

API

Im folgenden sind ein paar übliche Szenarien dargestellt, wie Kunden die Signatur Portal API nutzen. Alle Aufrufe an die API sind in der mitgelieferten Postman Collection wiederzufinden.

Dieses Dokument soll für einen besseren Einstieg zur REST API des Signatur Portal Servers sorgen. Dafür werden die Szenarien jeweils aus Nutzersicht, technischer Sicht und den zugehörigen API Aufrufen beschrieben.

Prefix und Voraussetzungen

Dieses Dokument sollte zusammen mit einer Postman Collection ausgehändigt sein. Alle in diesem Dokument enthaltenen Beispiele sind auch in der Postman Collection enthalten. Postman kann von [hier](#) bezogen werden. In Postman können die Requests ausgeführt und modifiziert werden. Ebenfalls kann Postman Code Snippets für die meisten geläufigen Sprachen generieren.

Die cURL Aufrufe in diesem Dokument müssen vorab modifiziert werden, damit die erwarteten Antworten erhalten werden. Grundvoraussetzung ist ein laufender Signatur Portal Server, sowie einen Application-Nutzer mit der Signatur Portal API Rolle. Bei allen Aufrufen, müssen die Server URL und der Authentifizierungs Token gesetzt sein. Bei Aufrufen die einen Nutzer Namen enthalten, muss der Nutzer im Server existieren.

Um die Postman Collection zu nutzen, muss diese in Postman importiert werden. Bevor Aufrufe in Postman ausgeführt werden können, muss den Collection Variablen Werte zugewiesen werden. Die Anleitung wie man diese setzt findet man [hier](#). Name und Passwort des Application-Nutzers und die URL mit der Endung `/rest/signatureportal/v1` müssen dort gesetzt werden. Wenn der Server als On-Premise Lösung läuft sollten die nötigen Werte bekannt sein. Andernfalls werden Ihnen die Werte von uns mitgeteilt.

1. Nutzer signiert selbst

Die Anwender, die in Ihrer Fachanwendung angemeldet sind, wollen selber ein Dokument signieren oder siegeln. Die sichtbare Signatur positioniert der Anwender im Signaturportal.

1.1 Nutzersicht

In der Fachanwendung wählt der Anwender die zu signierenden Dokumente aus und startet aus der Fachanwendung heraus eine Aktion zum Signieren der Dokumente. Daraufhin öffnet sich ein Tab in einem Web Browser mit dem Signatur Portal. Der Nutzer bekommt auf der Signatur Portal Website die zu signierenden Dokumente angezeigt und den Hinweis das er noch die Signatur positionieren muss.

Hinweis: Wenn es sich um eine qualifizierte Signatur handelt wird bspw. noch geprüft ob der Benutzer schon identifiziert ist für die Fernsignatur. Wenn nicht, dann bekommt er einen Hinweis, dass er das zunächst tun muss. Nachdem der Anwender die Signatur im Dokument positioniert hat und das Dokument signiert hat, wird dem Anwender eine Bestätigung über die eben durchgeführte Aktion angezeigt. Danach kann der Tab geschlossen werden und der Anwender kann zu der Fachanwendung zurückkehren. Dem Anwender stehen nun die signierten Dokumente in der Fachanwendung zur Verfügung.

1.2 Technische Sicht

Die Fachanwendung übergibt in einem einzigem API Aufruf an das Signatur Portal folgende Informationen:

- Mit welchem Signaturtyp signiert bzw. gesiegelt werden soll (z.B. einfache Signatur, fortgeschrittene Signatur, qualifizierte Signatur oder Siegel).
- Die Nutzer, die signieren sollen. Nutzernamen und Nutzer Typ.
- Die zu signierenden Dokumente (z.B ein pdf).
- Wie die Signaturen in den Dokumenten zu positionieren sind. Im einfachsten Falle kann der Nutzer die Position selbst festlegen.

Hinweis: Die Fachanwendung muss vorab den Nutzernamen kennen. Das ist der Nutzernamen mit dem sich ein Nutzer im Signatur Portal anmelden kann. Je nach Konfiguration des Signatur Portal Servers kann der Nutzernamen identisch mit dem Nutzernamen sein, mit dem sich ein Nutzer in der Fachanwendung anmeldet.

Der Server antwortet unter anderem mit einer PortfolioId und einer URL zum Signatur Portal, die die Fachanwendung in einem Webbrowser öffnet. Sobald der Signierprozess abgeschlossen ist, kann die Fachanwendung über einen weiteren API Aufruf die

Dokumente mittels der Portfolioid herunterladen und dem Anwender zur Verfügung stellen.

Eine Portfolioid identifiziert eindeutig ein Portfolio. Ein Portfolio ist die Dokumentenmappe die einem Workflow zugehörig ist. Workflows bestehen aus einer beliebigen Anzahl von Schritten. Jeder Schritt definiert eine Aktion für einen Unterzeichner/Prüfer, die ein oder mehrere Dokumente betreffen kann. Jede Signatur hat einen vordefinierten Signaturtyp und kann zusätzliche Identifikationsschritte über einen externen Signaturanbieter erfordern. Welche Dokumente sichtbar sind, kann für jeden Schritt konfiguriert werden. Auf diese Weise bleiben interne Dokumente intern, während nur explizit freigegebene Dokumente von externen Unterzeichnern gesehen und signiert werden können.

Aktuell muss die Fachanwendung den Status des Workflows regelmäßig abrufen (polling) um zu erfahren wann der Workflow erfolgreich abgeschlossen wurde und somit auch die signierten Dokumente zur Verfügung stehen.

Unmittelbar vor der Veröffentlichung steht aber auch die Möglichkeit, dass das Signatur Portal proaktiv der aufrufenden Anwendung „Bescheid“ gibt, wenn die Signierten Dokumente verfügbar sind. Dies wird mittels WebHooks umgesetzt.

1.3 API Aufrufe

In folgenden ein Beispiel mit minimalen Aufrufen der API die den oben genannten Prozess widerspiegeln:

Die cURL Beispiele können mit leichten Modifizierungen ausgeführt werden. Der Domänen Name `signaturportal-server.com` muss angepasst werden, sowie der auth header.

Die Zugangsdaten sind base64 codiert. Wenn man Zugangsdaten hat muss man `YOUR_BASE64_ENCODED_CREDENTIALS` mit dem Ergebnis folgenden Aufrufs ersetzen, dabei ist `username` und `password` anzupassen:

```
echo -n 'username:password' | base64
```

Der folgende Aufruf erstellt einen Workflow im Signatur Portal. In diesem Workflow soll ein im Signatur Portal registrierter Nutzer mit dem Namen `franz` das Dokument `beispiel.pdf` mit einer einfachen Signatur signieren. Damit der Aufruf gelingt muss der Nutzer `franz` auf dem Server existieren. Um den Aufruf zu testen, muss der Pfad zu dem pdf geändert werden.

```
curl --location \  
--request POST 'https://signaturportal-server.com/rest/signatureportal/  
v1/SPWorkflow/Start' \  
--header 'Content-Type: multipart/form-data' \  
--header 'Accept: application/json' \  

```

```

--header 'Authorization: Basic YOUR_BASE64_ENCODED_CREDENTIALS' \
--form 'filenames=@"/path/to/beispiel.pdf"' \
--form 'steps'=[{
  \"signers\": [{
    \"name\": \"franz\",
    \"signer_type\": \"user\"}],
  \"action\": \"simple-signature\"}]" \
--form 'sigposbysigner="true"'

```

Der Signatur Portal Server antwortet mit einem JSON Object:

```

{
  "portfolioid": 2241,
  "workflowid": 2360,
  "portfoliostate": 6,
  "portfoliostatename": "pending (6)",
  "workflowstepid": 2242,
  "workflowstepurl": "https://signaturportal-server.com/rest/signatureportal/v1/ui/SignatureEditor/Portfolio/2241/WorkflowStep/2242"
}

```

Gekürzte JSON Response

Die Anwendung öffnet die URL, die in `workflowstepurl` enthalten ist, damit der Anwender die Dokumente signieren kann. Daraufhin wird regelmäßig der Status des Portfolios gepollt, um zu erfahren wann die signierten Dokumente zum Download zur Verfügung stehen. Die `Portfolioid` wird aus der vorhergehenden Antwort des Server aus dem Feld `portfolioid` entnommen.

```

curl --location \
--request GET 'https://signaturportal-server.com/rest/signatureportal/v1/SPPortfolio/2241'\
--header 'Accept: application/json' \
--header 'Authorization: Basic YOUR_BASE64_ENCODED_CREDENTIALS'

```

Der Signatur Portal Server antwortet wieder mit einem JSON Object, das relativ viele Informationen zu dem Portfolio enthält. Die momentan einzig relevante Information ist dem Feld `portfoliostate`.

```

{
  "note": null,
  "workflowpresetname": null,
  "suppresswlfendmail": false,
  ...
  "datelocked": 1673606467764,
  "datecreated": 1673606357195,
  "portfoliostate": 3,
  "orgid": 1,
  "signerareordered": true,
  ...
  "workflowstep": null
}

```

Gekürzte JSON Response

Der Wert 3 in `portfoliostate` bedeutet, dass das Portfolio im Zustand `finished` ist und somit erfolgreich signiert wurde. Der Wert 6 steht für `pending` und bedeutet, dass der Workflow noch nicht abgeschlossen ist.

Sobald `portfoliostate` den Wert 3 annimmt kann die Fachanwendung, die Dokumente herunterladen. Es wird eine komprimierte Datei (zip) zurückgegeben, da in einem Portfolio mehrere Dokumente enthalten sein können.

```
curl --location \  
--request GET 'https://signaturportal-server.com/rest/signatureportal/v1/  
SPPortfolio/2241/Download' \  
--header 'Accept: application/zip' \  
--header 'Authorization: Basic YOUR_BASE64_ENCODED_CREDENTIALS'
```

2. Variationen

Im folgenden sind Variationen des vorhergehenden Szenarios dargestellt.

2.1 Anfrage einer Signatur

Ein Anwender, der in Ihrer Fachanwendung angemeldet ist, fragt eine Signatur einer anderen Person an. Die größten Unterschiede bei dieser Variation befinden sich in der Nutzersicht.

2.1.1 Nutzersicht

In der Fachanwendung wählt der Anwender die zu signierenden Dokumente aus und startet aus der Fachanwendung heraus eine Aktion zum Anfragen von Signaturen für die Dokumente. Der Anwender wählt eine Person aus, die die Dokumente signieren soll, sowie das entsprechende Signatur Niveau. Nun wird die Anfrage abgeschickt.

Die ausgewählte Person erhält daraufhin eine Mail, mit einem Link zum Signatur Portal und der Bitte die ausgewählten Dokumente zu signieren. Diese Person öffnet dann den Link aus der Mail heraus.

Daraufhin öffnet sich ein Tab in einem Web Browser mit dem Signatur Portal. Die Person bekommt auf der Signatur Portal Website die zu signierenden Dokumente angezeigt und den Hinweis, dass noch die Signatur positioniert werden muss. Nachdem die Person die Signatur im Dokument positioniert und das Dokument signiert hat, wird der Person eine Bestätigung über die eben durchgeführte Aktion angezeigt. Danach kann der Tab geschlossen werden.

Dem ursprünglichen Anwender stehen nun die signierten Dokumente in der Fachanwendung zur Verfügung.

2.1.2 Technische Sicht

Der Prozess ist nahezu identisch mit dem vorherigem Beispiel. Der einzige Unterschied ist, dass die Fachanwendung nicht die URL aus der Antwort des Servers öffnet.

2.1.3 API Aufrufe

Die API Aufrufe sind nahezu identisch mit dem vorhergehendem Beispiel. Es werden nur Werte aus dem Form Parameter `steps` angepasst. `signer_type` kann nur bestimmte Werte annehmen. Einfache Beispiele sind `user` und `external`. `user` bedeutet, dass die Person ein Account im Signatur Portal hat. `name` ist dann der Benutzername des Nutzers. `external` bedeutet, dass die Person keinen Account im Signatur Portal besitzt. `name` ist dem Fall eine Mail Adresse.

Der Form Parameter `steps` würde also folgendermaßen gestaltet sein in einem cURL Aufruf, für eine externe Person mit der Mail Adresse `name@beispiel.com`:

```
--form 'steps="[{"signers":[{"name":"name@beispiel.com", "signer_type":"external"}], "action":"simple-signature"}]" \
```

2.2 Einstellungen zur Signatur Position

Es gibt drei grundlegende Optionen wie die Position für eine Signatur in einem Dokument festgelegt wird.

- Der Anwender entscheidet selbst beim Signieren per drag-and-drop wo die Signatur im Dokument positioniert werden soll.
- Der Anwender legt die Position beim Erstellen des Workflows in einem Dokument fest.
- Es kann ein Suchbegriff festgelegt werden, nach dem im Dokument gesucht wird. Dazu wird eine Konfiguration benötigt, die angibt wo die Signatur Annotation im Verhältnis zur gefundenen Position platziert werden soll.

Der erste Fall wird bereits im ersten Beispiel genutzt. `sigposbysigner` hat den Wert `true`, das erlaubt den Anwender selbst die Signatur beim Signieren zu positionieren. Hat `sigposbysigner` den Wert `false` oder ist nicht in Aufruf enthalten, muss eine Signatur Position mitgeliefert werden im `steps` Parameter.

2.3 Anfrage von mehreren Signaturen

Workflows sind granular konfigurierbar. Oft werden Signaturen mit bestimmten Signatur Typen von unterschiedlichen Personen benötigt. In diesem Beispiel soll ein Dokument von zwei Personen signiert werden. Person A soll mit einer einfachen Signatur signieren. Danach soll Person B mit einer fortgeschritten Signatur signieren.

2.3.1 Nutzersicht

Wir nehmen an, dass der Workflow bereits auf irgendeinem Wege gestartet wurde. Zuerst erhält nur Person A eine Mail mit der Bitte ein Dokument zu signieren. Sobald das geschehen erhält Person B eine Mail mit der Bitte das Dokument zu signieren. Sobald alle geforderten Signaturen durchgeführt wurden, wird der Workflow Starter über den Workflow Abschluss informiert.

2.3.2 API Aufrufe

Die technische Sicht entfällt in diesem Szenario, da die Fachanwendung keine Berührungspunkte mit dem Signatur Portal Server hat, während der Workflow nicht abgeschlossen ist.

Als Beispiel wird der API Aufruf aus 1.3 modifiziert. Es wird dazu lediglich der Form Parameter `steps` angepasst. Der Übersicht halber wird der Parameter als JSON array dargestellt.

Person A entspricht dem Nutzer mit dem Namen hans. Person B heißt hier entsprechend franz. Pro benötigter Unterschrift wird also ein weiterer Schritt dem Workflow hinzugefügt.

```
[{
  "signers": [{
    "name": "hans",
    "signer_type": "user"
  }],
  "action": "simple-signature"
},
{
  "signers": [{
    "name": "franz",
    "signer_type": "user"
  }],
  "action": "advanced-signature"
}]
```

2.4 Hinzufügen von Workflow Manager

Es empfiehlt sich einen Workflow Manager zu einem Workflow hinzuzufügen. Ein Workflow Manager kann im API Aufruf mitgegeben oder im Signatur Portal beim Erstellen eines Templates/Workflows hinzugefügt werden. Ein Workflow Manager kann den zugehörigen Workflow inspizieren und bearbeiten.

2.4.1 API Aufruf

Ein Workflow Manager kann zum Beispiel beim Erstellen eines Workflows hinzugefügt werden. Dafür wird der API Aufruf aus 1.3 um den Form Parameter `portfoliomanagers` erweitert.

Der cURL Aufruf kann also um folgenden Parameter erweitert werden, damit der User mit Namen franz als Workflowmanager für den erzeugten Workflow gilt.

```
--form 'portfoliomanagers="[{"
  \"user_type\": \"user\",
  \"name\": \"franz\"
}]\"'
```

3. Templates

Alle hier beschriebenen Szenarien lassen sich auch mit Templates durchführen. Ein Template kann sowohl per API als auch im Signatur Portal direkt erstellt und konfiguriert werden. Templates lohnen sich besonders dann, wenn der gleiche Prozess häufiger durchgeführt wird, mit kleinen Variationen. Zum Beispiel eine Freigabe, die von mindestens zwei Personen freigegeben werden muss, die jeweils einer bestimmten Gruppe zugehörig sind. Dafür lässt sich ein Template konfigurieren. Der Anwender kann anhand des Templates und dem Dokument einen Workflow starten. Dies kann aus der Fachanwendung heraus aufgerufen werden. In diesem Szenario wird davon ausgegangen, dass bereits ein Template mit der ID 2246 existiert, in dem die signierende Person festgelegt ist.

3.1 Nutzersicht

Dieser Part ist nahezu identisch mit 2.1.1. In der Fachanwendung wählt der Anwender ein Dokument aus, das von zwei Personen freigegeben werden muss und startet über einen Button mit dem Namen Freigabe den Prozess. Sobald der Workflow abgeschlossen ist, wird der Anwender darüber per Mail informiert.

3.2 Technische Sicht

Um ein Template zu starten, ist je nach Template nur die Template ID nötig. In diesem Beispiel sendet die Fachanwendung die ausgewählte Datei und die Template ID an den Signatur Portal Server.

3.3 API Aufrufe

Der folgende API Aufruf startet einen Workflow auf einem Template basierend mit der templateid 2246. Die Id wird im URL Pfad mitgegeben. Es wird zusätzlich nur die zu signierende Datei mitgeschickt.

```
curl --location \
--request POST 'https://signaturportal-server.com/rest/signatureportal/
v1/SPWorkflow/2246/Start' \
--header 'Content-Type: multipart/form-data' \
--header 'Accept: application/json' \
--header 'Authorization: Basic YOUR_BASE64_ENCODED_CREDENTIALS' \
--form 'filenames=@"/path/to/beispiel.pdf"'
```

Der Server antwortet mit:

```
{
  "portfolioid": 2244,
  "workflowid": 2246,
```

```
"portfoliostate": 6,  
"portfoliostatename": "pending (6)",  
  ...  
"workflowstepid": 2245,  
"workflowstepurl": "https://signaturportal-server.com/rest/  
signatureportal/v1/ui/SignatureEditor/Portfolio/2244  
WorkflowStep/2245"  
}
```

Gekürzte JSON Response

Danach wird der Zustand des Portfolios gepollt, um anschließend die Dokumente herunterzuladen und in der Fachanwendung bereitzustellen.

4. Siegeln mit unsichtbarer Siegel Annotation

Der Anwender hat die Möglichkeit Dokumente zu signieren/siegeln ohne die Fachanwendung zu verlassen. Des Weiteren ist zum Beispiel beim Siegeln die visuelle Siegel Annotation auf dem Dokument optional. In diesem Szenario wird ein Dokument gesiegelt ohne eine visuelle Annotation im Dokument zu haben und ohne die Fachanwendung zu verlassen.

4.1 Nutzersicht

Der Anwender öffnet ein Kontextmenü oder ähnliches in der Fachanwendung auf einer Datei und wählt den Button Siegeln aus. Kurz darauf wird die Datei mit der gesiegelten Version ersetzt.

4.2 Technische Sicht

Die Fachanwendung schickt das Dokument in einem API Aufruf ans Signatur Portal zusammen mit ein paar Konfigurationsparameter und bekommt als Antwort die gesiegelte Datei entweder als zip oder als pdf wieder. Die Fachanwendung ersetzt die ursprüngliche Datei mit der gesiegelten Datei.

4.3 API Aufrufe

Es ist hier nur ein einziger API Aufruf nötig. Der folgende Aufruf schickt ein Dokument mit dem Namen sample.pdf an das Signatur Portal. Es ist zwingend eine SealConfigID nötig. Das ist die ID einer Siegel Konfiguration, die im Signatur Portal definiert werden muss. Darüber lässt sich bestimmen welcher Siegel Anbieter genutzt wird. Mit `responsetype` wird angegeben, ob man entweder eine einzelne pdf Datei zurück bekommen möchte oder ein zip. Die Flag `companystampvisible` gibt an, ob eine visuelle Siegel Annotation im Dokument erstellt werden soll. Wenn der Wert `true` ist, dann ist ein weiterer Parameter zur Position der Siegel Annotation nötig.

```
curl --location \
--request POST 'https://signaturportal-server.com/rest/signatureportal/
v1/SPSignature/Seal' \
--header 'Content-Type: multipart/form-data' \
--header 'Accept: application/zip' \
--header 'Authorization: Basic YOUR_BASE64_ENCODED_CREDENTIALS' \
--form 'filenames=@"/path/to/beispiel.pdf"' \
--form 'sealconfigid="1000"' \
--form 'responsetype="singlepdf"' \
--form 'companystampvisible="false"'
```

Als Antwort auf diesen Aufruf erhält die Fachanwendung, das gesiegelte Dokument als pdf.