد.