



Manual de Integración

Viafirma Inbox

Tabla de contenido

Introducción	1.1
Servicio REST	1.2
Seguridad	1.2.1
Notificaciones	1.2.2
Métodos	1.2.3
Crear un usuario	1.2.3.1
Actualizar un usuario	1.2.3.2
Obtener datos de un usuario	1.2.3.3
Desactivar un usuario	1.2.3.4
Activar un usuario	1.2.3.5
Buscar usuarios	1.2.3.6
Crear un borrador	1.2.3.7
Obtener el enlace de un borrador	1.2.3.8
Crear una petición	1.2.3.9
Obtener datos de una petición	1.2.3.10
Eliminar una petición	1.2.3.11
Finalizar una petición	1.2.3.12
Añadir un comentario	1.2.3.13
Añadir un anexo	1.2.3.14
Descargar documento	1.2.3.15
Descargar documento firmado	1.2.3.16
Listado de cargos	1.2.3.17
Listado de grupos	1.2.3.18
Listado de flujos públicos	1.2.3.19
Obtener datos de un flujo	1.2.3.20
Listado de tipos de rechazo	1.2.3.21
Listado de entidades	1.2.3.22
Crear o modificar una entidad	1.2.3.23
Obtener datos de una entidad	1.2.3.24
Listado de metadatos	1.2.3.25
Listado de tipos de acceso a la verificación	1.2.3.26
Estadísticas	1.2.4
Usuarios creados	1.2.4.1
Usuarios autenticados	1.2.4.2
Peticiónes creadas	1.2.4.3
Documentos creados	1.2.4.4
Documentos firmados	1.2.4.5
Documentos firmados con viafirma platform	1.2.4.5.1
Documentos firmados con viafirma fortress	1.2.4.5.2

Documentos firmados con OTP SMS	1.2.4.5.3
Firmas	1.2.4.6
Firmas con viafirma platform	1.2.4.6.1
Firmas con viafirma fortress	1.2.4.6.2
Firmas con OTP SMS	1.2.4.6.3
Enumerados	1.2.5
Timezones	1.2.5.1
Excepciones	1.2.6
Servicio SOAP	1.3
Seguridad	1.3.1
Notificaciones	1.3.2
Funcionalidades disponibles	1.3.3
Modelo de dominio	1.3.4
AccessTypeDTO	1.3.4.1
AccessTypesResponseDTO	1.3.4.2
ActionTypeDTO	1.3.4.3
ActionTypesResponseDTO	1.3.4.4
AddAnnexesResponseDTO	1.3.4.5
AddCommentResponseDTO	1.3.4.6
ComposeRequestResponseDTO	1.3.4.7
DocumentDTO	1.3.4.8
GroupDTO	1.3.4.9
GroupsResponseDTO	1.3.4.10
JobDTO	1.3.4.11
JobsResponseDTO	1.3.4.12
MetadataDTO	1.3.4.13
MetadataResponseDTO	1.3.4.14
PingResponseDTO	1.3.4.15
RejectTypeDTO	1.3.4.16
RejectTypesResponseDTO	1.3.4.17
RequestAddresseeDTO	1.3.4.18
RequestAnnexesResponseDTO	1.3.4.19
RequestCommentDTO	1.3.4.20
RequestCommentsResponseDTO	1.3.4.21
RequestDocumentsResponseDTO	1.3.4.22
RequestDTO	1.3.4.23
RequestInfoResponseDTO	1.3.4.24
RequestMetadataDTO	1.3.4.25
RequestMetadataResponseDTO	1.3.4.26
RequestStatusResponseDTO	1.3.4.27
RequestUserDTO	1.3.4.28
SendRequestResponseDTO	1.3.4.29

SignatureDocumentDTO	1.3.4.30
SignaturePositionDTO	1.3.4.31
SignatureTicketResponseDTO	1.3.4.32
SignedDocumentResponseDTO	1.3.4.33
SignRequestResponseDTO	1.3.4.34
StatusDTO	1.3.4.35
StatusResponseDTO	1.3.4.36
TemplateDTO	1.3.4.37
TemplateFieldDTO	1.3.4.38
UserDTO	1.3.4.39
UserResponseDTO	1.3.4.40
UsersResponseDTO	1.3.4.41
WorkflowDTO	1.3.4.42
WorkflowRequestResponseDTO	1.3.4.43
WorkflowsResponseDTO	1.3.4.44
WorkflowUsersDTO	1.3.4.45
SignersApprovalsDataDTO	1.3.4.46
SignersApprovalsDataResponseDTO	1.3.4.47
WSInformationDTO	1.3.4.48
Enumerados	1.3.5
ResponseCode	1.3.5.1
NotificationLevel	1.3.5.2
AuthSignType	1.3.5.3
Descripción del WSDL	1.3.6
Códigos de error	1.3.6.1
addAnnexes	1.3.6.2
addComment	1.3.6.3
deleteRequest	1.3.6.4
endRequest	1.3.6.5
evaluateGlobalWorkflow	1.3.6.6
getAccessTypes	1.3.6.7
getActionTypes	1.3.6.8
getGroups	1.3.6.9
getJobs	1.3.6.10
getMetadataKeys	1.3.6.11
getRejectTypes	1.3.6.12
getRequestAnnexes	1.3.6.13
getRequestComments	1.3.6.14
getRequestDocuments	1.3.6.15
getRequestInfo	1.3.6.16
getRequestMetadata	1.3.6.17
getRequestStatus	1.3.6.18

getSignatureTicket	1.3.6.19
getSignedDocument	1.3.6.20
getStatus	1.3.6.21
getUserByPersonId	1.3.6.22
getUsers	1.3.6.23
getUsersBySearchTerm	1.3.6.24
getWorkflows	1.3.6.25
ping	1.3.6.26
prepareComposeRequest	1.3.6.27
prepareComposeRequestAdvanced	1.3.6.28
prepareSignRequest	1.3.6.29
prepareSignRequestAdvanced	1.3.6.30
prepareWorkflowRequest	1.3.6.31
prepareWorkflowRequestAdvanced	1.3.6.32
sendRequest	1.3.6.33
sendRequestWithStrWorkflow	1.3.6.34
sendRequestWithTemplate	1.3.6.35
getSignersApprovalsData	1.3.6.36
Ejemplos de integración	1.3.7
Aplicación de ejemplo en Java	1.3.7.1
Aplicación de ejemplo en .NET	1.3.7.2

Manual de integración de viafirma inbox

El presente documento pretende ser una guía de integración mediante servicios web para el proyecto **Viafirma Inbox**.

Existen dos posibilidades de integración:

- [Servicio REST](#)
- [Servicio SOAP](#)

Si lo desea puede descargar este manual en pdf

- [Castellano](#).
- [Catalán](#).

Servicio REST

Desde la versión 3.0.0, viafirma inbox proporciona un servicio rest para la integración de sistemas externos con viafirma inbox.

Este servicio nos permite realizar las siguientes acciones:

- [Crear un usuario](#)
- [Actualizar un usuario](#)
- [Obtener datos de un usuario](#)
- [Desactivar un usuario](#)
- [Activar un usuario](#)
- [Buscar usuarios](#)
- [Crear un borrador](#)
- [Obtener el enlace de un borrador](#)
- [Crear una petición](#)
- [Obtener datos de una petición](#)
- [Eliminar una petición](#)
- [Finalizar una petición](#)
- [Añadir un comentario](#)
- [Añadir un anexo](#)
- [Descargar documento](#)
- [Descargar documento firmado](#)
- [Listado de cargos](#)
- [Listado de grupos](#)
- [Listado de flujos públicos](#)
- [Obtener datos de un flujo](#)
- [Listado de tipos de rechazo](#)
- [Listado de entidades](#)
- [Crear o modificar entidad](#)
- [Obtener datos de una entidad](#)
- [Listado de metadatos](#)
- [Listado de tipos de acceso a la verificación](#)

Seguridad

Los servicios web de viafirma inbox están protegidos por usuario y contraseña, que deben ser proporcionados por un usuario administrador de la instancia de viafirma inbox. Estos usuarios son los encargados del [mantenimiento de las credenciales de acceso](#).

El servicio rest de viafirma inbox esta securizado empleando una seguridad básica.

El servicio soap de viafirma inbox requiere la identificación del solicitante con usuario y password haciendo uso de Jaxws, según la especificación Web Services Security UsernameToken Profile 1.0.

Notificaciones

Las aplicaciones terceras que utilicen los servicios web de Viafirma Inbox podrán, por ejemplo, crear peticiones, consultar su estado, etc. Sin embargo, en algunas ocasiones no basta con que estas aplicaciones puedan consultar el estado de una petición, sino que se requiere que el sistema les notifique cualquier novedad sobre estas peticiones creadas desde el aplicativo externo.

Esto está permitido y contemplado en el API de la plataforma. Al hora de enviar una nueva petición al sistema, entre los datos que se solicita (si bien no es obligatorio) está la URL a la cual Viafirma Inbox enviará las notificaciones de cambios sobre la petición creada. De este modo, cuando una notificación sea modificada por otro usuario (firmada, rechazada o dado el visto bueno), Viafirma Inbox hará un GET HTTP sobre la url proporcionada a la que se le añadirán los siguientes campos en el query string de la operación:

- **action:** acción realizada sobre la petición. Los posibles valores son:
 - NUEVO: Un borrador creado desde los servicios web ha sido enviado.
 - FIRMADO: Un usuario ha firmado la petición.
 - VISTOBUENO: Un usuario ha dado un visto bueno a la petición.
 - DEVUELTO: Un usuario ha rechazado la petición.
 - ELIMINADO: Un usuario ha eliminado la petición.
- **label:** Identificador (Public access id) de la petición que ha sido modificada (este parámetro se devuelve entre otros cuando se crea una petición a través del API, e identifica de forma unívoca una petición).
- **finished:** Este parámetro se enviará sólo en el caso de que la petición de la que se recibe la notificación ha finalizado. Su valor será siempre el mismo: "ok".

Métodos

Este servicio nos permite realizar las siguientes acciones:

- [Crear un usuario](#)
- [Actualizar un usuario](#)
- [Obtener datos de un usuario](#)
- [Desactivar un usuario](#)
- [Activar un usuario](#)
- [Buscar usuarios](#)
- [Crear un borrador](#)
- [Obtener el enlace de un borrador](#)
- [Crear una petición](#)
- [Obtener datos de una petición](#)
- [Eliminar una petición](#)
- [Finalizar una petición](#)
- [Añadir un comentario](#)
- [Añadir un anexo](#)
- [Descargar documento](#)
- [Descargar documento firmado](#)
- [Listado de cargos](#)
- [Listado de grupos](#)
- [Listado de flujos públicos](#)
- [Obtener datos de un flujo](#)
- [Listado de tipos de rechazo](#)
- [Listado de entidades](#)
- [Crear o modificar entidad](#)
- [Obtener datos de una entidad](#)
- [Listado de metadatos](#)
- [Listado de tipos de acceso a la verificación](#)

Crear un usuario

Método para la creación de un nuevo usuario. Los usuarios creados por este método tendrán el rol de **Usuario**. Puede consultar la funcionalidad asociada a cada rol en el [manual de administración](#).

- Tipo: **POST**
- URL: **url-inbox/api/v3/users**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **userCode**: Código (Obligatorio)
- **universalCode**: Usuario del directorio activo.
- **name**: Nombre (Obligatorio)
- **surname1**: Primer apellido (Obligatorio)
- **surname2**: Segundo apellido
- **phone**: Teléfono
- **defaultStamp**: Indica el sello de firma por defecto que tendrá seleccionado el usuario al redactar peticiones. Si este campo no es indicado, el usuario tomará el sello de firma por defecto de la instancia. Los posibles valores que puede tomar este campo son:
 - **INSTANCE_DEFAULT_STAMP**: por defecto, el usuario tendrá seleccionado al redactar el sello de firma indicado por defecto en la instancia.
 - **NO_STAMP**: por defecto, el usuario no tendrá seleccionado ningún sello de firma al redactar.
 - **[CÓDIGO DEL SELLO]**: por defecto, el usuario tendrá seleccionado al redactar el sello cuyo código corresponda con el indicado. Por ejemplo, indicando esto **"defaultStamp": "ASDFGHJKLZ"** el usuario tendrá asignado por defecto el sello de firma cuyo código sea igual a **ASDFGHJKLZ**.
- **entities**: Listado de entidades del usuario. Para cada entidad se muestra:
 - **entityCode**: Código de la entidad.(Obligatorio)
 - **email**: Correo electrónico del usuario asociado a la entidad.(Obligatorio)
 - **isDefault**: Indica si es la entidad por defecto del usuario.(Obligatorio)
 - **jobs**: Listados de cargos del usuario asociados a la entidad. Se debe indicar el código del cargo. Se pueden consultar los cargos disponibles con el servicio [listado de cargos](#).
 - **groups**: Listados de grupos del usuario asociados a la entidad. Se debe indicar el código del grupo. Se pueden consultar los grupos disponibles con el servicio [listado de grupos](#).
- **cmis**: Información del repositorio personal de CMIS. Para instancias que tienen activo un repositorio de CMIS es posible indicar el repositorio personal de un usuario. De forma que el usuario podrá añadir documentos desde este repositorio en la redacción de peticiones. Los datos que se muestran son:
 - **pathbase**: Path base del repositorio del usuario.
 - **folderId**: Identificador de la carpeta del usuario.
 - **user**: Usuario del repositorio CMIS.
 - **password**: Contraseña del repositorio CMIS.
- **timezone**: Zona horaria del usuario. Puede consultar los valores permitidos [aquí](#). Valor por defecto EUROPE_MADRID.
- **locale**: Locale del usuario. Posibles valores:
 - **ES**: Castellano
 - **EN**: Inglés
 - **FR**: Francés
 - **CA**: Catalán
 - **EU**: Euskera

Valor por defecto ES.

- **notificationsLevel**: Nivel de notificación del usuarios. Posibles valores:
 - **HIGH**: Se reciben varios correos por petición (para cada lectura, firma o visto bueno).
 - **MEDIUM**: No se envían los emails de lectura.
 - **LOW**: Solamente se envían emails de aviso cuando una petición en espera pasa a estar disponible para realizar la acción.
 - **NEWSLETTER**: Se recibe un boletín con la periodicidad seleccionada, con todas las peticiones recibidas en este periodo (lecturas, firmas o vistos bueno).
- **newsletterFrequencyDays**: Para usuario con nivel de notificación boletín. Indica la periodicidad del boletín en días.
- **isSender**: Indica si el usuario tiene permisos para el envío de peticiones. Por defecto true.
- **canSendAllEntity**: Para usuarios con permisos de envío de peticiones indica si el usuario puede enviar únicamente a usuarios de sus entidades o a usuarios de todas las entidades. Por defecto true.
- **canDelegate**: Indica si el usuario puede delegar su firma/visto bueno.
- **canViewWorkflow**: Indica si el usuario puede ver el flujo de firmas de las peticiones en las que participa.
- **isServerSign**: Indica si es un usuario de firma automatizada.
- **serverSignAlias**: Alias del certificado del usuario de firma automatizada.
- **serverSignPassword**: Contraseña del certificado del usuario de firma automatizada.
- **numberIds**: Identificadores de los certificados con los que el usuario puede realizar firmas.
- **isExternalSign**: Indica si el usuario es externo o interno.
- **defaultExternalSign**: En caso de tratarse de un usuario externo indica el tipo de firma por defecto para dicho usuario. Posibles valores: PLATFORM,FORTRESS,COSIGN,OTPSMS,WEB.

[Ejemplo de json de entrada](#)

Json de salida

El json de salida será el mismo que el del servicio [obtener datos de un usuario](#)

[Ejemplo de json de salida](#)

Actualizar un usuario

Método para la actualización de un usuario. Si el usuario existe y su rol no es de tipo administrador lo actualiza. Puede consultar la funcionalidad asociada a cada rol en el [manual de administración](#).

- Tipo: **PUT**
- URL: **url-inbox/api/v3/users/update**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **userCode**: Código (Obligatorio)
- **universalCode**: Usuario del directorio activo.
- **name**: Nombre (Obligatorio)
- **surname1**: Primer apellido (Obligatorio)
- **surname2**: Segundo apellido
- **phone**: Teléfono
- **defaultStamp**: Indica el sello de firma por defecto que tendrá seleccionado el usuario al redactar peticiones. Si este campo no es indicado, el usuario tomará el sello de firma por defecto de la instancia. Los posibles valores que puede tomar este campo son:
 - **INSTANCE_DEFAULT_STAMP**: por defecto, el usuario tendrá seleccionado al redactar el sello de firma indicado por defecto en la instancia.
 - **NO_STAMP**: por defecto, el usuario no tendrá seleccionado ningún sello de firma al redactar.
 - **[CÓDIGO DEL SELLO]**: por defecto, el usuario tendrá seleccionado al redactar el sello cuyo código corresponda con el indicado. Por ejemplo, indicando esto **"defaultStamp": "ASDFGHJKLZ"** el usuario tendrá asignado por defecto el sello de firma cuyo código sea igual a **ASDFGHJKLZ**.
- **entities**: Listado de entidades del usuario. Para cada entidad se muestra:
 - **entityCode**: Código de la entidad.(Obligatorio)
 - **email**: Correo electrónico del usuario asociado a la entidad.(Obligatorio)
 - **isDefault**: Indica si es la entidad por defecto del usuario.(Obligatorio)
 - **jobs**: Listados de cargos del usuario asociados a la entidad. Se debe indicar el código del cargo. Se pueden consultar los cargos disponibles con el servicio [listado de cargos](#).
 - **groups**: Listados de grupos del usuario asociados a la entidad. Se debe indicar el código del grupo. Se pueden consultar los grupos disponibles con el servicio [listado de grupos](#).
- **cmis**: Información del repositorio personal de CMIS. Para instancias que tienen activo un repositorio de CMIS es posible indicar el repositorio personal de un usuario. De forma que el usuario podrá añadir documentos desde este repositorio en la redacción de peticiones. Los datos que se muestran son:
 - **pathbase**: Path base del repositorio del usuario.
 - **folderId**: Identificador de la carpeta del usuario.
 - **user**: Usuario del repositorio CMIS.
 - **password**: Contraseña del repositorio CMIS.
- **timezone**: Zona horaria del usuario. Puede consultar los valores permitidos [aquí](#). Valor por defecto EUROPE_MADRID.
- **locale**: Locale del usuario. Posibles valores:
 - **ES**: Castellano
 - **EN**: Inglés
 - **FR**: Francés
 - **CA**: Catalán
 - **EU**: Euskera

Valor por defecto ES.

- **notificationsLevel**: Nivel de notificación del usuarios. Posibles valores:
 - **HIGH**: Se reciben varios correos por petición (para cada lectura, firma o visto bueno).
 - **MEDIUM**: No se envían los emails de lectura.
 - **LOW**: Solamente se envían emails de aviso cuando una petición en espera pasa a estar disponible para realizar la acción.
 - **NEWSLETTER**: Se recibe un boletín con la periodicidad seleccionada, con todas las peticiones recibidas en este periodo (lecturas, firmas o vistos bueno).
- **newsletterFrequencyDays**: Para usuario con nivel de notificación boletín. Indica la periodicidad del boletín en días.
- **isSender**: Indica si el usuario tiene permisos para el envío de peticiones. Por defecto true.
- **canSendAllEntity**: Para usuarios con permisos de envío de peticiones indica si el usuario puede enviar únicamente a usuarios de sus entidades o a usuarios de todas las entidades. Por defecto true.
- **canDelegate**: Indica si el usuario puede delegar su firma/visto bueno.
- **canViewWorkflow**: Indica si el usuario puede ver el flujo de firmas de las peticiones en las que participa.
- **isServerSign**: Indica si es un usuario de firma automatizada.
- **serverSignAlias**: Alias del certificado del usuario de firma automatizada.
- **serverSignPassword**: Contraseña del certificado del usuario de firma automatizada.
- **numberIds**: Identificadores de los certificados con los que el usuario puede realizar firmas.

[Ejemplo de json de entrada](#)

Json de salida

El json de salida será el mismo que el del servicio [obtener datos de un usuario](#)

[Ejemplo de json de salida](#)

Obtener datos de un usuario

Método para la obtener los datos de un usuario a partir de su código.

- Tipo: **GET**
- URL: **url-inbox/api/v3/users/{userCode}**
- Tipo de salida: **application/json**
- Parámetros:
 - **userCode**: código de usuario.

Json de salida

El json de salida está compuesto por los siguientes campos:

- **userCode**: Código (Obligatorio)
- **universalCode**: Usuario del directorio activo.
- **name**: Nombre (Obligatorio)
- **surname1**: Primer apellido (Obligatorio)
- **surname2**: Segundo apellido
- **role**: Perfil del usuario en la aplicación. Los posibles valores son: USER, ADMIN_FUNCTIONAL, ADMIN_INFRASTRUCTURE, ADMIN_GLOBAL.
- **phone**: Teléfono
- **defaultStamp**: Sello de firma por defecto que tendrá seleccionado el usuario al redactar peticiones. Los posibles valores son: INSTANCE_DEFAULT_STAMP, NO_STAMP o el código de un sello específico.
- **Listado de entidades del usuario**: Para cada entidad se muestra:
 - **entityCode**: Código de la entidad.(Obligatorio)
 - **email**: Correo electrónico del usuario asociado a la entidad.(Obligatorio)
 - **isDefault**: Indica si es la entidad por defecto del usuario.(Obligatorio)
 - **jobs**: Listados de cargos del usuario asociados a la entidad. Se debe indicar el código del cargo. Se pueden consultar los cargos disponibles con el servicio [listado de cargos](#).
 - **groups**: Listados de grupos del usuario asociados a la entidad. Se debe indicar el código del grupo. Se pueden consultar los grupos disponibles con el servicio [listado de grupos](#).
- **Información del repositorio personal de CMIS**: Para instancias que tienen activo un repositorio de CMIS es posible indicar el repositorio personal de un usuario. De forma que el usuario podrá añadir documentos desde este repositorio en la redacción de peticiones. Los datos que se muestran son:
 - **pathbase**: Path base del repositorio del usuario.
 - **folderId**: Identificador de la carpeta del usuario.
 - **user**: Usuario del repositorio CMIS.
 - **password**: Contraseña del repositorio CMIS.
- **timezone**: Zona horaria del usuario. Puede consultar los valores permitidos [aquí](#).
- **locale**: Locale del usuario. Posibles valores: ES, EN, FR, CA, EU.
- **notificationsLevel**: Nivel de notificación del usuarios. Posibles valores:
 - **HIGH**: Se reciben varios correos por petición (para cada lectura, firma o visto bueno).
 - **MEDIUM**: No se envían los emails de lectura.
 - **LOW**: Solamente se envían emails de aviso cuando una petición en espera pasa a estar disponible para realizar la acción.
 - **NEWSLETTER**: Se recibe un boletín con la periodicidad seleccionada, con todas las peticiones recibidas en este periodo (lecturas, firmas o vistos bueno).
- **newsletterFrequencyDays**: Para usuario con nivel de notificación boletín. Indica la periodicidad del boletín en días.
- **isSender**: Indica si el usuario tiene permisos para el envío de peticiones. Por defecto true.
- **canSendAllEntity**: Para usuarios con permisos de envío de peticiones indica si el usuario puede enviar únicamente a usuarios de sus entidades o a usuarios de todas las entidades. Por defecto true.

- **canDelegate**:
- **canViewWorkflow**: Indica si el usuario puede ver el flujo de firmas de las peticiones en las que participa.
- **isServerSign**: Indica si es un usuario de firma automatizada.
- **serverSignAlias**: Alias del certificado del usuario de firma automatizada.
- **serverSignPassword**: Contraseña del certificado del usuario de firma automatizada.
- **isExternalSign**: Indica si el usuario es externo o interno
- **defaultExternalSign**: En caso de tratarse de un usuario externo indica el tipo de firma por defecto para dicho usuario. Posibles valores: PLATFORM,FORTRESS,COSIGN,OTPSMS,WEB.
- **numberIds**: Identificadores de los certificados con los que el usuario puede realizar firmas.
- **isActive**: Indica si el usuario está activo.
- **delegationsTo** : Delegaciones otorgadas por otros usuarios al usuario. Los datos que se muestran son:
 - **userCodeTo**: Código del usuario que recibe la delegación.
 - **userCodeFrom**: Código del usuario que otorga la delegación.
 - **status**: Estado de la delegación. Posibles valores:
 - **PENDING**: Pendiente de firma por el usuario que delega.
 - **SIGNED_FROM**: Firmado por el usuario que delega, pero no por el usuario que recibe la delegación.
 - **ACTIVE**: La delegación está activa.
 - **EXPIRED**: La delegación ha caducado.
 - **permissions**: Permisos del usuario. Los valores posibles son:
 - **SIGN**: Firma, visto bueno y rechazo.
 - **APPROVAL**: Visto bueno y rechazo.
 - **COLLABORATOR**: Solo consulta de peticiones.
 - **isDeleted**: La delegación ha sido eliminada.
 - **dateFrom**: Fecha de inicio de la delegación.
 - **dateTo**: Fecha de fin de la delegación.
- **delegationsFrom**: Delegaciones otorgadas por el usuario. Los datos que se muestran son los mismos que en el campo *delegationsTo*.

[Ejemplo de json de salida](#)

Desactivar un usuario

Método para la desactivar un usuario.

- Tipo: **GET**
- URL: **url-inbox/api/v3/users/deactivate/{userCode}**
- Parámetros:
 - **userCode**: código de usuario.

Json de salida

El json de salida será el mismo que el del servicio [obtener datos de un usuario](#)

[Ejemplo de json de salida](#)

Activar un usuario

Método para la activar un usuario.

- Tipo: **GET**
- URL: **url-inbox/api/v3/users/activate/{userCode}**
- Parámetros:
 - **userCode**: código de usuario.

Json de salida

El json de salida será el mismo que el del servicio [obtener datos de un usuario](#)

[Ejemplo de json de salida](#)

Buscar usuarios

Método para la búsqueda de usuarios.

- Tipo: **POST**
- URL: **url-inbox/api/v3/users/search**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**

Json de entrada

El json de entrada está compuesto por los siguientes campos: private String ; private String ;

- **searchTerm**: Texto por el cual se realiza el filtro de los usuarios.
- **entityCode**: Código de la entidad.

Json de salida

Listado de usuarios que contengan el término de búsqueda y la entidad suministrados.

El json de salida será un listado de elementos del mismo tipo que el del servicio [obtener datos de un usuario](#)

[Ejemplo de json de salida](#)

Crear un borrador

Método para la creación de una petición en estado borrador que posteriormente podrá ser completada en viafirma inbox.

- Tipo: **POST**
- URL: **url-inbox/api/v3/drafts**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **request**: [json de entrada del servicio de creación de una nueva petición](#)
- **canChangeSender**: Indica si el usuario redactor puede modificar la entidad desde la que envía en caso de tener más de una entidad asociada. Por defecto true.
- **canChangeAddresseess**: Indica si el usuario redactor puede modificar los destinatarios. Por defecto true.
- **canChangeInternalNotifications**: Indica si el usuario redactor puede modificar los destinatarios de las comunicaciones internas. Por defecto true.
- **canChangeSubject**: Indica si el usuario redactor puede modificar el asunto. Por defecto true.
- **canChangeMessage**: Indica si el usuario redactor puede modificar el mensaje. Por defecto true.
- **canChangeReference**: Indica si el usuario redactor puede modificar la referencia. Por defecto true.
- **canChangeInitDate**: Indica si el usuario redactor puede modificar la fecha de inicio. Por defecto true.
- **canChangeExpirationDate**: Indica si el usuario redactor puede modificar la fecha de caducidad. Por defecto true.
- **canChangeVerificationAccess**: Indica si el usuario redactor puede modificar el tipo de verificación. Por defecto true.
- **canChangeStamp**: Indica si el usuario redactor puede modificar el sello. Por defecto true.
- **canChangeMetadatas**: Indica si el usuario redactor puede modificar los metadatos. Por defecto true.
- **canChangeDocumentsToSign**: Indica si el usuario redactor puede modificar los documentos a firmar. Por defecto true.
- **canChangeAnnexes**: Indica si el usuario redactor puede modificar los anexos. Por defecto true.
- **canChangeSignatureMethods**: Indica si el usuario redactor puede modificar el tipo de firma. Por defecto true.

[Ejemplo de json de entrada](#)

Json de salida

El json de salida será el mismo que el del servicio [obtener el enlace de un borrador](#)

[Ejemplo de json de salida](#)

Obtener el enlace de un borrador

Método para la obtener el enlace para acceder a un borrador. Estos enlaces tienen una validez de 30 minutos.

- Tipo: **GET**
- URL: **url-inbox/api/v3/drafts/link/{publicAccessId}**
- Tipo de salida: **application/json**
- Parámetros:
 - **publicAccessId**: código público de la petición.
 - **draftUrl**: enlace del borrador, con una validez de 30 minutos.

Json de salida

[Ejemplo de json de salida](#)

Crear una petición

Método para la creación de una nueva petición.

- Tipo: **POST**
- URL: **url-inbox/api/v3/requests**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **publicAccessId**: Código de la petición. En caso de no indicarse se autogenerará uno.
- **sender**: Remitente de la petición. En caso de no indicarse se tomará el usuario genérico para el web service como usuario remitente. Los datos a indicar son:
 - **userCode**: Código de usuario.
 - **entityCode**: Código de la entidad del remitente. En caso de no indicarse se tomará la entidad por defecto del usuario.
- **addresseeWorkflow**: Código público de un workflow de usuarios definido en viafirma inbox. Campo obligatorio a no ser que se indique los destinatarios en el campo **addresseeLines**.
- **addresseeLines**: Sólo para peticiones en las que no se ha indicado del campo **addresseeWorkflow**, en cuyo caso es un campo obligatorio. Líneas de los destinatarios de la petición. Cada línea de destinatarios está compuesto por una lista de grupos de destinatarios (**addresseeGroups**). Cada uno de estos grupos tiene los siguientes campos:
 - **isOrGroup**: true si sólo uno de los usuarios del grupo tiene que realizar la acción. Por defecto false.
 - **userEntities**: Lista de usuarios que componen el grupo. Cada uno de estos elementos puede venir dado por tres diferentes criterios: a través de su código de usuario y de entidad, o a partir de un código de cargo o de grupo. Los cargos o grupos indicados se reemplazarán por todos los usuarios que los componen.
 - **userCode**: Person ID del usuario. En caso de usuarios de firma desatendida debe indicarse el alias, si es firma con viafirma platform, o el código del certificado, si es firma con viafirma fortress.
 - **entityCode**: Código de la entidad del destinatario. En caso de no indicarse se tomará la entidad por defecto del usuario.
 - **jobCode**: Código del cargo
 - **groupCode**: Código del grupo
 - **userName**: Nombre del usuario si se trata de un usuario externo que no existe y debe darse de alta en el sistema.
 - **userSurname1**: Primer apellido del usuario si se trata de un usuario externo que no existe y debe darse de alta en el sistema.
 - **userSurname2**: Segundo apellido del usuario si se trata de un usuario externo que no existe y debe darse de alta en el sistema.
 - **userPhone**: Teléfono del usuario si se trata de un usuario externo que no existe y debe darse de alta en el sistema.

También se puede indicar el tipo de acción a realizar en el campo **action** cuyos posibles valores son *SIGN* y *APPROVAL*, siendo *SIGN* el valor por defecto en caso de no indicarse ninguno.

- **internalNotification**: Listado de usuarios que formarán parte de la comunicación interna asociada a la petición. Se indicará el código de usuario de cada uno de estos usuarios.
- **subject**: Asunto de la petición. Campo obligatorio.
- **message**: Mensaje de la petición.
- **reference**: Referencia.
- **initDate**: Fecha de inicio de la petición.

- **expirationDate**: Fecha de caducidad de la petición.
 - **verificationAccess**: Tipo de acceso a la verificación de la petición. Se deben indicar los siguientes campos:
 - **type**: Tipo de acceso. Por defecto ANONYMOUS. Los posibles valores son:
 - **NOTAVAILABLE**: La petición no está accesible desde el apartado de verificación.
 - **ANONYMOUS**: Cualquier usuario tiene acceso a la petición.
 - **USERPASSWORD**: Acceso mediante usuario y contraseña.
 - **CERTIFICATE**: Requiere autenticación con certificado. Y debe tratarse de uno de los usuarios que forman parte de la petición. Es decir que sea el remitente o uno de los destinatarios.
 - **PRIVATE**: Tan sólo el remitente tiene acceso a la verificación.
 - **username**: Usuario para el acceso a la petición para el tipo USERPASSWORD.
 - **password**: Contraseña para el acceso a la petición para el tipo USERPASSWORD.
 - **senderNotificationsLevel**: Nivel de notificaciones para el remitente de la petición. Los valores permitidos son:
 - **NO**: El remitente no recibe notificaciones para esta petición.
 - **ALL**: El remitente recibe notificaciones cada vez que un destinatario lee, firma, da visto bueno o rechaza. También recibirá una notificación cuando finalice la petición.
 - **MEDIUM**: El remitente recibe notificaciones cada vez que un destinatario firma, da visto bueno o rechaza. También recibirá una notificación cuando finalice la petición.
 - **FINISH**: EL remitente recibirá una notificación cuando finalice la petición.
 - **notificationUrl**: URL a la que notificamos tras la finalización de cada acción de firma o visto bueno definida en el proceso de firma.
 - **stampName**: Nombre o código del sello a emplear.
 - **useDefaultStamp**: Indica si se usa el sello por defecto.
 - **metadatas**: Listado de metadatos de la petición. Para cada metadata se indican:
 - **key**
 - **value**
 - **documentsToSign**: Listado de documentos a firmar. Campo obligatorio. Los documentos pueden venir dados de 3 formas distintas, dependiendo de cual se emplee se deberán indicar unos campos u otros.
 - **csv**: Código seguro de verificación personalizado para el documento. En caso de no indicarse se generará uno automáticamente. Se emplea para la generación de la url de verificación dependiendo de la configuración del sello.
 - **verificationUrl**: Url para la verificación del documento. Se emplea dependiendo de la configuración del sello.
- Para documentos generados a partir de plantillas existentes en viafirma inbox debe rellenar el campo **template** que está compuesto por los campos:

- **code**: Código de la plantilla.
- **vars**: Lista de variables para crear la plantilla. Para cada variable se indican los campos:
 - **key**
 - **value**

Para documentos en base 64 se deben indicar los siguientes campos:

- **filename**: Nombre del documento.
- **base64**: Contenido del documento en base64.

Para documentos disponibles en una url externa se deben indicar los siguientes campos:

- **filename**: Nombre del documento.
- **url**: Url para la descarga del documento.

Además se debe indicar las posiciones de las firmas para usuarios externos con firma web o para todo los usuarios si el sello de firma es de tipo sello individual. Para cada usuario, debe indicarse la posición del sello y el usuario al que corresponde. Los campos userCode y entityCode deben coincidir con los indicados en addresseeLines. Las posiciones de los sellos se hacen con una lista llamada stampPositions:

- **stampPositions**: Array de posiciones del sello de firma en el caso de firmas web. Cada posición viene dada por los campos:
 - **page**: Página en la que se ubicará la firma.

- **height**: Altura de la firma.
- **width**: Anchura de la firma.
- **x**: Posición x de la firma.
- **y**: Posición y de la firma.
- **userCode**: Código del usuario firmante.
- **entityCode**: Código de la entidad del usuario firmante.
- **documentsAnnexes**: Listado de documentos anexos. Se rellenan de forma análoga a los documents a firmar.
- **signWithCertificate**: En caso de valer *true* la firma de la petición deberá realizarse usando un sistema de firma con certificado. En caso contrario la petición puede ser firmada con cualquier sistema de firma (deprecado en favor del siguiente parámetro)
- **signatureLevel**: Nivel de firma admitido para la petición. Los valores permitidos son:
 - **ALL**: Cualquier tipo (certificado, OTP SMS, firma web...). Valor por defecto.
 - **CERTIFICATE_OTP_SMS**: Se admite solo certificado y OTP SMS.
 - **CERTIFICATE_ONLY**: Solo firma con certificado digital.
- **autoReminderPeriodicity**: En el caso de que esté activada por configuración la opción "Activar el envío de recordatorios automáticos a los destinatarios", este parámetro permite indicar cada cuántos días se envía por email un recordatorio al firmante que todavía no ha hecho su acción.
- **autoReminderAttempts**: Número máximo de recordatorios que se enviarán al usuario firmante. Si se da un valor a autoReminderPeriodicity debe informarse también éste, ya que no toma un valor por defecto (si solo se informa un valor, no se guarda ninguna configuración de recordatorios para la petición).
- **disableInboxEmailNotifications**: Si se envía este parámetro opcional con valor "true", Inbox no enviará ningún email a los participantes de la petición: creador, destinatarios, comunicaciones internas... por lo que la aplicación cliente será la responsable de realizar las notificaciones por email.

[Ejemplo de json de entrada](#)

Json de salida

El json de salida será el mismo que el del servicio [obtener datos de una petición](#)

[Ejemplo de json de salida](#)

Obtener datos de una petición

Método para la obtener los datos de una petición a partir de su código público.

- Tipo: **GET**
- URL: **url-inbox/api/v3/requests/{publicAccessId}**
- Tipo de salida: **application/json**
- Parámetros:
 - **publicAccessId**: código público de la petición.

Json de salida

El json de salida está compuesto por los siguientes campos:

- **sender**: Remitente de la petición, con los siguientes datos:
 - **userCode**: Código de usuario.
 - **entityCode**: Código de la entidad del remitente.
- **addresseeLines**: Líneas de los destinatarios de la petición. Cada línea de destinatarios está compuesto por una lista de grupos de destinatarios (**addresseeGroups**). Cada uno de estos grupos tiene los siguientes campos:
 - **isOrGroup**: true si sólo uno de los usuarios del grupo tiene que realizar la acción.
 - **userEntities**: Lista de usuarios que componen el grupo. Los datos que se indican son:
 - **userCode**: Código de usuario.
 - **entityCode**: Código de la entidad.
 - **status**: Estado. Posibles valores:
 - **NEW**: El usuario no ha realizado ninguna acción.
 - **READ**: El usuario ha leído la acción.
 - **SIGNED**: El usuario ha firmado la petición.
 - **APPROVAL**: El usuario ha dado visto bueno a la petición.
 - **REJECT**: El usuario ha rechazado la petición.
 - **NO_ACTION**: El usuario no tiene que realizar ninguna acción dado que se encuentra dentro de un grupo de tipo OR donde otro usuario ya ha realizado la acción.
 - **action**: Acción que realiza el usuario. Posibles valores:
 - **SIGN**: Firma.
 - **APPROVAL**: Visto bueno.
 - **actionInfo**: Información de la acción realizada por el usuario. Se muestran los siguientes datos:
 - **userCode**: Código del usuario que ha realizado la acción. En acciones realizadas por usuario con delegación no coincidirá con el código del remitente.
 - **date**: Fecha en la que se realizó la acción.
 - **rejectType**: Tipo de rechazo
 - **rejectReason**: Motivo de rechazo
 - **externalSignUrl**: Url para el acceso del usuario externo a la petición.
- **internalNotification**: Listado de usuarios que formarán parte de la comunicación interna asociada a la petición. Se indicará el código de usuario de cada uno de estos usuarios.
- **subject**: Asunto de la petición.
- **message**: Mensaje de la petición.
- **publicAccessId**: Código público de la petición.
- **reference*: Referencia
- **creationDate**: Fecha de creación de la petición.
- **initDate**: Fecha de inicio de la petición.
- **expirationDate**: Fecha de caducidad de la petición.
- **sendDate**: Fecha de envío de la petición.
- **verificationAccess**: Tipo de acceso a la verificación de la petición. Se deben mostrar los siguientes campos:

- **type**: Tipo de acceso. Por defecto ANONYMOUS. Los posibles valores son:
 - **NOTAVAILABLE**: La petición no está accesible desde el apartado de verificación.
 - **ANONYMOUS**: Cualquier usuario tiene acceso a la petición.
 - **USERPASSWORD**: Acceso mediante usuario y contraseña.
 - **CERTIFICATE**: Requiere autenticación con certificado. Y debe tratarse de uno de los usuarios que forman parte de la petición. Es decir que sea el remitente o uno de los destinatarios.
 - **PRIVATE**: Tan sólo el remitente tiene acceso a la verificación.
- **username**: Usuario para el acceso a la petición para el tipo USERPASSWORD.
- **password**: Contraseña para el acceso a la petición para el tipo USERPASSWORD.
- **senderNotificationLevel**: Nivel de notificaciones para el remitente de la petición. Los valores permitidos son:
 - **NO**: El remitente no recibe notificaciones para esta petición.
 - **ALL**: El remitente recibe notificaciones cada vez que un destinatario lee, firma, da visto bueno o rechaza. También recibirá una notificación cuando finalice la petición.
 - **MEDIUM**: El remitente recibe notificaciones cada vez que un destinatario firma, da visto bueno o rechaza. También recibirá una notificación cuando finalice la petición.
 - **FINISH**: EL remitente recibirá una notificación cuando finalice la petición.
- **notificationUrl**: Url de notificación cuando se finalice la petición.
- **stampName**: Nombre del sello empleado.
- **status**: Estado de la petición. Los posibles valores son:
 - **NOT_STARTED**: Petición no iniciada.
 - **IN_PROCESS**: Petición en curso.
 - **COMPLETED**: Petición finalizada.
 - **EXPIRED**: Petición caducada.
 - **REJECTED**: Petición rechazada.
- **metadatas**: Listado de metadatos de la petición. Para cada metadata se indican:
 - **key**
 - **value**
- **documentsToSign**: Listado de documentos a firmar. Los datos que se muestran para cada documento son:
 - **filename**: Nombre del fichero.
 - **publicAccessId**: Código público del documento.
- **documentsAnnexes**: Listado de documentos anexos. Los datos que se muestran para cada documento son:
 - **filename**: Nombre del fichero.
 - **publicAccessId**: Código público del documento.
- **rejectInfo**: Información relacionada con el rechazo de una petición, en caso de haber sido rechazada.
 - **rejectUser**: Nombre del usuario que ha rechazado la petición.
 - **rejectType**: Tipo de rechazo seleccionado.
 - **rejectReason**: Motivo del rechazo.
 - **rejectDate**: Fecha de rechazo.
 - **autoReminderPeriodicity**: (solo en el caso de que esté activada por configuración la opción "Activar el envío de recordatorios automáticos a los destinatarios"). Cada cuántos días se envía por email un recordatorio al firmante que todavía no ha hecho su acción.
 - **autoReminderAttempts**: Número máximo de recordatorios que se enviarán al usuario firmante.
 - **disableInboxEmailNotifications**: Si es true, Inbox no envía ningún email a los participantes de la petición: creador, destinatarios, comunicaciones internas... por lo que la aplicación cliente será la responsable de realizar las notificaciones por email.

Ejemplo de json de salida

Eliminar una petición

Método para la eliminación de una petición. La eliminación de peticiones sólo está disponible para peticiones sobre las que no se haya realizado ninguna acción de tipo firma, visto bueno o rechazo.

En caso de haberse realizado alguna acción sobre la petición se devolverá un error de tipo `WS_ERROR_CODE_51`.

- Tipo: **GET**
- URL: `url-inbox/api/v3/requests/delete/{publicAccessId}`
- Parámetros:
 - **publicAccessId**: código público de la petición.

Finalizar una petición

Método para la finalización de una petición. Elimina los destinatarios que no hayan realizado acción, y da por finalizada la petición con las acciones de firma o visto bueno realizadas hasta el momento.

En caso de no haberse realizado alguna acción sobre la petición se devolverá un error de tipo `WS_ERROR_CODE_52`.

- Tipo: **GET**
- URL: `url-inbox/api/v3/requests/end/{publicAccessId}`
- Parámetros:
 - **publicAccessId**: código público de la petición.

Añadir un comentario

Método para añadir un comentario a una petición.

- Tipo: **POST**
- URL: **url-inbox/api/v3/requests/{publicAccessId}/comments**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**
- Parámetros:
 - **publicAccessId**: código público de la petición.

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **userCode**: Código del usuario. En caso de no indicarse se tomará el del usuario web service.
- **comment**: Comentario.

[Ejemplo de json de entrada](#)

Json de salida

El json de salida está compuesto por los siguientes campos:

- **userCode**: Código del usuario.
- **entityCode**: Código de la entidad.
- **comment**: Comentario.
- **date**: Fecha del comentario

[Ejemplo de json de salida](#)

Añadir un anexo

Método para añadir un anexo a una petición.

- Tipo: **POST**
- URL: **url-inbox/api/v3/requests/{publicAccessId}/annexes**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**
- Parámetros:
 - **publicAccessId**: código público de la petición.

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **userCode**: Código del usuario. En caso de no indicarse se tomará el del usuario web service.
- **document**: Documento. Los documentos pueden venir dados de 3 formas distintas, dependiendo de cual se emplee se deberán indicar unos campos u otros.

Para documentos generados a partir de plantillas existentes en viafirma inbox debe rellenar el campo **template** que está compuesto por los campos:

- **code**: Código de la plantilla.
- **vars**: Lista de variables para crear la plantilla. Para cada variable se indican los campos:
 - **key**
 - **value**

Para documentos en base 64 se deben indicar los siguientes campos:

- **filename**: Nombre del documento.
- **base64**: Contenido del documento en base64.

Para documentos disponibles en una url externa se deben indicar los siguientes campos:

- **filename**: Nombre del documento.
- **url**: Url para la descarga del documento.

[Ejemplo de json de entrada](#)

Json de salida

El json de salida está compuesto por los siguientes campos:

- **filename**: Nombre del fichero.
- **publicAccessId**: Código público del documento.

[Ejemplo de json de salida](#)

Descargar documento

Método para descargar un documento.

- Tipo: **GET**
- URL: **url-inbox/api/v3/documents/{publicAccessId}**
- Tipo de salida: **application/json**
- Parámetros:
 - **publicAccessId**: código público del documento.

Json de salida

El json de salida está compuesto por los siguientes campos:

- **filename**: nombre del documento con su extensión. Por ejemplo, "documento.pdf".
- **base64**: documento codificado en base64.

Descargar documento firmado

Método para descargar un documento firmado dado el código público de un documento o de una petición. En caso de indicarse el de una petición y esta tenga varios documentos firmados el fichero resultante será un zip que contendrá los distintos ficheros firmados.

- Tipo: **GET**
- URL: **url-inbox/api/v3/documents/{publicAccessId}/signed**
- Tipo de salida: **application/json**
- Parámetros:
 - **publicAccessId**: código público del documento o de la petición.

Json de salida

El json de salida está compuesto por los siguientes campos:

- **filename**: nombre del documento firmado con su extensión. Por ejemplo, "documento.pdf".
- **base64**: documento firmado codificado en base64.

Listado de cargos

Método para la obtener los cargos existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/jobs**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de cargos. Para cada uno de ellos se muestran los siguientes campos:

- **name**: Nombre del cargo
- **publicAccessId**: Código público del cargo

[Ejemplo de json de salida](#)

Listado de grupos

Método para la obtener los grupos existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/groups**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de grupos. Para cada uno de ellos se muestran los siguientes campos:

- **name**: Nombre del grupo
- **publicAccessId**: Código público del grupo

[Ejemplo de json de salida](#)

Listado de flujos públicos

Método para la obtener los flujos existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/workflows**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de flujos. Para cada uno de ellos se muestran los siguientes campos:

- **name**: Nombre del grupo
- **publicAccessId**: Código público del grupo
- **addresseeLines**: Líneas de los destinatarios. Cada línea de destinatarios está compuesto por una lista de grupos de destinatarios (**addresseeGroups**). Cada uno de estos grupos tiene los siguientes campos:
 - **isOrGroup**: true si sólo uno de los usuarios del grupo tiene que realizar la acción.
 - **userEntities**: Lista de usuarios que componen el grupo. Cada uno de estos elementos puede venir dado por tres diferentes criterios: a través de su código de usuario y de entidad, o a partir de un código de cargo o de grupo. Los cargos o grupos indicados se reemplazarán por todos los usuarios que los componen.
 - **userCode**
 - **entityCode**
 - **jobCode**
 - **groupCode**
- **stringFormat**: Destinatarios del flujo en formato texto.

También se puede indicar el tipo de acción a realizar en el campo **action** cuyos posibles valores son *SIGN* y *APPROVAL*, siendo *SIGN* el valor por defecto en caso de no indicarse ninguno.

[Ejemplo de json de salida](#)

Obtener datos de un flujo

Método para la obtener los datos de un flujo a partir de su código público.

- Tipo: **GET**
- URL: **url-inbox/api/v3/workflows/{code}**
- Tipo de salida: **application/json**
- Parámetros:
 - **code**: código del flujo

Json de salida

El json de salida está compuesto por los siguientes campos:

- **name**: Nombre del grupo
- **publicAccessId**: Código público del grupo
- **addresseeLines**: Líneas de los destinatarios. Cada línea de destinatarios está compuesto por una lista de grupos de destinatarios (**addresseeGroups**). Cada uno de estos grupos tiene los siguientes campos:
 - **isOrGroup**: true si sólo uno de los usuarios del grupo tiene que realizar la acción.
 - **userEntities**: Lista de usuarios que componen el grupo. Cada uno de estos elementos puede venir dado por tres diferentes criterios: a través de su código de usuario y de entidad, o a partir de un código de cargo o de grupo. Los cargos o grupos indicados se reemplazarán por todos los usuarios que los componen.
 - **userCode**
 - **entityCode**
 - **jobCode**
 - **groupCode**
- **stringFormat**: Destinatarios del flujo en formato texto.

También se puede indicar el tipo de acción a realizar en el campo **action** cuyos posibles valores son *SIGN* y *APPROVAL*, siendo *SIGN* el valor por defecto en caso de no indicarse ninguno.

[Ejemplo de json de salida](#)

Listado de tipos de rechazo

Método para la obtener los tipo de rechazo existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/rejectTypes**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de tipos de rechazo. Para cada uno de ellos se muestran los siguientes campos:

- **name**: Nombre del tipo de rechazo
- **description**: Descripción del tipo de rechazo

[Ejemplo de json de salida](#)

Listado de entidades

Método para la obtener las entidades existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/entities**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de entidades. Para cada una de ellas se muestran los siguientes campos:

- **code**: Código de la entidad
- **name**: Nombre de la entidad
- **description**: Descripción de la entidad
- **filter**: Filtros para los certificados en firmas realizadas con viafirma platform

[Ejemplo de json de salida](#)

Crear una entidad

Método para la creación de una nueva petición.

- Tipo: **POST**
- URL: **url-inbox/api/v3/entities**
- Tipo de entrada: **application/json**
- Tipo de salida: **application/json**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **code**: Código de la entidad
- **name**: Nombre de la entidad
- **description**: Descripción de la entidad
- **filter**: Filtros para los certificados en firmas realizadas con viafirma platform

[Ejemplo de json de entrada](#)

Json de salida

El json de salida será el mismo que el del servicio [obtener datos de una entidad](#)

[Ejemplo de json de salida](#)

Obtener datos de una entidad

Método para la obtener los datos de una entidad a partir de su código público.

- Tipo: **GET**
- URL: **url-inbox/api/v3/entities/{code}**
- Tipo de salida: **application/json**
- Parámetros:
 - **code**: código de la entidad.

Json de salida

El json de salida está compuesto por los siguientes campos:

- **code**: Código de la entidad
- **name**: Nombre de la entidad
- **description**: Descripción de la entidad
- **filter**: Filtros para los certificados en firmas realizadas con viafirma platform

[Ejemplo de json de salida](#)

Listado de metadatos

Método para la obtener los metadatos existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/metadatos**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de metadatos. Para cada una de ellos se muestran los siguientes campos:

- **key**: Código del metadato
- **defaultValue**: Valor por defecto (En caso de tratarse de un metadato de tipo combo este valor serán los posibles valores separados por |)
- **description**: Descripción
- **isRequired**: Indica si el metadato es obligatorio

[Ejemplo de json de salida](#)

Listado de tipos de acceso a la verificación

Método para la obtener los tipos de acceso a la verificación existentes.

- Tipo: **GET**
- URL: **url-inbox/api/v3/accessTypes**
- Tipo de salida: **application/json**

Json de salida

El json de salida está compuesto por una lista de tipos de acceso en el campo values del json de salida:

- **values:** Tipos de acceso

[Ejemplo de json de salida](#)

Estadísticas

Este servicio nos permite obtener los siguientes datos estadísticos:

- [Usuarios creados](#)
- [Usuarios autenticados](#)
- [Peticiones creadas](#)
- [Documentos creados](#)
- [Documentos firmados](#)
- [Firmas](#)

Usuarios creados

Método para obtener el número de usuarios creados en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/users/new**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de usuarios creados en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de usuarios creados a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de usuarios creados hasta la fecha dada.

Usuarios autenticados

Método para obtener el número de usuarios autenticados en un periodo de tiempo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/users/login**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por el siguiente campo:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse fecha de inicio se devolverá el número total de usuarios autenticados en el sistema.

Peticiones creadas

Método para obtener el número de peticiones creadas en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/requests/new**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.
- **entitycode**: Código de la entidad.

NOTA: En caso de no indicarse el código de la entidad se contarán las peticiones creadas para todas las entidades.

En caso de no indicarse fecha de inicio ni de fin, se devolverá el número total de peticiones creadas en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de peticiones creadas a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de peticiones creadas hasta la fecha dada.

Documentos creados

Método para obtener el número de documentos creados en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/documents/new**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de documentos creados en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de documentos creados a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de documentos creados hasta la fecha dada.

Documentos firmados

Método para obtener el número de documentos firmados en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/documents/signed**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de documentos firmados en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de documentos firmados a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de documentos firmados hasta la fecha dada.

Documentos firmados con viafirma platform

Método para obtener el número de documentos firmados con viafirma platform en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/documents/signed/platform**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de documentos firmados en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de documentos firmados a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de documentos firmados hasta la fecha dada.

Documentos firmados con viafirma fortress

Método para obtener el número de documentos firmados con viafirma fortress en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/documents/signed/fortress**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de documentos firmados en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de documentos firmados a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de documentos firmados hasta la fecha dada.

Documentos firmados con OTP SMS

Método para obtener el número de documentos firmados con OTP SMS en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/documents/signed/otpsms**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de documentos firmados en el sistema. Si se indica sólo la fecha de inicio se devolverá el número de documentos firmados a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de documentos firmados hasta la fecha dada.

Firmas

Método para obtener el número de firmas realizadas en un periodo dado.

- Tipo: **POST**
- URL: **/stats/signs**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de firmas realizadas. Si se indica sólo la fecha de inicio se devolverá el número de firmas realizadas a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de firmas realizadas hasta la fecha dada.

Firmas con viafirma platform

Método para obtener el número de firmas realizadas con viafirma platform en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/signs/platform**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de firmas realizadas. Si se indica sólo la fecha de inicio se devolverá el número de firmas realizadas a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de firmas realizadas hasta la fecha dada.

Firmas con viafirma fortress

Método para obtener el número de firmas realizadas con viafirma fortress en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/signs/fortress**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de firmas realizadas. Si se indica sólo la fecha de inicio se devolverá el número de firmas realizadas a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de firmas realizadas hasta la fecha dada.

Firmas con OTP SMS

Método para obtener el número de firmas realizadas con OTP SMS en un periodo dado.

- Tipo: **POST**
- URL: **url-inbox/api/v3/stats/signs/otpsms**
- Tipo de entrada: **application/json**
- Tipo de salida: **text/plain**

Json de entrada

El json de entrada está compuesto por los siguientes campos:

- **initDate**: Fecha de inicio. Se tomarán las 00:00:00 del día indicado.
- **endDate**: Fecha de fin. Se tomarán las 23:59:59 del día indicado.

NOTA: En caso de no indicarse ni fecha de inicio ni de fin, se devolverá el número total de firmas realizadas. Si se indica sólo la fecha de inicio se devolverá el número de firmas realizadas a partir de la fecha dada. En caso de indicarse únicamente la fecha de fin se indicará el número de firmas realizadas hasta la fecha dada.

Enumerados

Timezones

Valor	Descripción	Zona horaria
KWAJALEIN	Kwajalein	(GMT -12:00) Eniwetok, Kwajalein
PACIFIC_SAMOA	Pacific/Samoa	(GMT -11:00) Midway Island, Samoa
PACIFIC_TAHITI	Pacific/Tahiti	(GMT -10:00) Hawaii, Tahiti, Rarotonga
US_ALASKA	US/Alaska	(GMT -9:00) Alaska
US_PACIFIC	US/Pacific	(GMT -8:00) Pacific Time (US & Canada)
US_MOUNTAIN	US/Mountain	(GMT -7:00) Mountain Time (US & Canada)
US_CENTRAL	US/Central	(GMT -6:00) Central Time (US & Canada), Mexico City
US_EASTERN	US/Eastern	(GMT -5:00) Eastern Time (US & Canada), Bogota, Lima
AMERICA_CARACAS	America/Caracas	(GMT -4:30) Caracas
CANADA_ATLANTIC	Canada/Atlantic	(GMT -4:00) Atlantic Time (Canada), Santiago, Bermuda, Halifax
AMERICA_SANTO_DOMINGO	America/Santo_Domingo	(GMT -4:00) Caribe, Puerto Rico, Santo Domingo, La Paz
CANADA_NEWFOUNDLAND	Canada/Newfoundland	(GMT -3:30) Newfoundland
AMERICA_ARGENTINA_BUENOS_AIRES	America/Argentina/Buenos_Aires	(GMT -3:00) Brazil, Buenos Aires, Georgetown
ATLANTIC_SOUTH_GEORGIA	Atlantic/South_Georgia	(GMT -2:00) Mid-Atlantic
ATLANTIC_AZORES	Atlantic/Azores	(GMT -1:00) Azores, Cape Verde Islands
EUROPE_LONDON	Europe/London	(GMT) Western Europe Time, London, Lisbon, Casablanca, Canary Islands
EUROPE_MADRID	Europe/Madrid	(GMT +1:00) Brussels, Copenhagen, Madrid, Paris
EUROPE_KALININGRAD	Europe/Kaliningrad	(GMT +2:00) Athens, Kaliningrad, Kiev, South Africa, Turkey, Israel
ASIA_BAGHDAD	Asia/Baghdad	(GMT +3:00) Baghdad, Riyadh, Moscow, St. Petersburg
ASIA_TEHRAN	Asia/Tehran	(GMT +3:30) Tehran
ASIA_DUBAI	Asia/Dubai	(GMT +4:00) Abu Dhabi, Dubai, Muscat, Baku, Tbilisi
ASIA_KABUL	Asia/Kabul	(GMT +4:30) Kabul
ASIA_YEKATERINBURG	Asia/Yekaterinburg	(GMT +5:00) Ekaterinburg,

ASIA_CALCUTTA	Asia/Calcutta	(GMT +5:30) Bombay, Calcutta, Madras, New Delhi
ASIA_KATHMANDU	Asia/Kathmandu	(GMT +5:45) Kathmandu
ASIA_ALMATY	Asia/Almaty	(GMT +6:00) Almaty, Dhaka, Colombo
ASIA_BANGKOK	Asia/Bangkok	(GMT +7:00) Bangkok, Hanoi, Jakarta
ASIA_HONG_KONG	Asia/Hong_Kong	(GMT +8:00) Beijing, Perth, Singapore, Hong Kong
ASIA_TOKYO	Asia/Tokyo	(GMT +9:00) Tokyo, Seoul, Osaka, Sapporo, Yakutsk
AUSTRALIA_ADELAIDE	Australia/Adelaide	(GMT +9:30) Adelaide, Darwin
AUSTRALIA_CANBERRA	Australia/Canberra	(GMT +10:00) Eastern Australia, Guam, Vladivostok
ASIA_MAGADAN	Asia/Magadan	(GMT +11:00) Magadan, Solomon Islands, New Caledonia
PACIFIC_AUCKLAND	Pacific/Auckland	(GMT +12:00) Auckland, Wellington, Fiji, Kamchatka");

Excepciones

Todos los métodos del Servicio Rest en caso de producirse un error devuelve un json con los siguientes campos:

- **errorCode**: Código del error (String)
- **errorMessage**: Mensaje de error (String)

[Acceso al listado de códigos de errores](#)

Servicio SOAP

El acceso al API de integración se hará accediendo a una URL similar a la que adjuntamos a continuación, dirección donde se encuentra ubicado el WSDL del API:

```
http://<servidor>:<puerto>/inbox/app/<instancia>/serviceWS?wsdl
```

Además de la URL concreta (con la instancia de la aplicación escogida), deberemos disponer de unas credenciales de acceso al API, como veremos en el siguiente apartado.

Funcionalidades disponibles

El API de integración de Viafirma Inbox permite realizar las siguientes acciones:

- Realizar un ping al WS para comprobar la conexión con el mismo.
- Recuperar listado de todos los usuarios activos del sistema
- Buscar usuario por el campo "Person Id"
- Buscar usuarios por un campo de texto
- Recuperar listado de cargos
- Recuperar listado de grupos
- Recuperar listado de workflows públicos
- Recuperar tipos de rechazos configurados en el sistema
- Recuperar tipos de estado disponibles para una petición
- Recuperar tipos de acciones disponibles para una petición
- Recuperar tipos de acceso a la aplicación de verificación para una petición
- Enviar una petición
- Añadir comentarios a una petición
- Recuperar información de una petición
- Recuperar los comentarios de una petición
- Recuperar los ficheros de una petición y su enlace de descarga
- Redacción de una petición de firma desde la aplicación Viafirma Inbox y redirección al sistema externo
- Redacción de una petición de firma desde la aplicación Viafirma Inbox aplicando un flujo existente en el sistema y redirección al sistema externo
- Firma de un fichero desde la aplicación Viafirma Inbox y redirección al sistema externo
- Evaluar un flujo global, recuperando las líneas de firma (destinatarios)
- Añadir anexos a una petición
- Eliminar una petición
- Finalizar una petición
- Obtención del estado actual de una petición, y la información de los actores y acciones realizadas hasta el momento, incluyendo el caso de acciones realizadas por delegación.

Modelo de dominio

- [AccessTypeDTO](#)
- [AccessTypesResponseDTO](#)
- [ActionTypeDTO](#)
- [ActionTypesResponseDTO](#)
- [AddAnnexesResponseDTO](#)
- [AddCommentResponseDTO](#)
- [ComposeRequestResponseDTO](#)
- [DocumentDTO](#)
- [GroupDTO](#)
- [GroupsResponseDTO](#)
- [JobDTO](#)
- [JobsResponseDTO](#)
- [MetadataDTO](#)
- [MetadataResponseDTO](#)
- [PingResponseDTO](#)
- [RejectTypeDTO](#)
- [RejectTypesResponseDTO](#)
- [RequestAddresseeDTO](#)
- [RequestAnnexesResponseDTO](#)
- [RequestCommentDTO](#)
- [RequestCommentsResponseDTO](#)
- [RequestDocumentsResponseDTO](#)
- [RequestDTO](#)
- [RequestInfoResponseDTO](#)
- [RequestMetadataDTO](#)
- [RequestMetadataResponseDTO](#)
- [RequestStatusResponseDTO](#)
- [RequestUserDTO](#)
- [SendRequestResponseDTO](#)
- [SignatureDocumentDTO](#)
- [SignatureTicketResponseDTO](#)
- [SignedDocumentResponseDTO](#)
- [SignRequestResponseDTO](#)
- [StatusDTO](#)
- [StatusResponseDTO](#)
- [TemplateDTO](#)
- [TemplateFieldDTO](#)
- [UserDTO](#)
- [UserResponseDTO](#)
- [UsersResponseDTO](#)
- [WorkflowDTO](#)
- [WorkflowRequestResponseDTO](#)
- [WorkflowsResponseDTO](#)
- [WorkflowUsersDTO](#)
- [WSInformationDTO](#)

AccessTypeDTO

Campo	Descripción	Tipo
name	Nombre del tipo de acceso	String
shortName	Nombre corto del tipo de acceso	String

AccessTypesResponseDTO

Campo	Descripción	Tipo
result	Listado de tipos de acceso	AccessTypeDTO []

ActionTypeDTO

Campo	Descripción	Tipo
name	Nombre del tipo de acción	String
shortName	Nombre corto del tipo de acción	String

ActionTypesResponseDTO

Campo	Descripción	Tipo
result	Listado de tipos de acción	ActionTypeDTO []

AddAnnexesResponseDTO

Clase que extiende de [WSInformationDTO](#).

AddCommentResponseDTO

Clase que extiende de [WSInformationDTO](#).

ComposeRequestResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
redirectUrl	URL de redirección tras la redacción de una petición	String

DocumentDTO

Campo	Descripción	Tipo
boollsAnnex	¿Es un anexo?	boolean
data	Byte array con los datos del fichero	byte[]
downloadUrl	URL de descarga del fichero	String
fileExtension	Extensión del fichero	String
fileHash	Hash del fichero	String
fileMimeType	Mimetype del fichero	String
fileName	Nombre del fichero	String
fileSize	Tamaño del fichero en bytes	long
publicAccessId	Identificador único del fichero para su acceso público	String
reference	Referencia del fichero	String
signFormat	Formato de la firma	String
signType	Tipo de firma	String

GroupDTO

Campo	Descripción	Tipo
name	Nombre del grupo	String
publicAccessId	Identificador único del grupo para su acceso público	String

GroupsResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de grupos	GroupDTO []

JobDTO

Campo	Descripción	Tipo
name	Nombre del cargo	String
publicAccessId	Identificador único del cargo para su acceso público	String

JobsResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de cargos	JobDTO []

MetadataDTO

Campo	Descripción	Tipo
key	Clave del metadato	String
defaultValue	Valor por defecto del metadato	String
description	Descripción del metadato	String

MetadataResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de metadatos	MetadataDTO []

PingResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Resultado de la petición de ping	String

RejectTypeDTO

Campo	Descripción	Tipo
name	Nombre del tipo de rechazo	String
description	Descripción del tipo de rechazo	String

RejectTypesResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de motivos de rechazo	RejectTypeDTO []

RequestAddresseeDTO

Campo	Descripción	Tipo
action	Acción a realizar por le destinatario	String
rejectReason	Motivo del rechazo	String
rejectType	Tipo de rechazo	RejectTypeDTO
status	Estado de la petición para el destinatario	String
updated	Fecha de actualización	java.util.Date
user	Usuario destinatario	UserDTO

RequestAnnexesResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de documentos anexos	DocumentDTO []

RequestCommentDTO

Campo	Descripción	Tipo
comment	Mensaje del comentario	String
date	Fecha del comentario	java.util.Date
idRequest	Identificador de la petición	String
status	Estado de la petición para el destinatario	String
user	Usuario que realiza el comentario	UserDTO

RequestCommentsResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de comentarios	RequestCommentDTO []

RequestDocumentsResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de documentos	DocumentDTO []

RequestDTO

Campo	Descripción	Tipo
user	Usuario redactor de la petición	UserDTO
creationDate	Fecha de creación de la petición	java.util.Date
sendDate	Fecha de envío de la petición	java.util.Date
expirationDate	Fecha de expiración de la petición	java.util.Date
subject	Asunto de la petición	String
idRequest	Identificador de la petición	String
message	Mensaje de la petición	String
reference	Referencia de la petición	String
publicAccessId	Identificador único de la petición para su acceso público	String
accessType	Mensaje de la petición	String
accessUser	Usuario para el acceso a la verificación de una petición protegida por usuario y contraseña	String
accessPassword	Contraseña para el acceso a la verificación de una petición protegida por usuario y contraseña	String
notificationUrl	Url para la notificación de los cambios de estado de la petición	String
boolNotifyIfRead	¿Ha sido leída la petición?	boolean
boolNotifyIfReturned	¿Ha sido rechazada la petición?	boolean
boolNotifyIfSigned	¿Ha sido firmada la petición?	boolean
idViafirmaSign	Identificador de la firma en Víafirma	String
notificationLevel	Nivel de notificación al remitente de la petición	NotificationLevel
boolUseDefaultSignatureStamp	En caso de informarse y valer false se asume que el integrador no desea que la petición tenga sello de firma. En caso contrario la petición tendrá el sello de firma por defecto definido en la instancia	boolean
boolSignWithCertificate	En caso de valer true la firma de la petición deberá realizarse usando un sistema de firma con certificado. En caso contrario la petición puede ser firmada con cualquier sistema de firma	boolean

RequestInfoResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Petición	RequestDTO

RequestMetadataDTO

Campo	Descripción	Tipo
key	Clave del metadato	String
value	Valor del metadato	String
description	Descripción del metadato	String

RequestMetadataResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de metadatos	RequestMetadataDTO []

RequestStatusResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de destinatarios	RequestAddresseeDTO []

RequestUserDTO

Campo	Descripción	Tipo
personId	Identificador del usuario	String
action	Acción que realiza el usuario: F (firma) o VB (Visto Bueno)	String
signaturePositionDTO	Posición del sello de firma en el caso de firmas web	SignaturePositionDTO

SendRequestResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de peticiones	RequestDTO []

SignatureDocumentDTO

Campo	Descripción	Tipo
fileName	Nombre del fichero	String
data	Datos del fichero	byte[]

SignaturePositionDTO

Campo	Descripción	Tipo
page	Página en la que se ubicará la firma	Long
height	Altura de la firma	Double
width	Anchura de la firma	Double
x	Posición x de la firma	Long
y	Posición y de la firma	Long

SignatureTicketResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Informe de firmas	SignatureDocumentDTO

SignedDocumentResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Documento firmado	SignatureDocumentDTO

SignRequestResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
redirectUrl	URL de redirección tras la firma del documento	String

StatusDTO

Campo	Descripción	Tipo
name	Nombre del estado	String

StatusResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de estados	StatusDTO []

TemplateDTO

Campo	Descripción	Tipo
code	Código de la plantilla	String
params	Listado de parámetros de la plantilla	TemplateFieldDTO []

TemplateFieldDTO

Campo	Descripción	Tipo
key	Clave del parámetro de la plantilla	String
value	Valor del parámetro de la plantilla	String

UserDTO

Campo	Descripción	Tipo
nif	NIF del usuario	String
name	Nombre del usuario	String
surname1	Primer apellido del usuario	String
surname2	Segundo apellido del usuario	String
email	Email del usuario	String
phone	Teléfono del usuario	String

UserResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Usuario	UserDTO

UsersResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de usuarios	UserDTO []

WorkflowDTO

Campo	Descripción	Tipo
name	Nombre del flujo	String
publicAccessId	Identificador único del flujo para su acceso público	String

WorkflowRequestResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
redirectUrl	URL de redirección tras la petición	String

WorkflowsResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
result	Listado de flujos	WorkflowDTO []

WorkflowUsersDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
workflowStringRepresentation	Cadena representativa de un flujo	String

SignersApprovalsDataDTO

Campos propios de la clase

Campo	Descripción	Tipo
originalUser	Usuario original al que se le ha enviado la petición	UserDTO
actionUser	Usuario que realmente realiza la acción sobre la petición. En caso de no haber delegación esta propiedad tiene el mismo valor que originalUser	UserDTO
originalUserJob	Cargo del usuario original al que se le ha enviado la petición	String
actionUserJob	Cargo del usuario que realiza la acción sobre la petición	String
actionType	Tipo de acción realizada sobre la petición. Posibles valores: FIRMA, VISTOBUENO	String
actionStatus	Estado de la acción a llevar a cabo por el destinatario. Posibles valores: NUEVO, SIN_ACCION, LEIDO, ENESPERA, FIRMADO, VISTOBUENO, DEVUELTO, FINALIZADO, REACTIVADA	String
actionDate	Fecha de realización de la acción	java.util.Date
delegationText	Texto personalizado asociado a la delegación. Por ejemplo, motivo de la delegación, etc.	String

SignersApprovalsDataResponseDTO

Clase que extiende de [WSInformationDTO](#).

Campos propios de la clase

Campo	Descripción	Tipo
requestStatus	Estado actual global de la petición. Posibles estados: NOTSTARTED, INPROCESS, EXPIRED, REJECTED, COMPLETED	String
result	Listado de estados reales de la petición para cada uno de los destinatarios	SignersApprovalsDataDTO []

WSInformationDTO

Campo	Descripción	Tipo
error	¿Hay error?	boolean
responseCode	Código de respuesta	String
errorMessage	Mensaje de error	String

Posibles valores:

- **SUCCESS**: El proceso ha terminado correctamente
- **WS_ERROR_CODE_0**: Id de petición vacío
- **WS_ERROR_CODE_1**: Error al recuperar petición
- **WS_ERROR_CODE_2**: Error al recuperar el usuario anónimo WS
- **WS_ERROR_CODE_3**: Error al crear el comentario
- **WS_ERROR_CODE_4**: Error comentario vacío
- **WS_ERROR_CODE_5**: NIF vacío
- **WS_ERROR_CODE_6**: Destinatario no existe
- **WS_ERROR_CODE_7**: Error insertando petición
- **WS_ERROR_CODE_8**: Acción de petición inexistente
- **WS_ERROR_CODE_9**: Tipo de acceso a petición vacío o no existe
- **WS_ERROR_CODE_10**: Usuarios repetidos en la petición
- **WS_ERROR_CODE_11**: Lista de destinatarios vacía
- **WS_ERROR_CODE_12**: Lista de documentos vacía
- **WS_ERROR_CODE_13**: Nombre del documento vacío
- **WS_ERROR_CODE_14**: Contenido del documento vacío
- **WS_ERROR_CODE_15**: Id de flujo vacío
- **WS_ERROR_CODE_16**: Instancia de Inbox incorrecta
- **WS_ERROR_CODE_17**: Usuario no encontrado
- **WS_ERROR_CODE_18**: No se puede descargar el documento firmado
- **WS_ERROR_CODE_19**: Petición incompleta o problema al recuperar la última firma
- **WS_ERROR_CODE_20**: Error recuperando el documento de firma custodiado
- **WS_ERROR_CODE_21**: No se puede descargar el informe en pdf
- **WS_ERROR_CODE_22**: La petición de firma aún no está terminada
- **WS_ERROR_CODE_23**: Error generando el informe de firma en pdf
- **WS_ERROR_CODE_24**: Flujo no localizado**: no es de la instancia o no es un flujo global
- **WS_ERROR_CODE_25**: Error evaluando el flujo global
- **WS_ERROR_CODE_26**: Error evaluando la representación String del flujo
- **WS_ERROR_CODE_27**: Error generando plantillas
- **WS_ERROR_CODE_28**: La petición no fue creada vía API (Servicio Web)
- **WS_ERROR_CODE_29**: Error anexando documento a la petición
- **WS_ERROR_CODE_30**: Error faltan datos de usuario remitente
- **WS_ERROR_CODE_31**: Error al crear el usuario remitente
- **WS_ERROR_CODE_32**: Tipo de firma no soportado
- **WS_ERROR_CODE_33**: Tipo de autenticación no soportado
- **WS_ERROR_CODE_34**: No existe el usuario remitente
- **WS_ERROR_CODE_35**: Debe indicar un asunto para la petición
- **WS_ERROR_CODE_36**: Debe indicar un mensaje para la petición
- **WS_ERROR_CODE_37**: Debe indicar el nombre del usuario
- **WS_ERROR_CODE_38**: Debe indicar el primer apellido del usuario
- **WS_ERROR_CODE_39**: Ya existe un usuario con el mismo código
- **WS_ERROR_CODE_40**: Debe indicar el email del usuario
- **WS_ERROR_CODE_41**: Error al crear un usuario
- **WS_ERROR_CODE_42**: Debe indicarse la entidad por defecto del usuario
- **WS_ERROR_CODE_43**: No se encuentra la entidad
- **WS_ERROR_CODE_44**: No se encuentra la instancia
- **WS_ERROR_CODE_45**: Debe indicar un tipo de firma por defecto para el usuario externo
- **WS_ERROR_CODE_46**: El usuario existe y no puede ser modificado. Se trata de un administrador
- **WS_ERROR_CODE_47**: Datos incompletos al crear una entidad
- **WS_ERROR_CODE_48**: Error al crear una entidad
- **WS_ERROR_CODE_49**: Entidad no encontrada

- **WS_ERROR_CODE_50**: Error al crear un borrador
- **WS_ERROR_CODE_51**: No se puede eliminar una petición porque hay firmas o vistos buenos realizados
- **WS_ERROR_CODE_52**: No se puede finalizar la petición porque no hay firmas o vistos buenos realizados
- **WS_ERROR_CODE_53**: La fecha de caducidad debe ser posterior a la fecha de inicio
- **WS_ERROR_CODE_54**: Sello no encontrado
- **WS_ERROR_CODE_55**: La fecha de expiración, debe ser posterior a la actual
- **WS_ERROR_CODE_56**: La fecha de inicio, debe ser posterior a la actual
- **WS_ERROR_CODE_57**: Hay metadatos obligatorios no indicados
- **WS_ERROR_CODE_58**: Hay metadatos con valores no permitidos
- **WS_ERROR_CODE_59**: Debe posicionar todas las firmas
- **WS_ERROR_CODE_60**: Usuarios externos no permitidos en la comunicación interna
- **WS_ERROR_CODE_61**: Los usuarios inactivos no pueden recibir peticiones
- **WS_ERROR_CODE_62**: Los documentos enviados no cumplen con lo configurado (extensiones, size, numero)
- **WS_ERROR_CODE_63**: Plantilla no encontrada
- **WS_ERROR_CODE_64**: El mensaje excede el tamaño permitido
- **WS_ERROR_CODE_65**: Petición rechazada
- **WS_ERROR_CODE_66**: Petición caducada
- **WS_ERROR_CODE_67**: Número de telefono no valido
- **WS_ERROR_CODE_68**: Ya existe un documento con el csv indicado
- **WS_ERROR_CODE_69**: Existen dos o más sellos con el mismo nombre o código.
- **WS_ERROR_CODE_70**: Destinatario de la comunicación interna no existe
- **WS_ERROR_CODE_71**: El sistema API no puede usar al menos uno de los usuarios de firma desatendida
- **APP_TOKEN_INVALID**: El token de aplicación no es válido
- **APP_TOKEN_EXPIRED**: El token de aplicación ha expirado
- **WS_ERROR_CODE_72**: El tamaño de los documentos es mayor que el soportado en el envío de emails
- **WS_ERROR_CODE_401**: Error de autenticación
- **WS_ERROR_CODE_500**: Error general. Error no esperado o debido a fallos en sistemas externos.

Posibles valores:

- NO: No se notifica al remitente ninguna acción realizada sobre la petición
- ALL: Se notifican todas las acciones realizadas sobre la petición (Lectura, Firma/Visto bueno, Rechazo, Finalización)
- MEDIUM: Se notifican las acciones de Firma/Visto bueno, Rechazo, Finalización
- FINISH: Se notifica al remitente solamente cuando la petición ha sido finalizada (todos los destinatarios han realizado su acción)

Posibles valores:

- **VIAFIRMA_PLATFORM**: Autenticar utilizando Viafirma Platform
- **VIAFIRMA_FORTRESS**: Autenticar utilizando Viafirma Fortress

Descripción del WSDL

Todas las respuestas proporcionadas por los métodos del API de integración de Viafirma Inbox contendrán siempre la siguiente información (a parte de la información específica del método en concreto):

- **Error** (boolean): indica si ha habido error o no en la petición
- **ResponseCode** (String): Código de respuesta
- **ErrorMessage** (String): Si ha ocurrido un error en la petición, el mensaje de error describiendo el error ocurrido

Códigos de error

A continuación se enumeran los posibles códigos de error que devolverían los servicios web:

- SUCCESS,
- WS_ERROR_CODE_0, // Empty request id - Error id petición vacío
- WS_ERROR_CODE_1, // Unable to retrieve request info - Error al recuperar petición
- WS_ERROR_CODE_2, // Unable to retrieve WS User - Error al recuperar el usuario anónimo WS
- WS_ERROR_CODE_3, // Unable to create comment - Error al crear el comentario
- WS_ERROR_CODE_4, // Empty comment error - Error comentario vacío
- WS_ERROR_CODE_5, // Empty user ID error - Error nif vacío
- WS_ERROR_CODE_6, // Recipient not found error - Error destinatario no existe
- WS_ERROR_CODE_7, // Unable to create request - Error insertando petición
- WS_ERROR_CODE_8, // Unknown - Request action type - Acción de petición inexistente
- WS_ERROR_CODE_9, // Request access level empty or not found - Tipo de acceso a petición vacío o no existe
- WS_ERROR_CODE_10, // Repeated users in request - Usuarios repetidos en la petición
- WS_ERROR_CODE_11, // Empty recipients list - lista de destinatarios vacía
- WS_ERROR_CODE_12, // Empty documents list - Lista de documentos vacía
- WS_ERROR_CODE_13, // No document name found - Nombre del documento vacío
- WS_ERROR_CODE_14, // Empty document content - Contenido del documento vacío Manual de Integración ::
viafirma inbox Página 12 de 34
- WS_ERROR_CODE_15, // Empty workflow ID - Id de flujo vacío
- WS_ERROR_CODE_16, // Wrong Inbox Instance - Instancia de Inbox incorrecta
- WS_ERROR_CODE_17, // User not found - Usuario no encontrado
- WS_ERROR_CODE_18, // Unable to download signed document - No se puede descargar el documento firmado
- WS_ERROR_CODE_19, // The request is not finished or error while retrieving last signature - Petición incompleta o problema al recuperar la última firma
- WS_ERROR_CODE_20, // Unable to retrieve stored signature document - Error recuperando el documento de firma custodiado
- WS_ERROR_CODE_21, // Unable to retrieve pdf report - No se puede descargar el informe en pdf
- WS_ERROR_CODE_22, // Request not completed - La petición de firma aún no está terminada
- WS_ERROR_CODE_23, // Unable to generate PDF signature report - Error generando el informe de firma en pdf
- WS_ERROR_CODE_24, // Unable to found workflow, not global or not in this instance - Workflow no localizado, no es de la instancia o no es global
- WS_ERROR_CODE_25, // Error while evaluating global workflow - Error evaluando el workflow global
- WS_ERROR_CODE_26, // Error evaluating workflow string representation // Error evaluando la representación String del workflow
- WS_ERROR_CODE_401; // Authentication error - Error de autenticación
- WS_ERROR_CODE_500; // General error - Error general

addAnnexes

Añade un o varios documentos anexos a una petición existente.

Parámetros

- requestId (String): Id de la petición a la que se desea añadir los anexos.
- annexes ([DocumentDTO\[\]](#)): Array que contiene los documentos anexos de la petición. Deben informarse al menos los campos:
 - data
 - fileExtension
 - fileName
 - mimeType

Respuesta

[AddAnnexesResponseDTO\[\]](#)

- error (boolean): Indica si se ha producido un error al añadir el anexo.
- responseCode (String): Código del error en caso de haberlo.
- errorMessage (String): Mensaje del error en caso de haberlo.

addComment

Añade un comentario a una petición existente.

Parámetros

- requestId (String): Id de la petición a la que se desea añadir el comentario.
- body (String): Contenido del comentario a incorporar

Respuesta

[AddCommentRespondeDTO](#)

- error (boolean): Indica si se ha producido un error al añadir el comentario.
- responseCode (String): Código del error en caso de haberlo.
- errorMessage (String): Mensaje del error en caso de haberlo.

deleteRequest

Eliminar una petición que ha sido creada desde el Web Service. Las peticiones creadas desde la aplicación web no podrán eliminarse desde este método.

Parámetros

- requestId (String) (Obligatorio): El identificador de la petición que se desea eliminar.

Respuesta

[WSInformationDTO](#)

- error(Boolean): Indica si se ha producido algún error al eliminar la petición.
- reponseCode(String): Código del error.
- errorMessage(String): Mensaje del error.

endRequest

Finaliza una petición. Elimina los destinatarios que no hayan realizado acción, y da por finalizada la petición con las acciones de firma/visto bueno realizadas hasta el momento.

Parámetros

- requestId (String) (Obligatorio): El identificador de la petición que se desea finalizar.

Respuesta

WSInformationDTO

- error(Boolean): Indica si se ha producido algún error.
- reponseCode(String): Código del error.
- errorMessage(String): Mensaje del error.

evaluateGlobalWorkflow

Analiza un workflow global disponible en Viafirma Inbox y devuelve un objeto WorkflowUsersDTO que incluye como atributo una representación String del workflow, que puede ser utilizado posteriormente en `sendRequestWithStrWorkflow`. Sirve así para enviar una petición utilizando un flujo global de la misma instancia del usuario WS.

Respuesta

[WorkflowUsersDTO](#)

- `workflowStringRepresentation` (String): String del workflow, que puede usarse posteriormente en `sendRequestWithStrWorkflow`.

getAccessTypes

Recupera el listado de posibles accesos a la herramienta de verificación.

Respuesta

[AccessTypesResponseDTO](#)

- [AccessTypeDTO](#)[]): Array con la información de cada una de los tipos de acceso disponibles.

getActionTypes

Recupera el listado de acciones disponibles a realizar en una petición de Viafirma Inbox.

Respuesta

[ActionTypesResponseDTO](#)

- [ActionTypeDTO](#)[]): Array con la información de cada una de las acciones disponibles.

getGroups

Recupera el listado de grupos de trabajo disponibles en Viafirma Inbox

Respuesta

[GroupsResponseDTO](#)

- [GroupDTO\[\]](#): Array con la información de cada uno de los grupos. Uno de los campos devueltos es el `publicAccessId` que es el que tendrá que ser usado en el resto de métodos de la API.

getJobs

Recupera el listado de cargos disponibles en Viafirma Inbox

Respuesta

[JobsResponseDTO](#)

- [JobDTO\[\]](#): Array con la información de cada uno de los cargos. Uno de los campos devueltos es el `publicAccessId` que es el que tendrá que ser usado en el resto de métodos de la API.

getMetadataKeys

Recupera el listado de keys de los metadatos disponibles en Viafirma Inbox.

Respuesta

[MetadataResponseDTO](#)

- [MetadataDTO\[\]](#): Array con la información de cada uno de los keys de los metadatos disponibles en Viafirma Inbox.

getRejectTypes

Recupera el listado de motivos de rechazo disponibles en Viafirma Inbox

Respuesta

[RejectTypesResponseDTO](#)

- [RejectTypeDTO](#)[]): Array con la información de cada uno de los motivos de rechazo.

getRequestAnnexes

Recupera el listado de documentos anexos a una petición.

Parámetros

- RequestId (String) (Obligatorio): Id de la petición de la que se desea recuperar la información.

Respuesta

[RequestAnnexesResponseDTO](#)

- [DocumentDTO\[\]](#): Array con la información de cada uno de los documentos así como la url de acceso directo a la descarga del documento.

getRequestComments

Recupera el listado de comentarios asociados a una petición

Parámetros

- requestId (String) (Obligatorio): Id de la petición a la que se desea añadir el comentario.

Respuesta

[RequestCommentsResponseDTO](#)

- [RequestCommentDTO](#)[]): Array con la información de cada uno de los comentarios realizados a la petición con id pasado por parámetro.

getRequestDocuments

Recupera el listado de documentos a firmar asociados a una petición

Parámetros

- requestId (String) (Obligatorio): Id de la petición a la que se desea recuperar los ficheros

Respuesta

[RequestDocumentsResponseDTO](#)

- `RequestDocumentDTO[]` : Array con la información de cada uno de los documentos a firmar asociados a la petición con id pasado por parámetro.

getRequestInfo

Recupera la información general de una petición

Parámetros

- requestId (String) (Obligatorio): Id de la petición a la que se desea recuperar los ficheros

Respuesta

[RequestInfoResponseDTO](#)

- [RequestDTO](#): Objeto con la información asociada a la petición cuyo id corresponde al pasado por parámetro

getRequestMetadata

Recupera el listado de keys de los metadatos para una petición dada en Viafirma Inbox.

Parámetros

- requestId (String) (Obligatorio): Id de la petición de la que se desean obtener los keys de los metadatos.

Respuesta

[RequestMetadataResponseDTO](#)

- [RequestMetadataDTO](#)[]): Array con la información de cada uno de los keys de los metadatos de la petición en Viafirma Inbox.

getRequestStatus

Recupera el estado de la petición para cada uno de los destinatarios de la misma

Parámetros

- RequestId (String) (Obligatorio): Id de la petición a la que se desea recuperar los ficheros

Respuesta

[RequestStatusResponseDTO](#)

- [RequestAddresseeDTO](#)[]): Array con la información de cada uno de los destinatarios de la petición y el estado en el que se encuentra la misma para dicho destinatario.

getSignatureTicket

Recupera el resguardo de la petición.

Parámetros

- requestId (String) (Obligatorio): Id de la petición.

Respuesta

[SignatureTicketResponseDTO](#)

- [SignatureDocumentDTO](#) : fichero de resguardo de las firmas de la petición.

getSignedDocument

Recupera el documento o documentos firmados, en función del tipo de firma puede ser pdf o xml.

Parámetros

- requestId (String) (Obligatorio): Id de la petición.

Respuesta

[SignedDocumentResponseDTO](#)

- SignedDocumentDTO : fichero de firmado de la petición.

getStatus

Recupera el listado de posibles estados en los que se puede encontrar una petición con respecto a un usuario en concreto.

Respuesta

[StatusResponseDTO](#)

- [StatusDTO](#)[]): Array con la información de cada uno de los posibles estados.

getUserByPersonId

Recupera un usuario a partir de su personId.

Parámetros

- personId (String) (Obligatorio): Identificador del usuario.

Respuesta

[UserResponseDTO](#)

- [UserDTO](#): Datos del usuario con el personId dado.

getUsers

Recupera el listado de usuarios activos en Viafirma Inbox.

Respuesta

GetUsersResponseDTO

- [UserDTO](#)[]): Array con la información de cada uno de los usuarios recuperados.

getUsersBySearchTerm

Recupera el listado de usuarios activos de Viafirma Inbox que cumplen el criterio de búsqueda

Parámetros

- searchTerm (String) (Obligatorio): Término de búsqueda por el que se hará el filtrado de los usuarios

Respuesta

GetUsersResponseDTO

- [UserDTO](#)[]): Array con la información de cada uno de los usuarios recuperados.

getWorkflows

Recupera el listado de workflows disponibles en Viafirma Inbox.

Respuesta

`GetWorkflowsResponseDTO`

- [WorkflowDTO](#)[]): Array con la información de cada uno de los workflows disponibles en Viafirma Inbox. Uno de los campos devueltos es el `publicAccessId` que es el que tendrá que ser usado en el resto de métodos de la API.

ping

Método para hacer test de WS

Parámetros

- param (String): Cadena de texto que será devuelta en la respuesta.

Respuesta

[PingResponseDTO](#)

- result (String): La misma cadena pasada por parámetro al método

prepareComposeRequest

Prepara una petición y devuelve la dirección a la que tendremos que redireccionar. Una vez enviada la petición al WS, la aplicación tendrá que redireccionar a una ventana de navegador a la url proporcionada en la repuesta de la llamada donde se solicitará el certificado digital del usuario que envía la petición. Tras proporcionar dicho certificado, aparecerá la pantalla de redacción de peticiones con los datos ya cumplimentados (excepto el destinatario de la petición). Al pulsar el botón de "Enviar", una vez enviada la petición, el usuario será redireccionado a la url de retorno proporcionada en la petición inicial.

Amodo resumen, la secuencia de pasos a seguir son los siguientes:

- Realizar una petición a prepareComposeRequest pasando datos (firmante, fichero, etc.). En la respuesta viene una URL.
- Hacer redirect a esa URL (salto). El usuario está en la pantalla de redacción de peticiones de firma, ya en la pantalla con datos precargados: el fichero, el asunto, etc.
- Cuando el usuario envía la petición, Inbox responde a la URL de retorno pasada en la llamada del paso 1. En esa URL Inbox pasa por querystring los siguientes datos:
 - res= ok si todo ha ido bien o errores (peticionnoexiste, usercancel, errorviafirma, ...)
 - message= (si hay error, mensaje de error)
 - label= ID de la petición creada (Ej. R032-ICG9-RDD0-KWHR)

Parámetros

- PersonId (String) (Obligatorio): PersonId del usuario remitente de la petición (deberá ser el mismo que luego seleccione su certificado)
- DocumentName (String) (Obligatorio): Nombre del documento enviado
- DocumentContent (byte[]) (Obligatorio): Byte array del documento enviado
- returnUrl (String): URL donde se redireccionará al usuario al enviar la petición
- NotificationUrl (String): URL donde se notificarán los cambios realizado en la nueva petición creada.

Respuesta

[ComposeRequestResponseDTO](#)

- RedirectUrl: Url donde tendremos que redireccionar el navegador para que salte la pantalla de redacción de VíaFirma Inbox.

prepareComposeRequestAdvanced

Prepara una petición y devuelve la dirección a la que tendremos que redireccionar. Una vez enviada la petición al WS, la aplicación tendrá que redireccionar a una ventana de navegador a la url proporcionada en la repuesta de la llamada donde se solicitará el certificado digital del usuario que envía la petición. Tras proporcionar dicho certificado, aparecerá la pantalla de redacción de peticiones con los datos ya cumplimentados (excepto el destinatario de la petición). Al pulsar el botón de "Enviar", una vez enviada la petición, el usuario será redireccionado a la url de retorno proporcionada en la petición inicial.

Amodo resumen, la secuencia de pasos a seguir son los siguientes:

- Realizar una petición a prepareComposeRequestAdvanced pasando datos (firmante, fichero, etc.). En la respuesta viene una URL.
- Hacer redirect a esa URL (salto). El usuario está en la pantalla de redacción de peticiones de firma, ya en la pantalla con datos precargados: el fichero, el asunto, etc.
- Cuando el usuario envía la petición, Inbox responde a la URL de retorno pasada en la llamada del paso 1. En esa URL Inbox pasa por querysting los siguientes datos:
 - res= ok si todo ha ido bien o errores (peticionnoexiste, usercancel, errorviafirma, ...)
 - message= (si hay error, mensaje de error)
 - label= ID de la petición creada (Ej. R032-ICG9-RDD0-KWHR)

Parámetros

- senderPersonId (String) (Obligatorio): PersonId del usuario remitente de la petición (deberá ser el mismo que luego seleccione su certificado)
- addresseePersonId (Obligatorio): PersonId del usuario destinatario de la petición (deberá ser el mismo que luego seleccione su certificado)
- subject (String) (Obligatorio): Asunto del mensaje
- message (String): Mensaje del mensaje
- documentName (String) (Obligatorio): Nombre del documento enviado
- documentContent (byte[]) (Obligatorio): Byte array del documento enviado
- returnUrl (String): URL donde se redireccionará al usuario al enviar la petición
- notificationUrl (String): URL donde se notificarán los cambios realizado en la nueva petición creada.
- authSignType ([AuthSignType](#)): Plataforma empleada para realizar la firma

Respuesta

[ComposeRequestResponseDTO](#)

- RedirectUrl: Url donde tendremos que redireccionar el navegador para que salte la pantalla de redacción de VíaFirma Inbox.

prepareSignRequest

Prepara una petición de firma y devuelve la dirección a la que tendremos que redireccionar. Una vez enviada la petición de firma, la aplicación tendrá que redireccionar a una ventana de navegador a la url proporcionada en la respuesta de la llamada donde se solicitará el certificado digital del usuario que envía la petición para realizar la firma del fichero enviado. Tras proporcionar dicho certificado y realizar la firma del fichero, el usuario será redireccionado a la url de retorno proporcionada en la petición inicial. A modo resumen, la secuencia de pasos a seguir son los siguientes:

1. Realizar una petición a prepareSignRequest pasando datos (firmante, fichero, etc.). En la respuesta viene una URL.
2. Hacer redirect a esa URL (salto). El usuario será reenviado a la pantalla de firma donde tendrá que seleccionar su certificado.
3. Cuando el usuario firma la petición, Inbox responde a la URL de retorno pasada en la llamada del paso 1. En esa URL Inbox pasa por querystring los siguientes datos:
 - res= ok si todo ha ido bien o errores (particionnoexiste, usercancel, errorviafirma, ...)
 - message= (si hay error, mensaje de error)
 - label= ID de la petición creada (Ej. R032-ICG9-RDD0-KWHR)

Parámetros

- personId (String) (Obligatorio): PersonId del usuario remitente de la petición (deberá ser el mismo que luego seleccione su certificado)
- documentName (String) (Obligatorio): Nombre del documento enviado
- documentContent (byte[]) (Obligatorio): Byte array del documento enviado
- returnUrl (String): URL donde se redireccionará al usuario al enviar la petición
- notificationUrl (String): URL donde se notificarán los cambios realizado en la nueva petición creada.

Respuesta

[SignRequestResponseDTO](#)

- redirectUrl: Url donde tendremos que redireccionar el navegador para que el usuario realice la firma del fichero.

prepareSignRequestAdvanced

Se creará una nueva petición en viafirma inbox, con los datos indicados en los parámetros de entrada y el usuario será redirigido directamente a la pantalla de firma. Una vez completada la firma se redirigirá al usuario a la url indicada en el parámetro returnUrl.

Prepara una petición de firma y devuelve la dirección a la que tendremos que redireccionar. Una vez enviada la petición de firma, la aplicación tendrá que redireccionar a una ventana de navegador a la url proporcionada en la respuesta de la llamada donde se solicitará el certificado digital del usuario que envía la petición para realizar la firma del fichero enviado. Tras proporcionar dicho certificado y realizar la firma del fichero, el usuario será redireccionado a la url de retorno proporcionada en la petición inicial.

Amodo resumen, la secuencia de pasos a seguir son los siguientes:

1. Realizar una petición a prepareSignRequest pasando datos (firmante, fichero, etc.). En la respuesta viene una URL.
2. Hacer redirect a esa URL (salto). El usuario será reenviado a la pantalla de firma donde tendrá que seleccionar su certificado.
3. Cuando el usuario firma la petición, Inbox responde a la URL de retorno pasada en la llamada del paso 1. En esa URL Inbox pasa por querystring los siguientes datos:
 - res= ok si todo ha ido bien o errores (prtionnoexiste, usercancel, errorviafirma, ...)
 - message= (si hay error, mensaje de error)
 - label= ID de la petición creada (Ej. R032-ICG9-RDD0-KWHR)

Parámetros

- senderPersonId (String) (Obligatorio): PersonId del usuario remitente de la petición (deberá ser el mismo que luego seleccione su certificado)
- addresseePersonId (Obligatorio): PersonId del usuario destinatario de la petición (deberá ser el mismo que luego seleccione su certificado)
- subject (String) (Obligatorio): Asunto del mensaje
- message (String): Mensaje del mensaje
- documentName (String) (Obligatorio): Nombre del documento enviado
- documentContent (byte[]) (Obligatorio): Byte array del documento enviado
- returnUrl (String): Url donde se redireccionará al usuario al enviar la petición
- notificationUrl (String): Url donde se notificarán los cambios realizado en la nueva petición creada.
- metadatos ([RequestMetadataDTO](#)): Metadatos asociados a la petición.
- authSignType ([AuthSignType](#)): Plataforma empleada para realizar la firma.

Respuesta

[SignRequestResponseDTO](#)

- redirectUrl: Url donde tendremos que redireccionar el navegador para que el usuario realice la firma del fichero.

prepareWorkflowRequest

Prepara una petición y devuelve la dirección a la que tendremos que redireccionar. Una vez enviada la petición, la aplicación tendrá que redireccionar a una ventana de navegador a la url proporcionada en la respuesta de la llamada donde se solicitará el certificado digital del usuario que envía la petición. Una vez proporcionado dicho certificado, aparecerá la pantalla de redacción de peticiones con los datos ya cumplimentados junto con los destinatarios a los que se enviará la petición (en función del workflow seleccionado). Al pulsar el botón de "Enviar", una vez enviada la petición, el usuario será redireccionado a la url de retorno proporcionada en la petición inicial.

Ammodo resumen, la secuencia de pasos a seguir son los siguientes:

1. Realizar una petición a prepareWorkflowRequest pasando datos (firmante, fichero, etc.). También debemos indicar el publicAccessId del workflow que queremos utilizar en la petición. En la respuesta viene una URL. Manual de Integración :: viafirma inbox Página 23 de 34
2. Hacer redirect a esa URL (salto). El usuario será reenviado a la pantalla de redacción de peticiones con todos los datos precargados, incluso los datos de los destinatarios que serán aquellos usuarios que pertenezcan al flujo seleccionado.
3. Cuando el usuario pulse el botón de envío de petición, Inbox responde a la URL de retorno pasada en la llamada del paso 1. En esa URL Inbox pasa por querystring los siguientes datos:
 - res= ok si todo ha ido bien o errores (prtionnoexiste, usercancel, errorviafirma, ...)
 - message= (si hay error, mensaje de error)
 - label= ID de la petición creada (Ej. R032-ICG9-RDD0-KWHR)

Parámetros

- personId (String) (Obligatorio): PersonId del usuario remitente de la petición (deberá ser el mismo que luego seleccione su certificado)
- documentName (String) (Obligatorio): Nombre del documento enviado
- documentContent (byte[]) (Obligatorio): Byte array del documento enviado
- workflowId (String) (Obligatorio): Identificador público del workflow seleccionado
- returnUrl (String): URL donde se redireccionará al usuario al enviar la petición
- notificationUrl (String): URL donde se notificarán los cambios realizado en la nueva petición creada.

Respuesta

[WorkflowRequestResponseDTO](#)

- redirectUrl: Url donde tendremos que redireccionar el navegador para que salte la pantalla de redacción de Víafirma Inbox.

prepareWorkflowRequestAdvanced

Prepara una petición y devuelve la dirección a la que tendremos que redireccionar. Una vez enviada la petición, la aplicación tendrá que redireccionar a una ventana de navegador a la url proporcionada en la respuesta de la llamada donde se solicitará el certificado digital del usuario que envía la petición. Una vez proporcionado dicho certificado, aparecerá la pantalla de redacción de peticiones con los datos ya cumplimentados junto con los destinatarios a los que se enviará la petición (en función del workflow seleccionado). Al pulsar el botón de "Enviar", una vez enviada la petición, el usuario será redireccionado a la url de retorno proporcionada en la petición inicial.

Amodo resumen, la secuencia de pasos a seguir son los siguientes:

1. Realizar una petición a prepareWorkflowRequest pasando datos (firmante, fichero, etc.). También debemos indicar el publicAccessId del workflow que queremos utilizar en la petición. En la respuesta viene una URL. Manual de Integración :: viafirma inbox Página 23 de 34
2. Hacer redirect a esa URL (salto). El usuario será reenviado a la pantalla de redacción de peticiones con todos los datos precargados, incluso los datos de los destinatarios que serán aquellos usuarios que pertenezcan al flujo seleccionado.
3. Cuando el usuario pulse el botón de envío de petición, Inbox responde a la URL de retorno pasada en la llamada del paso 1. En esa URL Inbox pasa por querystring los siguientes datos:
 - o res= ok si todo ha ido bien o errores (prtionnoexiste, usercancel, errorviafirma, ...)
 - o message= (si hay error, mensaje de error)
 - o label= ID de la petición creada (Ej. R032-ICG9-RDD0-KWHR)

Parámetros

- personId (String) (Obligatorio): PersonId del usuario remitente de la petición (deberá ser el mismo que luego seleccione su certificado)
- workflowId (String) (Obligatorio): Identificador público del workflow seleccionado
- subject (String) (Obligatorio): Asunto del mensaje
- message (String): Mensaje del mensaje
- documentName (String) (Obligatorio): Nombre del documento enviado
- documentContent (byte[]) (Obligatorio): Byte array del documento enviado
- returnUrl (String): URL donde se redireccionará al usuario al enviar la petición
- notificationUrl (String): URL donde se notificarán los cambios realizado en la nueva petición creada.
- authSignType ([AuthSignType](#)): Plataforma empleada para realizar la firma.

Respuesta

[WorkflowRequestResponseDTO](#)

- redirectUrl: Url donde tendremos que redireccionar el navegador para que salte la pantalla de redacción de Víafirma Inbox.

sendRequest

Inserta una nueva petición

Parámetros

- request ([RequestDTO](#)) (Obligatorio). Datos de la petición. Debe informarse al menos el campo:
 - subject
- addressees ([RequestUserDTO](#)[][]) (Obligatorio): Array que contiene el flujo de firma completo (líneas, grupos y usuarios). Cada objeto RequestUserDTO contiene un person Id y la acción a realizar. La acción es de tipo ActionType (el sistema dispone de un método para recuperar los ActionType disponibles: F = firma, VB = visto bueno).

NOTA: existe el método evaluateGlobalWorkflow, que evalúa un flujo GLOBAL existente en la instancia, analizar los destinatarios, y recuperar un objeto de este tipo RequestUserDTO[][], de forma que se puede realizar primero esta llamada, y posteriormente sendRequest.

- documents ([DocumentDTO](#)[][]) (Obligatorio): Array que contiene los documentos a firmar. Deben informarse al menos los campos:
 - data
 - fileExtension
 - fileName
 - fileMimeType
- annexes ([DocumentDTO](#)[]): Array que contiene los documentos anexos de la petición. Deben informarse al menos los campos:
 - data
 - fileExtension
 - fileName
 - fileMimeType

Respuesta

[SendRequestResponseDTO](#)

- [RequestDTO](#): Objeto con toda la información relacionada con la nueva petición creada.

sendRequestWithStrWorkflow

Similar a `sendRequest`, con la principal diferencia que en lugar de usar un objeto `RequestUserDTO`, utiliza la representación (String) de un workflow en la caja predictiva de Viafirma Inbox. Por ejemplo: "NOMBRE USUARIO 1" [usuario1@viafirma.com] >"NOMBRE USUARIO 2" [usuario2@viafirma.com]>.

También se dispone de un método `evaluateGlobalWorkflow`, que devuelve un objeto con esa representación String, de forma que se puede llamar primero a ese método, y pasar el resultado como parámetro a este `sendRequestWithStrWorkflow`.

Parámetros

- `request` ([RequestDTO](#)) (Obligatorio). Datos de la petición. Debe informarse al menos el campo:
 - `subject`
- `workflowStringRepresentation` (String) (Obligatorio): representación String del workflow, por ejemplo: "NOMBRE USUARIO 1" [usuario1@viafirma.com] >"NOMBRE USUARIO 2" [usuario2@viafirma.com]>
- `documents` ([DocumentDTO](#)[][]) (Obligatorio): Array que contiene los documentos a firmar. Deben informarse al menos los campos:
 - `data`
 - `fileExtension`
 - `fileName`
 - `fileMimeType`
- `annexes` ([DocumentDTO](#)[][]): Array que contiene los documentos anexos de la petición. Deben informarse al menos los campos:
 - `data`
 - `fileExtension`
 - `fileName`
 - `fileMimeType`

Respuesta

[SendRequestResponseDTO](#)

- [RequestDTO](#): Objeto con toda la información relacionada con la nueva petición creada.

sendRequestWithTemplate

Inserta una nueva petición incluyendo los datos para la generación de documentos a partir de plantillas

Parámetros

- request ([RequestDTO](#)) (Obligatorio). Datos de la petición. Deben informarse al menos el campo:
 - subject
- addressees ([RequestUserDTO](#)[][]) (Obligatorio): Array que contiene el flujo de firma completo (líneas, grupos y usuarios). Cada objeto RequestUserDTO contiene un person Id y la acción a realizar. La acción es de tipo ActionType (el sistema dispone de un método para recuperar los ActionType disponibles: F = firma, VB = visto bueno).

NOTA: existe el método evaluateGlobalWorkflow, que evalúa un flujo GLOBAL existente en la instancia, analizar los destinatarios, y recuperar un objeto de este tipo RequestUserDTO[][], de forma que se puede realizar primero esta llamada, y posteriormente sendRequest.

- documents ([DocumentDTO](#)[]) (Obligatorio): Array que contiene los documentos a firmar. Deben informarse al menos los campos:
 - data
 - fileExtension
 - fileName
 - fileMimeType
- annexes ([DocumentDTO](#)[]): Array que contiene los documentos anexos de la petición. Deben informarse al menos los campos:
 - data
 - fileExtension
 - fileName
 - fileMimeType
- templates ([TemplateDTO](#)[]): Array que contiene los datos de las plantillas para la generación de documentos. Deben informarse al menos los campos:
 - code
 - params

Respuesta

[SendRequestResponseDTO](#)

- [RequestDTO](#): Objeto con toda la información relacionada con la nueva petición creada.

getSignersApprovalsData

Método que devuelve el estado actual de una petición, y la información de los actores y acciones realizadas hasta el momento, incluyendo el caso de acciones realizadas por delegación.

Parámetros

- requestId (String) (Obligatorio): El identificador de la petición de la que se desea obtener su estado actual, sus actores finales y sus respectivas acciones.

Respuesta

[SignersApprovalsDataResponseDTO](#)

Ejemplos de integración

Aplicación de ejemplo en Java

En la zona de **Documentación** de la aplicación Viafirma Inbox existe una [aplicación de ejemplo programada en Java](#) que realiza una petición a cada uno de los métodos definidos en el WSDL.

En la página index.jsp de la aplicación de ejemplo hay una serie de variables que hay que modificar para el correcto funcionamiento de la misma:

- **urlConexion:** Url donde se encuentra instalado Viafirma Inbox.
- **returnUrl:** Dirección de retorno. En este caso redireccionamos a la misma página jsp
- **notificationUrl:** Url donde se recibirán las notificaciones. En este ejemplo tenemos creado un servlet que escribe un mensaje en la consola de Java cada vez que se recibe una notificación.
- **System:** Sistema utilizado de Viafirma Inbox. Si no está activa la seguridad en la API, este valor es vacío
- **key:** Key de API de Viafirma Inbox. Vacío si la seguridad no está activada
- **nifPruebas:** Nif de pruebas. En la aplicación de ejemplo (/site/docs) hay un certificado de pruebas (nombre: Juan Gil Palomo, nif: 11111111R, clave: 12345). Este certificado será solicitado por Viafirma para las peticiones asíncronas. Puede ser modificada esta variable con otro nif.
- **WorkflowTest:** PublicAccessId del workflow que queremos usar para el ejemplo de redacción de petición utilizando un workflow disponible.

El primer paso para poder realizar una integración con Java, es acceder a la aplicación de ejemplo y copiamos todo lo necesario. Tendremos que copiar todo el package “com.viafirma.tray.ws”. En este package se encuentran definidos todos los DTO necesarios para enviar los datos de las peticiones y recibir las respuestas. Otra opción sería generar todos los objetos de acceso desde el WSDL directamente.

Existe un servlet en la aplicación de ejemplo “NoficationServlet.java” que recibirá todas las notificaciones de las peticiones enviadas desde el Webservice. Ahora mismo en dicho servlet simplemente pintaremos por consola el label de la petición modificada, la acción realizada y si se han finalizado o no todas las acciones de la petición.

En función de si tenemos activa o no la seguridad en la API de Viafirma Inbox, tendremos que enviar las cabeceras con los datos de acceso a la API. En el caso de Java, para instanciar el cliente sería como se indica en los siguientes ejemplos (estos ejemplos corresponden a la conexión utilizando el estándar JaxWS):

- API con seguridad activada:

```

41
42 public ServiceWS getService() throws MalformedURLException{
43     QName qNameService = new QName("http://tray.viavansi.com", "ServiceWS");
44     URL wsdl = new URL(url + "?wsdl");
45     Service service = Service.create(wsdl, qNameService);
46     ServiceWS serviceWS = service.getPort(ServiceWS.class);
47     Map<String, Object> map=((BindingProvider)serviceWS).getRequestContext();
48     map.put(BindingProvider.USERNAME_PROPERTY, user);
49     map.put(BindingProvider.PASSWORD_PROPERTY, key);
50     map.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, url);
51
52     return serviceWS;
53 }
```

- API con seguridad desactivada:

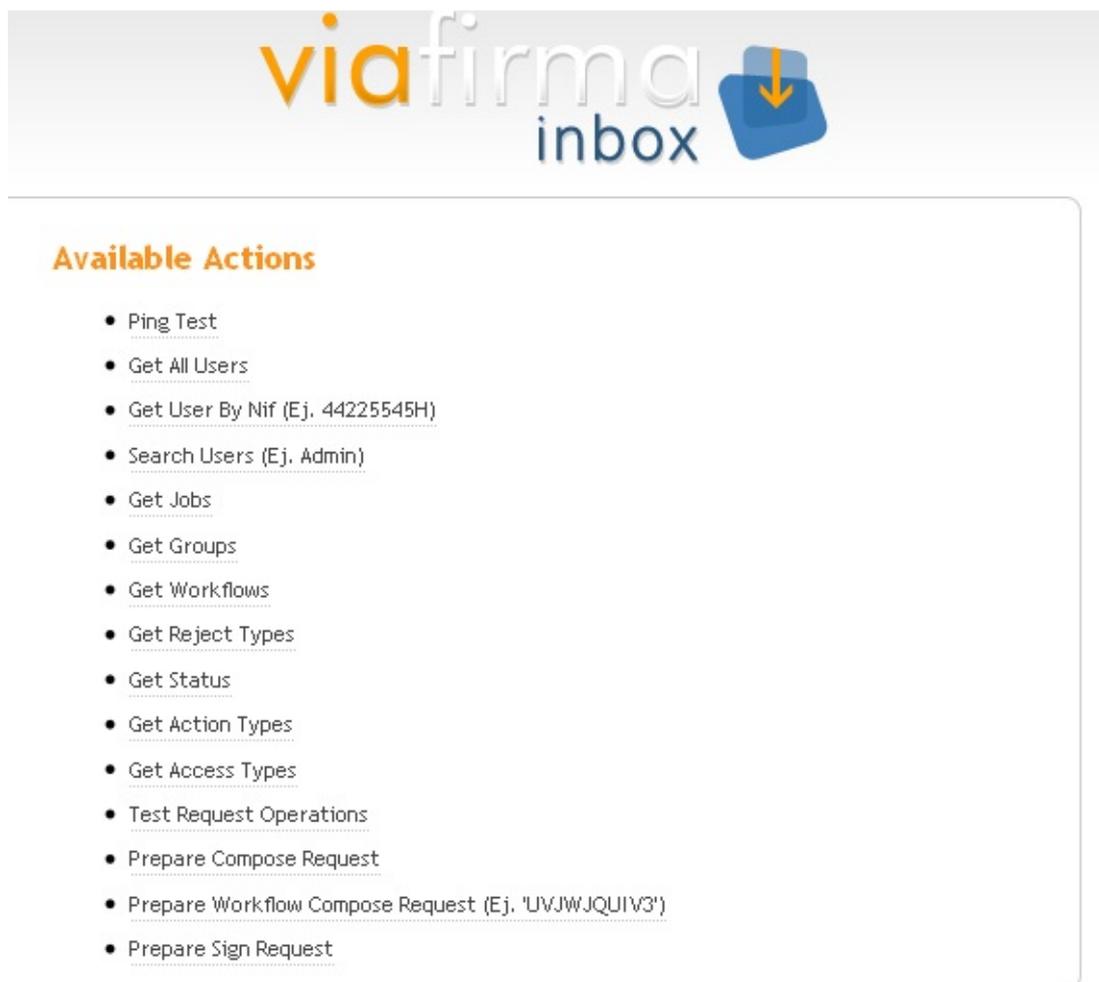
```

41
42 public ServiceWS getService() throws MalformedURLException{
43     QName qNameService = new QName("http://tray.viavansi.com", "ServiceWS");
44     URL wsdl = new URL(url + "?wsdl");
45     Service service = Service.create(wsdl, qNameService);
46     ServiceWS serviceWS = service.getPort(ServiceWS.class);
47
48     return serviceWS;
49 }

```

De esta forma, ya tenemos acceso a un objeto ServiceWS que es el que tiene disponibles todos los métodos descritos en el apartado anterior.

En la siguiente captura podemos ver la aplicación de ejemplo:



2010 Viafirma.

Cada uno de los enlaces de la aplicación de ejemplo realiza la acción que indica. La acción “Test Request Operations” es un poco especial porque no realiza una única acción. Al pulsar sobre esta acción se realizarán las siguientes operaciones con el API:

- Envía una petición
- Añade un comentario a la nueva petición
- Recupera la información de la petición
- Recupera los comentarios de la petición
- Recupera los ficheros a firmar de la petición y pinta un enlace de descarga de cada fichero
- Recupera los ficheros anexos de la petición y pinta un enlace de descarga de cada fichero.

Aplicación de ejemplo en .NET

En la zona de “Documentación” de la aplicación Viafirma Inbox existe una [aplicación de ejemplo programada en .NET](#) que realiza una petición a cada uno de los métodos definidos en el WSDL.

En la página Default.aspx de la aplicación de ejemplo hay una serie de variables que hay que modificar para el correcto funcionamiento de la misma:

- **urlConexion:** Url donde se encuentra instalado Viafirma Inbox
- **returnUrl:** Dirección de retorno. En este caso redireccionamos a la misma página jsp
- **notificationUrl:** Url donde se recibirán las notificaciones. En este ejemplo tenemos creado un servlet que escribe un mensaje en la consola de Java cada vez que se recibe una notificación.
- **System:** Sistema utilizado de Viafirma Inbox. Si no está activa la seguridad en la API, este valor es vacío
- **key:** Key de API de Viafirma Inbox. Vacío si la seguridad no está activada
- **nifPruebas:** Nif de pruebas. En la aplicación de ejemplo (/docs) hay un certificado de pruebas (nombre: Juan Gil Palomo, nif: 11111111R, clave: 12345). Este certificado será solicitado por Viafirma para las peticiones asíncronas. Puede ser modificada esta variable con otro nif.
- **WorkflowTest:** PublicAccessId del workflow que queremos usar para el ejemplo de redacción de petición utilizando un workflow disponible.

Este ejemplo realiza las mismas operaciones que el ejemplo Java anteriormente descrito. Para generar el cliente del Webservice en .NET Framework es necesario realizar las siguientes operaciones:

- Abrir la consola de .NET
- Ejecutar el comando “wsdl URL_WSDL” en el directorio donde queremos generar el cliente.
 - Si tenemos seguridad activada:
 - Sobreescibir el método “protected System.Net.WebRequest GetWebRequest(Uri uri)” de la clase generada que extiende de “System.Web.Services.Protocols.SoapHttpClientProtocol” (La clase debería llamarse “ServiceWS”):

```

79
80     protected override System.Net.WebRequest GetWebRequest(Uri uri)
81     {
82         HttpWebRequest request = (HttpWebRequest)base.GetWebRequest(uri);
83
84         NetworkCredential networkCredentials = Credentials.GetCredential(uri, "Basic");
85
86         if (networkCredentials != null)
87         {
88             byte[] credentialBuffer = new UTF8Encoding().GetBytes(
89                 networkCredentials.UserName + ":" + networkCredentials.Password);
90             request.Headers["Authorization"] = "Basic" +
91                 Convert.ToBase64String(credentialBuffer);
92         }
93         else
94         {
95             throw new ApplicationException("No network credentials");
96         }
97
98         return request;
99     }

```

- Para instanciar el cliente:

```
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63
```

```
public ServiceWS getRemoteObject()  
{  
    ServiceWS servicio = new ServiceWS(urlWS);  
  
    Uri uri = new Uri(urlWS);  
    NetworkCredential netCredential = new NetworkCredential(user, pass);  
    ICredentials credentials = netCredential.GetCredential(uri, "Basic");  
    servicio.Credentials = credentials;  
  
    return servicio;  
}
```

- o Si no tenemos seguridad activada:

- Para instanciar el cliente:

```
51  
52  
53  
54  
55  
56
```

```
public ServiceWS getRemoteObject()  
{  
    ServiceWS servicio = new ServiceWS(urlWS);  
  
    return servicio;  
}
```