# viafirma

## documents sdk java

viafirma
la firma universal

# Tabla de contenido

# Viafirma Documents SDK (Java)

SDK generada para facilitar la integración con los servicios rest desde aplicaciones java.

Haciendo uso de este SDK podrá interactuar con facilidad con la funcionalidades disponibles en la Suite Viafirma Documents y sus aplicaciones móviles.

## *Versión 3.4.3 actualizada el 12/04/2017*

# Integración en proyecto java

## Para proyecto maven

Es necesario añadir el siguiente repositorio

```
<repository>
    <id>viavansi-repo</id>
    <name>Viavansi Maven Repository</name>
    <url>http://repositorio.viavansi.com/repo</url>
</repository>
```

Añadir la siguiente dependencia

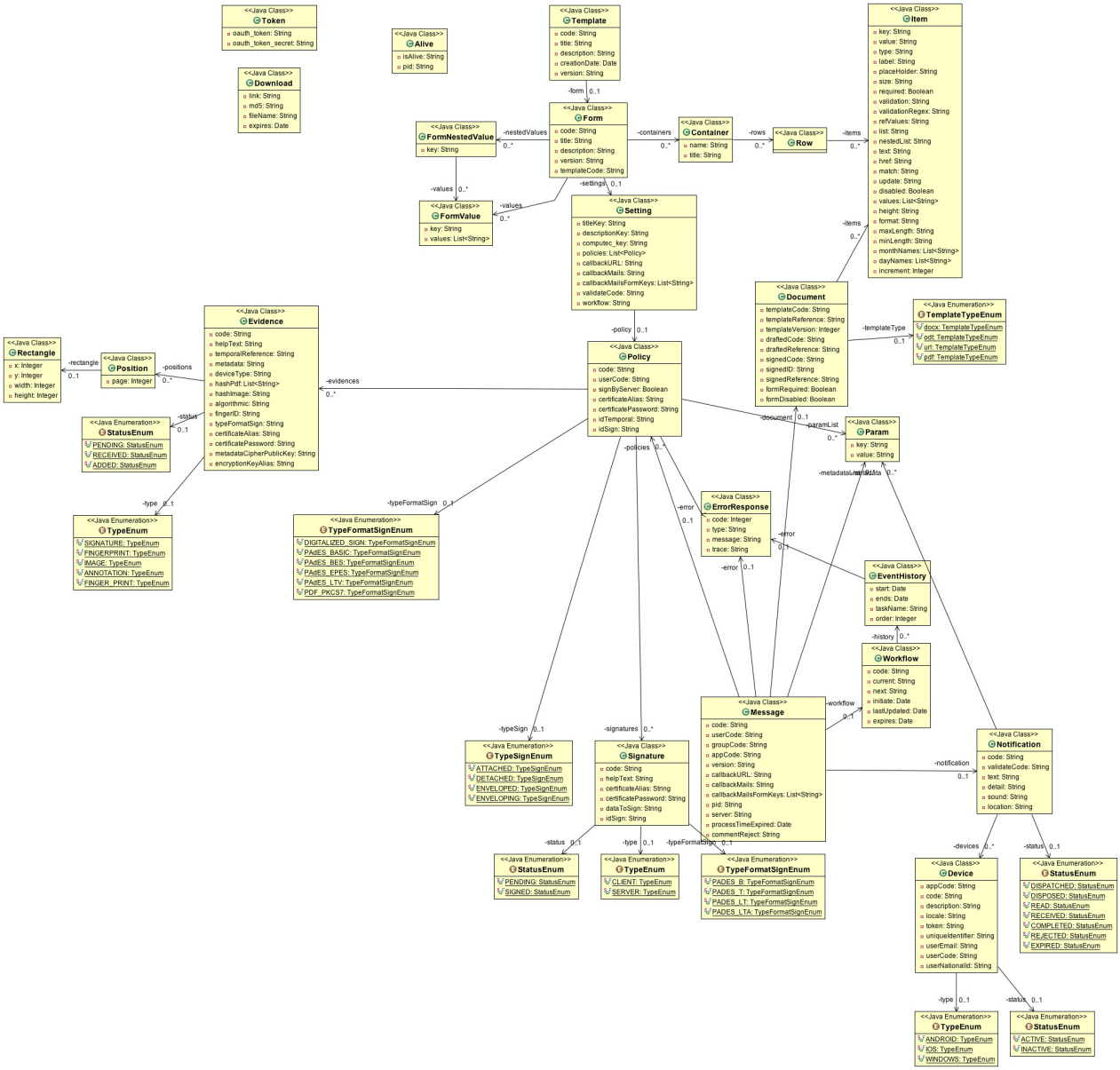
```
<dependency>
    <groupId>com.viafirma</groupId>
    <artifactId>documents-sdk-java</artifactId>
    <version>3.X.X</version>
</dependency>
```

## Descargar el sdk

También puede descargar directamente el jar de la versión correspondiente desde el siguiente enlace http://repositorio.viavansi.com/artifactory/libs-releases/com/viafirma/documents-sdk-java/3.X.X/documents-sdk-java-2.X.X.jar y añadirlo directamente a su proyecto

# Modelo de datos

En el siguiente diagrama de clases puede ver las relaciones entres los objetos utilizados por este SDK y sus atributos.



# Ejemplos de uso del SDK

## Parámetros de configuración para los ejemplos

- **API_URL** Dirección para el acceso a los servicios rest por ejemplo https://localhost:8080/mobile-services/api
- **CONSUMER_KEY** Identificador de la aplicación configurada en el backend que permite el accesos a los servicios rest.
- **CONSUMER_SECRET** Clave de la aplicación anterior.
- **USER_CODE** Código de acceso del usuario que accede a los servicios.
- **USER_PASSWORD** Clave de usuario que accede a los servicios.
- **OAUTH_TYPE** Tipo de autenticación para el acceso al API, puede tomar los valores OAUTH_APPLICATION | OAUTH_USER según la configuración de la aplicación en el backend, indica si se necesitan credencales de usuario o no para acceder a los servicios.
- **AUTH_MODE** indica el tipo de autenticación utilizado para aceptar un token de OAuth en la versión actual siempre tiene el valor **"client_auth"**
- **MESSAGE_CODE** Código del mensaje de ejemplo.
- **TEMPLATE_CODE** Código de la plantilla que queremos utilizar.
- **TEMPLATE_TYPE** Tipo la plantilla configurada en el parámetro TEMPLATE_CODE y puede tomar los valores (docx | odt | url).
- **DEVICE_CODE** Código del dispositivo de ejemplo.
- **DEVICE_DESCRIPTION** Descripción del dispositivo de ejemplo.
- **DEVICE_LOCALE** Idioma del dispositivo, por ejemplo "es_ES"
- **DEVICE_TYPE** Tipo de dispositivo (ANDROID|IOS)
- **DEVICE_APP_CODE** Código de la aplicación (configurado en el backend) al que pertenece el dispositivo de ejemplo

# Solicitud de acceso al API (OAuth 1.0)

Los servicios REST expuestos por viafirma documents están securizados con OAuth 1.0

Para más información se puede consultar la Documentación oficial de OAuth 1.0.

Desde la administración de aplicaciones del backend, podremos configurar la seguridad de acceso al API dos formas:

- **OAuth a nivel de aplicación** en el acceso a los servicios REST solo se identifica a la aplicación que hace uso de los mismo y no se requiere la renovación de token de acceso. Pensado para integraciones desde aplicaciones como CRM
- **OAuth a nivel de usuario** en el acceso a los servicios REST se identifica a la aplicación que hace uso de los servicios y al usuario que hace uso de la aplicación, se requiere la renovación de token cada X minutos, según lo configurado en el backend. Pensado para la integración de aplicaciones en la que que participa el usuario final.

## Acceso al API

En el siguiente ejemplo podemos ver como configurar el cliente para hacer uso de los direfentes servicios.

```
V3Api api = new V3Api();
api.setBasePath(API_URL);
api.setConsumerKey(CONSUMER_KEY);
api.setConsumerSecret(CONSUMER_SECRET);

//Configure proxy
//api.setProxyHost("127.0.0.1");
//api.setProxyPort(3128);

if (OAUTH_TYPE == OAuthType.OAUTH_USER) {
    api.setUser(USER_CODE);
    api.setPassword(USER_PASSWORD);
    api.setAuth_mode(AUTH_MODE);
    api.generateNewToken();
}
```

En el ejemplo anterior podemos observar como en el caso de necesitar el acceso al API con OAuth a nivel de usuario, indicamos los datos de acceso del usuario y con el código `api.generateNewToken();` solicitamos un nuevo token.

# Listar dispositivos

En el siguiente ejemplo puede ver como recuperar todos los dispositivos registrados por un usuario, este servicio le será de utilidad para localizar el dispositivo al que desea enviar el documento a firmar.

```
List<Device> devices = V3devicesApi.getInstance().fi
```

# Información de una plantilla

En el siguiente ejemplo puede ver como recuperar la información de las plantillas disponibles en el backend, este servicio le será de utilidad para crear nuevas solicitudes utilizadando la configuración de la plantilla seleccionada.

```
Template template = V3templateApi.getInstance().find
```

Aquí puede ver como recuperar las variables configuradas en una plantilla, a las que puede debe dar valor para la generación del documento.

```
//Find items in template form
Form form = template.getForm();
for(Container container: form.getContainers()){
    for(Row row: container.getRows()){
        for(Item item: row.getItems()){
            if (item.getKey().equals("KEY_01")) {
                item.setValue("Jhon");
            } else if (item.getKey().equals("KEY_02"
                item.setValue("Doe");
            } else if (item.getKey().equals("KEY_03"
                item.setValue("11111111T");
            }
        }
    }
}
```

Aquí puede ver como recuperar la configuración de firma configurada en la plantilla para utilizarla en su nueva solicitud.

```
            // Copy policies form template
            message.setPolicies(template.getForm().getSettings()
```

# Nuevo documento sin dispositivo

En el siguiente ejemplo puede ver como solicitar la generación de un documento a partir de una plantilla configurada en el backend. Para este caso debe de utilizar en workflow **EX005** dado que el documento no será enviado a ningún dispositivo.

```
Message message = new Message();

// Create notification info
Notification notification = new Notification();
notification.setText("Title");
notification.setDetail("Detail");
message.setNotification(notification);

Workflow workflow = new Workflow();
// only generate PDF from template. It´s not sent to
workflow.setCode("EX005");
message.setWorkflow(workflow);

Document document = new Document();
document.setTemplateCode(TEMPLATE_CODE);
document.setTemplateType(TEMPLATE_TYPE);
document.setItems(new ArrayList<Item>());

Item item01 = new Item();
item01.setKey("KEY_01");
item01.setValue("Jhon");
document.getItems().add(item01);

Item item02 = new Item();
item02.setKey("KEY_02");
item02.setValue("Doe");
document.getItems().add(item02);

Item item03 = new Item();
item03.setKey("KEY_03");
item03.setValue("11111111T");
```

```
            document.getItems().add(item03);

            message.setDocument(document);

            //java example in https://github.com/viavansi/ms-cal
            message.setCallbackURL("https://localhost:8080/ms-ca

            //send document by email (optional)
            message.setCallbackMails("user1@mail.com,user2@mail.

            messageCode = V3messagesApi.getInstance().sendMessag
```

En el ejemplo anterior podemos ver como podemos indicar la url del servicio donde se avisará cuando el documento esté disponible y como configurar emails a los que enviar el documento generado.

De todas formas siempre podemos esperar a que el documento sea generado, como podemos ver en el siguiente ejemplo

```
            int count = 100;
            String status = null;
            while (count > 0) {
                count--;
                Message msg = V3messagesApi.getInstance().getMes
                status = msg.getWorkflow().getCurrent();

                if ("RESPONSED".equals(status)) {
                    Download download = V3documentsApi.getInstar
                    byte[] pdf = IOUtils.toByteArray((new URL(do
                    Assert.assertNotNull(pdf);
                } else {
                    Thread.sleep(1000);
                }
            }
```

# Nuevo documento y envío al dispositivo

En el siguiente ejemplo puede ver como solicitar la generación de un documento y configurar el dispositivo al que deseamos enviar la solicitud de firma.

```
        Message message = new Message();

        Document document = new Document();
        document.setTemplateCode(TEMPLATE_CODE);
        document.setTemplateType(TEMPLATE_TYPE);
        document.setItems(new ArrayList<Item>());

        Item item01 = new Item();
        item01.setKey("KEY_01");
        item01.setValue("Jhon");
        document.getItems().add(item01);

        Item item02 = new Item();
        item02.setKey("KEY_02");
        item02.setValue("Doe");
        document.getItems().add(item02);

        Item item03 = new Item();
        item03.setKey("KEY_03");
        item03.setValue("11111111T");
        document.getItems().add(item03);

        message.setDocument(document);

        //Find device in user device list
        Device device = null;
        List<Device> devices = V3devicesApi.getInstance().findDe
        for (Device d: devices) {
            if (DEVICE_CODE.equals(d.getCode())) {
                device = d;
                break;
            }
```

```
        }

        Notification notification = new Notification();
        notification.setText("Notification Example");
        notification.setDevices(new ArrayList<Device>());
        notification.getDevices().add(device);

        message.setNotification(notification);

        message.setPolicies(new ArrayList<Policy>());
        Policy policy = new Policy();

        policy.setEvidences(new ArrayList<Evidence>());
        Evidence evidence = new Evidence();
        evidence.setType(TypeEnum.SIGNATURE);
        evidence.setHelpText("User signature");
        evidence.setTypeFormatSign("XADES_B");
        policy.getEvidences().add(evidence);

        policy.setSignatures(new ArrayList<Signature>());
        Signature signature = new Signature();
        signature.setType(com.viafirma.documents.sdk.java.model.
        signature.setHelpText("Server signature");
        signature.setTypeFormatSign(com.viafirma.documents.sdk.j
        policy.getSignatures().add(signature);

        message.getPolicies().add(policy);

        //java example in https://github.com/viavansi/ms-callbac
        message.setCallbackURL("https://localhost:8080/ms-callba

        //send document by email (optional)
        message.setCallbackMails("user1@mail.com,user2@mail.com'

        String messageCode = V3messagesApi.getInstance().sendMes
```

En el ejemplo anterior podemos ver como podemos indicar la url del servicio donde se avisará cuando el documento esté firmado y como configurar emails a los que enviar el documento firmado.

De todas formas siempre podemos esperar a que el documento sea firmado en el dispositivo, como podemos ver en el siguiente ejemplo.

```
int count = 100;
String status = null;
while (count > 0) {
    count--;
    Message msg = V3messagesApi.getInstance().getMes
    status = msg.getWorkflow().getCurrent();

    if ("RESPONSED".equals(status)) {
        Download download = V3documentsApi.getInstar
        byte[] pdf = IOUtils.toByteArray((new URL(do
        Assert.assertNotNull(pdf);
    } else {
        Thread.sleep(1000);
    }
}
```

# Nuevo documento desde plantilla

En el siguiente ejemplo puede ver como solicitar la generación de un documento, utilizando la configuración de firma definida en la plantilla seleccionada.

```
Message message = new Message();

// Create notification info
Notification notification = new Notification();
notification.setText("Title");
notification.setDetail("Detail");

// Find user device
Device device = V3devicesApi.getInstance().findDevic
notification.setDevices(new ArrayList<Device>());
notification.getDevices().add(device);
message.setNotification(notification);

Document document = new Document();
document.setTemplateCode(TEMPLATE_CODE);
document.setTemplateType(TEMPLATE_TYPE);
document.setItems(new ArrayList<Item>());

//GET Template info
Template template = V3templateApi.getInstance().finc

//Find items in template form
Form form = template.getForm();
for(Container container: form.getContainers()){
    for(Row row: container.getRows()){
        for(Item item: row.getItems()){
            if (item.getKey().equals("KEY_01")) {
                item.setValue("Jhon");
            } else if (item.getKey().equals("KEY_02'
                item.setValue("Doe");
            } else if (item.getKey().equals("KEY_03'
                item.setValue("11111111T");
```

```
                        }
                        //Add item to document
                        document.getItems().add(item);
                    }
                }
            }

            message.setDocument(document);

            //java example in https://github.com/viavansi/ms-cal
            message.setCallbackURL("https://localhost:8080/ms-ca

            //send document by email (optional)
            message.setCallbackMails("user1@mail.com,user2@mail.

            String messageCode = V3messagesApi.getInstance().ser
```

En el ejemplo anterior podemos ver como podemos indicar la url del servicio donde se avisará cuando el documento esté firmado y como configurar emails a los que enviar el documento firmado.

De todas formas siempre podemos esperar a que el documento sea firmado en el dispositivo, como podemos ver en el siguiente ejemplo.

```
int count = 100;
String status = null;
while (count > 0) {
    count--;
    Message msg = V3messagesApi.getInstance().getMes
    status = msg.getWorkflow().getCurrent();

    if ("RESPONSED".equals(status)) {
        Download download = V3documentsApi.getInstar
        byte[] pdf = IOUtils.toByteArray((new URL(dc
        Assert.assertNotNull(pdf);
    } else {
        Thread.sleep(1000);
    }
}
```

# Firma documento externo

En el siguiente ejemplo puede ver como solicitar la firmar de un documento previamente generado y disponible en una url, así como un ejemplo de configuración del dispositivo al que deseamos enviar la solicitud de firma.

```
Message message = new Message();

//Find device in user device list
Device device = null;
List<Device> devices = V3devicesApi.getInstance().fi
for (Device d: devices) {
    if (DEVICE_CODE.equals(d.getCode())) {
        device = d;
        break;
    }
}

// Create notification info
Notification notification = new Notification();
notification.setText("External document demo");
notification.setDetail("Sign a document available at
notification.setDevices(new ArrayList<Device>());
notification.getDevices().add(device);
message.setNotification(notification);

// Create a template document
Document document = new Document();
document.setTemplateReference("http://descargas.viav
document.setTemplateType(TemplateTypeEnum.url);
message.setDocument(document);

message.setPolicies(new ArrayList<Policy>());
Policy policy = new Policy();

policy.setEvidences(new ArrayList<Evidence>());
Evidence evidence = new Evidence();
evidence.setType(TypeEnum.SIGNATURE);
```

```
            evidence.setHelpText("User signature");
            evidence.setTypeFormatSign("XADES_B");
            policy.getEvidences().add(evidence);

            policy.setSignatures(new ArrayList<Signature>());
            Signature signature = new Signature();
            signature.setType(com.viafirma.documents.sdk.java.mo
            signature.setHelpText("Server signature");
            signature.setTypeFormatSign(com.viafirma.documents.s
            policy.getSignatures().add(signature);

            message.getPolicies().add(policy);

            //java example in https://github.com/viavansi/ms-cal
            message.setCallbackURL("https://localhost:8080/ms-ca

            //send document by email (optional)
            message.setCallbackMails("user1@mail.com,user2@mail.

            String messageCode = V3messagesApi.getInstance().ser
            Assert.assertNotNull(messageCode);
```

En el ejemplo anterior podemos ver como podemos indicar la url del servicio donde se avisará cuando el documento esté firmado y como configurar emails a los que enviar el documento firmado.

De todas formas siempre podemos esperar a que el documento sea firmado en el dispositivo, como podemos ver en el siguiente ejemplo.

```
int count = 100;
String status = null;
while (count > 0) {
    count--;
    Message msg = V3messagesApi.getInstance().getMes
    status = msg.getWorkflow().getCurrent();

    if ("RESPONSED".equals(status)) {
        Download download = V3documentsApi.getInstar
        byte[] pdf = IOUtils.toByteArray((new URL(do
        Assert.assertNotNull(pdf);
    } else {
        Thread.sleep(1000);
    }
}
```

# Consultar estado de la solicitud

En el siguiente ejemplo puede ver como consultar el estado o recuperar la información de una solicitud.

```
Message message = V3messagesApi.getInstance().getMes
```

# Descargar documento firmado

En el siguiente ejemplo puede ver descargar el documento firmado de una solitud terminada.

```
                    Download download = V3documentsApi.getInstar
                    byte[] pdf = IOUtils.toByteArray((new URL(dc
```

# Servlet para callback

En el proyecto https://github.com/viavansi/ms-callback puede consultar una implementación de ejemplo de un servlet en el que recibir la notificaciones de las solicitudes finalizadas.

```
private void doService(HttpServletRequest req, HttpServletRespor
    if(req.getParameter("message")!=null){
        try {
            ObjectMapper mapper = new ObjectMapper();
            mapper.configure(DeserializationFeature.FAIL_ON_UNKN
            Message message = mapper.readValue(req.getParameter(
            //check message status or download document
        } catch (Exception e) {
            log.log(Level.SEVERE,"Error parsing message JSON.",e
        }
    }else{
        log.log(Level.WARNING,"Empty request received.");
    }
}
```

En el ejemplo anterior extraido del proyecto de ejemplo puede ver como procesar las llamadas al servicio (servlet), creando un objeto mensaje que contiene toda la información referente a la petición y desde el que podría descargar el documento firmado, siguiendo los ejemplos de uso del sdk descritos en esta documentación.