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.

Integración en proyecto maven

Es necesario añadir el siguiente repositorio

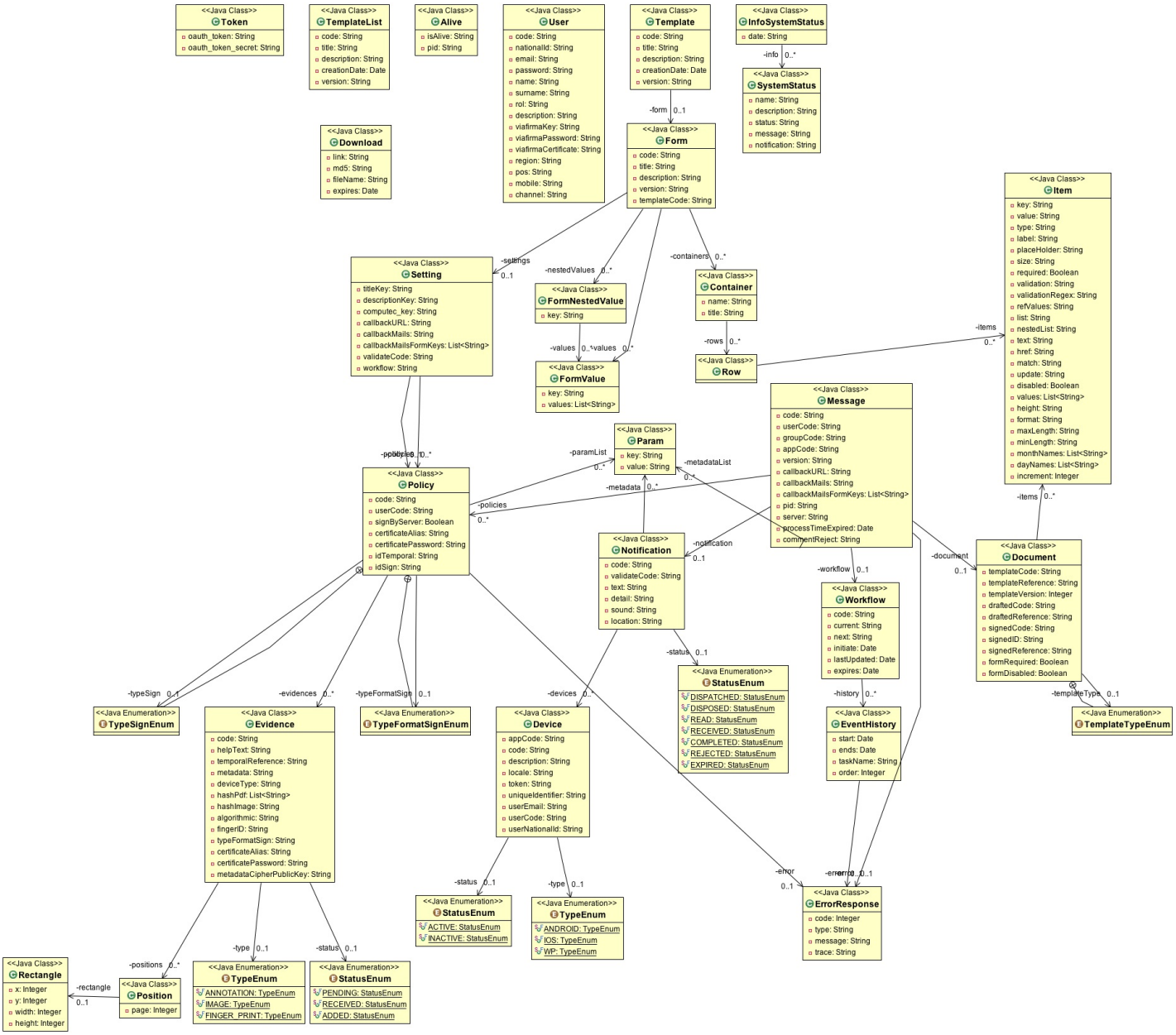
```
<repository>
    <id>viavansi-repo</id>
    <name>Viavansi Maven Repository</name>
    <url>http://repositorio.viavansi.com/repo</url>
</repository>
```

Añadir la siguiente dependencia

```
<dependency>
    <groupId>com.viafirma</groupId>
    <artifactId>ms-sdk-java</artifactId>
    <version>2.X.X</version>
</dependency>
```

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 I 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 I odt I 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 (ANDROIDIIOS)
- **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.

```
V1Api api = new V1Api();
api.setBasePath(API_URL);
api.setConsumerKey(CONSUMER_KEY);
api.setConsumerSecret(CONSUMER_SECRET);

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 = V1devicesApi.getInstance().findDeviceByUser(USER_CODE);
```

Información del usuario

En el siguiente ejemplo puede ver como recuperar la información de los usuarios registrados en el backend, este servicio le será de utilidad para mostrar información del usuario que está haciendo uso de la aplicación.

```
User user = V1usersApi.getInstance().findUserByCode(USER_CODE);
```

Listar plantillas

En el siguiente ejemplo puede ver como recuperar la información de las plantillas asignadas a un usuario en el backend, este servicio le será de utilidad para controlar los tipos de documentos que puede generar un determinado usuario.

```
List<TemplateList> templates = V1templateApi.getInstance().findTemplatesByUser(USER_CODE);
```

Información de una plantilla

Nuevo documento y envío al dispositivo

Nuevo documento desde plantilla

Descargar documento firmado

Compatibilidad SDK anterior a 2.5.X

Listar dispositivos utilizados por un usuario

```
List<Device> devices = api.findDeviceByUser(USER_CODE);
```

Recuperar información de un usuario configurada en el backend

```
User user = api.findUserByCode(USER_CODE);
```

Listar plantillas asignadas a un usuario

```
List<TemplateList> templates = api.findTemplatesByUser(USER_CODE);
```

Recuperar información de una plantilla configurada en el backend

```
Template template = api.findTemplateByCode(TEMPLATE_CODE);
```

Generación de un nuevo documento usando una plantilla configurada en el backend

```
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 mobile devices.
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-callback
message.setCallbackURL("https://localhost:8080/ms-callback/response");

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

messageCode = api.sendMessage(message);
```

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 = api.getMessageByCode(messageCode);
    status = msg.getWorkflow().getCurrent();

    if ("RESPONSED".equals(status)) {
        String documentCode = msg.getDocument().getDraftedCode();
        byte[] pdf = api.getDocument("temporal", messageCode, documentCode);
        Assert.assertNotNull(pdf);
    } else {
        Thread.sleep(1000);
    }
}
```

Generación de un nuevo documento y envío al dispositivo del usuario

```
List<Device> devices = api.findDeviceByUser(USER_CODE);

Device device = null;
for(Device cd: devices){
    if (cd.getCode().equals(DEVICE_CODE)) {
        device = cd;
    }
}
Assert.assertNotNull(devices);

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);

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.setTypeSign(TypeSignEnum.ATTACHED);
policy.setTypeFormatSign(TypeFormatSignEnum.DIGITALIZED_SIGN);
policy.setParamList(new ArrayList<Param>());

Param param01 = new Param();
param01.setKey("digitalizedSignHelpText");
param01.setValue("Firme Aquí");
policy.getParamList().add(param01);

Param param02 = new Param();
param02.setKey("fileName");
param02.setValue("example.pdf");
policy.getParamList().add(param02);

Param param03 = new Param();
param03.setKey("biometricAlias");
param03.setValue("viafirmadocuments");
policy.getParamList().add(param03);

Param param04 = new Param();
param04.setKey("biometricPass");
param04.setValue("12345");
policy.getParamList().add(param04);

Param param05 = new Param();
param05.setKey("op");
param05.setValue("pen");
policy.getParamList().add(param05);

Param param06 = new Param();
param06.setKey("signPositionEnable");
param06.setValue("true");
policy.getParamList().add(param06);

message.getPolicies().add(policy);

//java example in https://github.com/viavansi/ms-callback
message.setCallbackURL("https://localhost:8080/ms-callback/response");

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

String messageCode = api.sendMessage(message);
```

Generación de un nuevo documento y envío al dispositivo del usuario utilizando una plantilla

```
Message message = new Message();

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

// Find user device
Device device = api.findDeviceByUser(USER_CODE).get(0);
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 = api.findTemplateByCode(TEMPLATE_CODE);

//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);

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

//java example in https://github.com/viavansi/ms-callback
message.setCallbackURL("https://localhost:8080/ms-callback/response");

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

String messageCode = api.sendMessage(message);
```