



certillion

Manual de Integração

Sumário

1	Introdução.....	3
2	Terminologias.....	3
3	Links do projeto.....	3
3.1	WSDL de Assinatura.....	3
3.2	WSDL de Notificação.....	3
3.3	WS-Signer e AP-Samples.....	3
4	Autenticação.....	3
5	Serviços Disponibilizados pelo CSS.....	4
5.1	Descrição.....	4
5.2	Fluxos.....	4
5.2.1	Solicitação de Assinatura Síncrona.....	5
5.2.2	Solicitação de Assinatura Assíncrona Cliente – Servidor.....	6
5.3	Formato das Mensagens.....	7
5.3.1	SignatureSimpleDocumentReq.....	7
5.3.2	BatchSignatureComplexDocumentReq.....	7
5.3.3	SignatureTemplateReq.....	8
5.3.4	SignatureResp.....	8
5.3.5	SignatureStatusReq.....	9
5.3.6	SignatureStatusResp.....	9
5.3.7	ValidateReq.....	9
5.3.8	ValidateResp.....	9
5.3.9	BandwidthInfoType.....	10
5.4	Status.....	10
5.5	Erros.....	10
5.6	Serviços Adicionais.....	11
6	Solicitação de Assinatura Assíncrona Servidor – Servidor, a Abordagem Recomendada.....	11
6.1	Descrição.....	11
6.2	Fluxos.....	12
6.2.1	Solicitação de Assinatura Assíncrona Servidor - Servidor.....	12
6.3	Formato das Mensagens.....	13
6.3.1	SignatureNotificationReq.....	13
6.3.2	SignatureNotificationResp.....	13
6.4	Status.....	13
6.5	Erros.....	13
7	Exemplos.....	14
7.1	Pré-condições.....	14
7.2	Exemplos de solicitação de assinatura.....	14
7.2.1	Exemplo de assinatura de documento simples.....	14
7.2.2	Exemplo de assinatura de lote de documentos complexos.....	15
7.2.3	Exemplo de assinatura por template.....	16
7.3	Exemplo de validação de assinatura.....	16

1 Introdução

Este documento tem por objetivo auxiliar os desenvolvedores à realizarem a integração de suas soluções com o Web Service do Certillion Signature Service (CSS). Este Web Service possui os serviços necessários para que o Integrador possa realizar uma solicitação de assinatura a um cliente de forma prática e segura.

2 Terminologias

A fim de facilitar o entendimento deste documento, utilizaremos as seguintes siglas e terminologias:

- WS – Web Service
- CSS – Certillion Signature Service
- RSA e ECC – Algoritmos de criptografia assimétrica
- Integrador – empresa que integra um de seus serviços ao Certillion

3 Links do projeto

3.1 WSDL de Assinatura

<https://api.certillion.com/mss-ws/SignatureService/SignatureEndpointBean?wsdl>

3.2 WSDL de Notificação

<https://certillion.com/ap-ws/NotificationService/NotificationEndpoint.wsdl>

3.3 WS-Signer e AP-Samples

<https://certillion.com/download/integracao>

Este link disponibilizará o download do pacote de integração que contém em seu interior o proxy *WS-Signer* bem como seu manual e o projeto de exemplos AP-Samples.

4 Autenticação

O processo de autenticação de um Integrador frente ao CSS é realizado com um certificado digital ICP-Brasil de equipamento, emitido em nome de uma pessoa jurídica, no caso o Integrador. O certificado é utilizado para assinar todas as requisições SOAP utilizando o padrão WS-Security, para certificados X.509.

Para facilitar a implementação da assinatura das requisições, é disponibilizado um proxy de assinaturas, chamado WS-Signer, que recebe as requisições SOAP do software do Integrador, as assina utilizando o certificado ICP-Brasil de equipamento do Integrador, e repassa as requisições para o CSS.

Ressaltamos que apenas certificados ICP-Brasil são aceitos pela aplicação.

5 Serviços Disponibilizados pelo CSS

Esta seção descreve os serviços Web Services disponibilizados pelo CSS para o software do Integrador.

5.1 Descrição

A tabela abaixo contém um sumário dos serviços disponibilizados pelo CSS:

Nome do Serviço	Parâmetros	Retorno	Finalidade
SignatureSimpleDocument	SignatureSimpleDocumentReq	SignatureResp	Realizar a solicitação de assinatura de um documento simples a um usuário cadastrado no CSS. A solicitação pode ser síncrona ou assíncrona.
BatchSignatureComplexDocument	BatchSignatureComplexDocumentReq	SignatureResp	Realizar a solicitação de um lote de assinaturas de documentos complexos (PDF, XML, HTML, JPG,...) a um usuário cadastrado no CSS. A solicitação deve ser assíncrona.
SignatureByTemplate	SignatureTemplateReq	SignatureResp	Realizar a solicitação de assinatura a partir de um template a um usuário cadastrado no CSS. A solicitação pode ser síncrona ou assíncrona.
StatusQuery	SignatureStatusReq	SignatureStatusResp	Consultar o status de uma transação de assinatura realizada de forma assíncrona cliente-servidor.
Validate	ValidateReq	ValidateResp	Verificar a validade de uma assinatura.
FindBandwidthInfo	void	BandwidthInfoType	Consultar os dados de banda. Retorna o limite total, quantidade de bytes gastos e a data em que o limite será renovado.

5.2 Fluxos

Existem dois modos de assinatura: síncrona e assíncrona. No modo de assinatura síncrona, o Integrador realiza uma requisição de assinatura ao CSS e o mesmo envia a assinatura como resposta da requisição. No modo de assinatura assíncrona, o Integrador realiza uma requisição de assinatura ao CSS e o mesmo envia a assinatura em outra requisição. O modo assíncrono pode ser usado entre cliente-servidor, que será explicado nesse documento, e servidor-servidor, que será explicado na seção "Notificação".

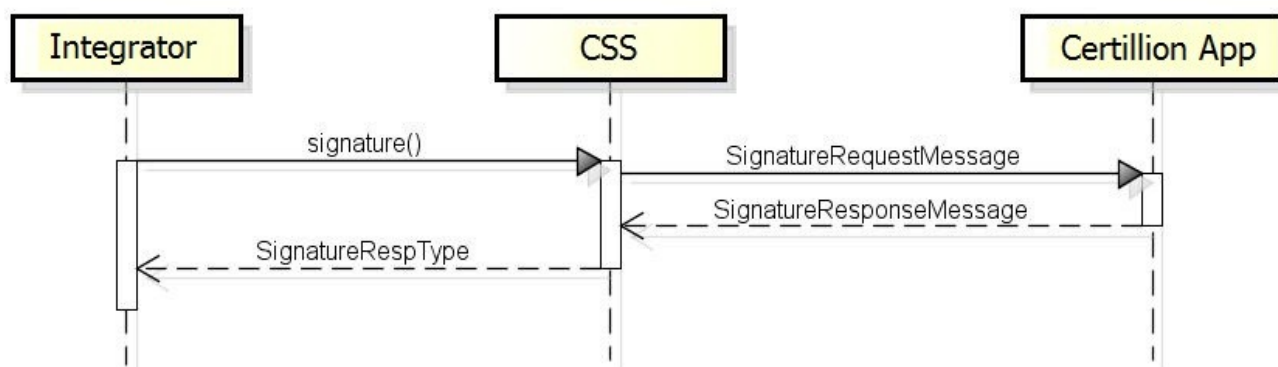
O Integrador pode solicitar a assinatura de três tipos de documentos, sendo eles:

- **Documento simples:** o Integrador usa essa funcionalidade para fazer requisições de assinatura em documentos simples que são textos puros. Para utilizar essa funcionalidade o Integrador deve usar o WS *SignatureSimpleDocument*;
- **Lote de Documentos:** o Integrador usa essa funcionalidade para fazer requisições de assinatura para um lote de documentos complexos, que podem ser em formatos PDF, XML, HTML, JPG, entre outros. Para utilizar essa funcionalidade o Integrador deve usar o WS *BatchSignatureComplexDocument*;
- **Template:** o Integrador usa essa funcionalidade para fazer requisições de assinatura usando um template. Para utilizar essa funcionalidade o Integrador deve usar o WS *SignatureByTemplate*;

Esses três tipos de requisição podem ser:

1. Síncrona;
2. Assíncrona cliente-servidor;
3. Assíncrona servidor-servidor (Serviço de Notificação), abordagem recomendada.

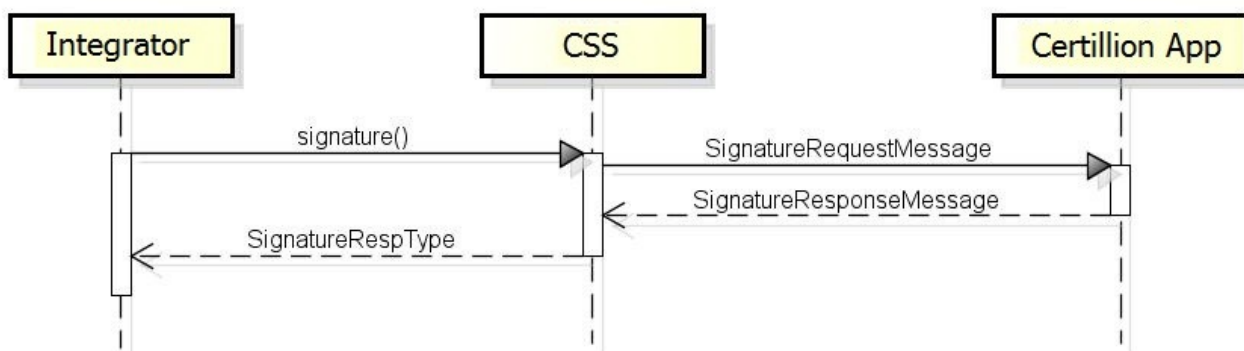
5.2.1 Solicitação de Assinatura Síncrona



1. O Integrador solicita a assinatura de um documento por intermédio do método de acordo com o tipo de assinatura que pode ser: WS *signatureSimpleDocument()*, WS *batchSignatureComplexDocument()* ou WS *signatureByTemplate()*;
2. O CSS realiza as seguintes verificações:
 1. Verifica se o Integrador solicitante é válido;
 2. Verifica se o usuário alvo da solicitação está cadastrado;
 3. Verifica se o usuário alvo da solicitação possui um certificado válido do tipo especificado pelo Integrador;
3. O CSS envia uma mensagem de solicitação de assinatura para a aplicação

- móvel;
4. A aplicação móvel assina o documento;
 5. A aplicação móvel envia a assinatura para o CSS;
 6. O CSS verifica a assinatura;
 7. O CSS informa ao Integrador que a assinatura foi realizada com sucesso.

5.2.2 Solicitação de Assinatura Assíncrona Cliente – Servidor



1. O Integrador solicita a assinatura de um documento de forma assíncrona pelo método de acordo com o tipo de assinatura que pode ser: WS `signatureSimpleDocument()`, WS `batchSignatureComplexDocument()` ou WS `signatureByTemplate()`;
2. O CSS realiza as seguintes verificações:
 1. Verifica se o Integrador solicitante é válido;
 2. Verifica se o usuário alvo da solicitação está cadastrado;
 3. Verifica se o usuário alvo da solicitação possui um certificado válido do tipo especificado pelo Integrador;
3. O CSS responde ao Integrador com o número da transação gerada (o sistema do Integrador deve guardar esse número pois ele deverá ser utilizado para consultar o status da transação);
4. O CSS envia uma mensagem de solicitação de assinatura para a aplicação móvel;
5. A aplicação móvel assina o documento;
6. A aplicação móvel envia a assinatura para o CSS;
7. O CSS verifica a assinatura;
8. O Integrador consulta se a transação já foi concluída pelo método WS

statusQuery();

9. O CSS processa a consulta e verifica se a solicitação já esta concluída.
10. Caso a solicitação de assinatura esteja concluída o CSS retorna a assinatura e seu status, caso contrário retorna o código de status 504;
11. O Integrador recebe a resposta de sua consulta e finaliza o processo caso a transação tenha sido concluída.

5.3 Formato das Mensagens

5.3.1 SignatureSimpleDocumentReq

Nome	Formato	Descrição	Obrigatório
MobileUser	MobileUserType	Identificador que permite que o CSS se comunique com o aplicativo mobile.	sim
DataToBeSigned	String	O dado que o Integrador deseja que o usuário assine.	sim
TimeOut	PositiveInteger	Representa o tempo em segundos que o CSS deve aguardar antes de responder ao Integrador. Se este argumento não for especificado, um TimeOut padrão será utilizado.	não
MessagingMode	MessagingModeType	Especifica se o modo da mensagem será síncrono, assíncrono cliente-servidor e assíncrono servidor-servidor.	sim
CertificateFilters	CertificateFiltersType	Especifica qual o tipo de certificado (cadeia de confiança, algoritmo e dono) que será aceito nesta transação.	não
AdditionalServices	AdditionalServiceType	Especifica se a transação faz uso de algum dos serviços adicionais (gasIdentificator, gasProtector, aetDriver);	não

5.3.2 BatchSignatureComplexDocumentReq

Nome	Formato	Descrição	Obrigatório
MobileUser	MobileUserType	Identificador que permite que o CSS se comunique com o aplicativo mobile.	sim
DocumentsToBeSigned	HashDocumentInfoType	Lista de documentos que o Integrador deseja que o usuário assine. Um objeto do tipo HashDocumentInfoType possui quatro parâmetros obrigatórios: ContentType (String), DocumentName (String), Url (String) e Hash (String). ;	sim
DataToBeDisplayed	String	A presença deste parâmetro indica que o DataToBeSigned é um dado não apresentável em texto puro (ex: um arquivo PDF). O conteúdo deste parâmetro será apresentado para o usuário.	sim
TimeOut	PositiveInteger	Representa o tempo em segundos que o CSS deve	não

Nome	Formato	Descrição	Obrigatório
		aguardar antes de responder ao Integrador. Se este argumento não for especificado, um TimeOut padrão será utilizado.	
MessagingMode	MessagingModeType	Especifica se o modo da mensagem será síncrono, assíncrono cliente-servidor e assíncrono servidor-servidor.	sim
CertificateFilters	CertificateFiltersType	Especifica qual o tipo de certificado (cadeia de confiança, algoritmo e dono) que será aceito nesta transação.	não
AdditionalServices	AdditionalServiceType	Especifica se a transação faz uso de algum dos serviços adicionais (gasIdentificator, gasProtector, aetDriver);	não
SignatureStandard	SignatureStandarType	Especifica o padrão da assinatura. Quando não especificado, utiliza-se o padrão CADES. Mas existe a opção do padrão de assinaturas ADOBEPDF	não

5.3.3 SignatureTemplateReq

Nome	Formato	Descrição	Obrigatório
MobileUser	MobileUserType	Identificador que permite que o CSS se comunique com o aplicativo mobile.	sim
TemplateToBeSigned	TemplateType	Informa qual é o template e os parâmetros que serão utilizados para montar o dado a ser assinado.	sim
TimeOut	PositiveInteger	Representa o tempo em segundos que o CSS deve aguardar antes de responder ao Integrador. Se este argumento não for especificado, um TimeOut padrão será utilizado.	não
MessagingMode	MessagingModeType	Especifica se o modo da mensagem será síncrono, assíncrono cliente-servidor e assíncrono servidor-servidor.	sim
CertificateFilters	CertificateFiltersType	Especifica qual o tipo de certificado (cadeia de confiança, algoritmo e dono) que será aceito nesta transação.	não
AdditionalServices	AdditionalServiceType	Especifica se a transação faz uso de algum dos serviços adicionais (gasIdentificator, gasProtector, aetDriver);	não

5.3.4 SignatureResp

Nome	Formato	Descrição	Obrigatório
TransactionId	MobileUserType	Número criado pelo CSS para esta transação	sim
Status	StatusType	Status atual da transação. REQUEST_OK (100) caso a requisição seja válida e aceita pelo CSS VALID_SIGNATURE (140) caso a assinatura for válida INVALID_SIGNATURE (700) caso a assinatura for	sim

Nome	Formato	Descrição	Obrigatório
		inválida	
Signature	SignatureType	Assinatura computada no aparelho móvel	não

5.3.5 SignatureStatusReq

Nome	Formato	Descrição	Obrigatório
TransactionId	Long	Neste parâmetro o Integrador menciona o número da transação da qual busca informações	sim

5.3.6 SignatureStatusResp

Nome	Formato	Descrição	Obrigatório
Signature	Base64Binary	Assinatura computada no aparelho móvel. Codificado em PKCS#7. (http://tools.ietf.org/html/rfc2315)	sim
Status	StatusType	Status atual da transação. Pode assumir todos os valores entre os códigos de status e de erros. No caso em que o método de status é chamado após uma requisição de assinatura assíncrona cliente-servidor, o Integrador receberá normalmente código de status e não de erro. O status típico para este método é o 110 TRANSACTION_IN_PROGRESS, que significa que a transação ainda não foi concluída e por consequência o Integrador deverá chamar o método de status novamente.	sim

5.3.7 ValidateReq

Nome	Formato	Descrição	Obrigatório
Signature	Base64Binary	O dado de uma assinatura PKCS#7 a ser validado	sim
Content	Base64Binary	Fornece o conteúdo que foi assinado caso a assinatura seja do tipo <i>detached</i>	não

5.3.8 ValidateResp

Nome	Formato	Descrição	Obrigatório
Error	ValidationErrorType	Este elemento estará presente apenas se a assinatura não puder ser verificada	não
Signatures	Lista de SignatureInfoType	Informações detalhadas de todas as assinaturas do documento. Este elemento estará presente apenas	não

Nome	Formato	Descrição	Obrigatório
		se a assinatura for verificada.	

5.3.9 BandwidthInfoType

Nome	Formato	Descrição	Obrigatório
totalBandwidth	Long	Limite total de banda contratada em bytes.	sim
usedBandwidth	Long	Volume de banda utilizada em bytes.	sim
dateToRenew	Date	Data em que a banda utilizada será zerada.	sim

5.4 Status

O CSS utiliza um conjunto de status para informar o resultado de operações diversas. Abaixo estão listados os status utilizados pelo WS de AP, bem como seus códigos e descrições.

Nome	Código	Descrição
REQUEST_OK	100	Requisição recebida com sucesso.
TRANSACTION_IN_PROGRESS	110	Transação ainda não processada. Retornada nos serviços de query (ex: CSSStatusQuery) quando a resposta da solicitação ainda não está disponível.
SIGNATURE_VALID	140	Assinatura válida.
CERTIFICATE_INVALID	600	Problema na verificação do certificado.
CERTIFICATE_REVOKED	604	Certificado revogado.
INVALID_SIGNATURE	700	Assinatura inválida.

5.5 Erros

Assim como os status, o CSS também utiliza um conjunto de código de erros para informar situações de inconformidade. O WS de AP utiliza os códigos de erro abaixo:

Nome	Código	Descrição
REQUEST_MISSING_PARAM	200	Retornado quando algum parâmetro da requisição não for encontrado.
REQUEST_WRONG_PARAM	201	Erro em algum parâmetro da requisição.
ACCESS_NOT_AUTHORIZED	220	A autenticação do requisitante falhou.
TRANSACTION_NOT_FOUND	310	A requisição não existe ou expirou.
IDENTIFIER_NOT_FOUND	311	O usuário final não foi encontrado. Por exemplo, no CSSFindUser.

Nome	Código	Descrição
MOBILE_SIGNATURE_ERROR	320	Erro geral de assinatura.
USER_CANCELED	400	O usuário cancelou a operação.
INTERNAL_ERROR	440	Erro geral do CSS.
CERTIFICATE_INVALID	600	Erro geral de verificação de certificado.
CERTIFICATE_NOT_FOUND	620	O usuário não possui um certificado do tipo especificado pelo Integrador.

5.6 Serviços Adicionais

O Integrador pode solicitar que as transações utilizem serviços adicionais de segurança. Os tipos de serviços de adições estão descritos abaixo:

Nome	Descrição
gasIdentificador	Fornecer um identificador único do dispositivo que garante a autenticidade do dispositivo.
gasProtector	Impede que aplicações maliciosas leiam dados sensíveis do aplicativo (anti-virus). (Ainda não suportado)
aetDriver	Permite acessar um hardware criptográfico para realizar assinaturas com certificados A3.

6 Solicitação de Assinatura Assíncrona Servidor – Servidor, a Abordagem Recomendada

Ficar bloqueado numa chamada síncrona ao CSS não é uma abordagem eficiente e envolve o problema de timeout. Por outro lado, utilizar a assinatura assíncrona resulta em ter de consultar periodicamente se a assinatura foi devolvida, o que pode levar a atrasos dependendo do intervalo a cada consulta.

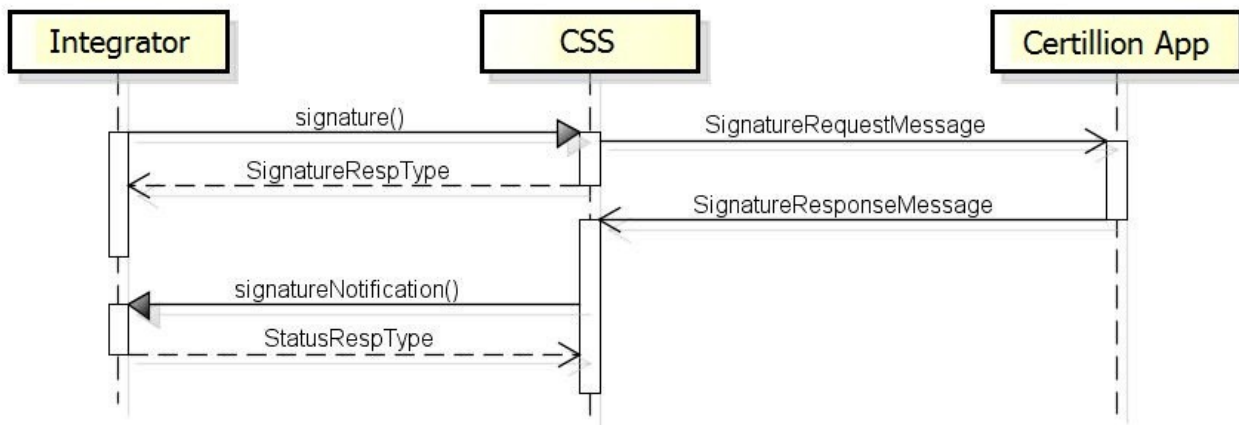
A abordagem recomendada é a solicitação assíncrona servidor-servidor, combinação da solicitação assíncrona e notificação do servidor. Assim que a assinatura for devolvida pelo usuário, o CSS notifica o serviço do Integrador, que pode obter a resposta através de uma consulta.

6.1 Descrição

Nome do Serviço	Parâmetros	Retorno	Finalidade
SignatureNotification	SignatureNotificationReq	SignatureNotificationResp	Notificar o Integrador da finalização da requisição de assinatura.

6.2 Fluxos

6.2.1 Solicitação de Assinatura Assíncrona Servidor - Servidor



O CSS disponibiliza um serviço de notificação quando a requisição de assinatura for concluída para facilitar a integração do Integrador.

1. o Integrador solicita a assinatura de um documento de forma assíncrona pelo método de acordo com o tipo de assinatura que pode ser: *WS signatureRequestMessage()*, *WS batchSignatureComplexDocument()* ou *WS signatureByTemplate()*;
2. O CSS realiza as seguintes verificações:
 1. Verifica se o Integrador solicitante é válida;
 2. Verifica se o usuário alvo da solicitação está cadastrado;
 3. Verifica se o usuário alvo da solicitação possui um certificado válido do tipo especificado pelo Integrador;
3. O CSS responde à AP com o número da transação gerada (o software de AP deve guardar esse número pois ele deverá ser utilizado para consultar o status da transação);
4. O CSS envia uma mensagem de solicitação de assinatura para a aplicação móvel;
5. A aplicação móvel assina o documento;
6. A aplicação móvel envia a assinatura para o CSS;
7. O CSS verifica a assinatura;
8. O CSS notifica o Integrador que a assinatura foi realizada com sucesso pelo método *WS signatureNotification()*;
9. o Integrador recebe a assinatura junto com status da transação.

6.3 Formato das Mensagens

6.3.1 SignatureNotificationReq

Nome	Formato	Descrição	Obrigatório
TransactionId	Long	Número criado pelo CSS para esta transação	sim
Status	StatusType	Status atual da transação. REQUEST_OK (100) caso a requisição seja válida e aceita pelo CSS SIGNATURE (500) caso o modo seja síncrono e o CSS possui uma assinatura realizada pelo usuário VALID_SIGNATURE (502) caso a assinatura for válida INVALID_SIGNATURE (503) caso a assinatura for inválida	sim
Signature	Base64Binary	Assinatura computada no aparelho móvel. Codificado em PKCS#7. (http://tools.ietf.org/html/rfc2315)	não

6.3.2 SignatureNotificationResp

Nome	Formato	Descrição	Obrigatório
Status	StatusType	Status atual da transação.	sim

6.4 Status

O CSS utiliza um conjunto de status para informar o resultado de operações diversas. Abaixo estão listados os status utilizados pelo WS de AP, bem como seus códigos e descrições.

Nome	Código	Descrição
REQUEST_OK	100	Requisição recebida com sucesso.
INVALID_CERTIFICATE	432	Problema na verificação do certificado.
SIGNATURE	500	Assinatura realizada.
REVOKED_CERTIFICATE	501	Certificado revogado.
VALID_SIGNATURE	502	Assinatura válida.
INVALID_SIGNATURE	503	Assinatura inválida.
OUTSTANDING_TRANSACTION	504	Transação ainda não processada. Retornada nos serviços de query (ex: mssStatusQuery) quando a resposta da solicitação ainda não está disponível.

6.5 Erros

Assim como os status, o CSS também utiliza um conjunto de código de erros para informar situações de inconformidade. O WS de AP utiliza os códigos de erro abaixo:

Nome	Código	Descrição
WRONG_PARAM	101	Erro em algum parâmetro da requisição.
MISSING_PARAM	102	Retornado quando algum parâmetro da requisição não for encontrado.
UNAUTHORIZED_ACCESS	104	A autenticação do requisitante falhou.
UNKNOWN_CLIENT	105	O usuário final não foi encontrado. Por exemplo, no mssFindUser.
EXPIRED_TRANSACTION	208	A requisição expirou.
USER_CANCEL	401	O usuário cancelou a operação.
MOBILE_SIGNATURE_ERROR	406	Erro geral de assinatura.
NO_CERT_FOUND	422	O usuário não possui um certificado do tipo especificado pelo Integrador.
ERROR_CERTIFICATE	425	Erro geral de verificação do certificado.
INTERNAL_ERROR	900	Erro geral do CSS.

7 Exemplos

Os exemplos do passo a passo dos fluxos são disponibilizados no projeto ICMP Sample – AP em conjunto com esta documentação. Para cada fluxo foram feitos exemplos dos três tipos de requisição de assinatura.

7.1 Pré-condições

Em todos os exemplos são necessários realizar os seguintes passos primeiramente:

1. Saber o endereço do servidor CSS;
2. Possuir um certificado de teste;
3. Configurar o WS Signer com o certificado de teste (Utilize o Manual WS-Signer.pdf);
4. Instalar e registrar o aplicativo no mobile;
5. Baixar o projeto "ICPM Sample – AP";
6. Instalar o projeto "ICPM Sample – AP" com as dependências necessárias.

7.2 Exemplos de solicitação de assinatura

7.2.1 Exemplo de assinatura de documento simples

Classe do fluxo de solicitação síncrona:

br.com.esec.icpm.sample.ap.SignatureSimpleDocumentSynchronousSample.java

Classe do fluxo de solicitação assíncrona cliente-servidor:

br.com.esec.icpm.sample.ap.SignatureSimpleDocumentAsynchronousSample.java

Comando para execução dos exemplos:

java -jar icpm-sample-ap.jar signature-sync <UniqueIdentifier> <TextToBeSigned>

java -jar icpm-sample-ap.jar signature-async <UniqueIdentifier> <TextToBeSigned>

Pré-condições do exemplo:

- Já possuir um certificado cadastrado no mobile;
- Saber o identificador do mobile que o certificado está cadastrado;

Argumentos que devem ser passados para sua execução:

1. UniqueIdentifier: identificador que permite que o CSS se comunique com o aplicativo mobile. Email em formato padrão. Ex: **usuario@esec.com.br**
2. TextToBeSigned: texto que o Integrador deseja que o usuário assine.

Ao final do exemplo, a assinatura bem como o texto a ser assinado são salvos no diretório local nos formatos **signature-<transactionId>.csr** e **data-<transactionId>.txt**.

7.2.2 Exemplo de assinatura de lote de documentos complexos

Classe do fluxo de solicitação assíncrona cliente-servidor:

br.com.esec.icpm.sample.ap.BatchSignatureAsynchronousSample.java

Comando para execução dos exemplos:

**java -jar icpm-sample-ap.jar batch-signature-async <UniqueIdentifier>
<DataToBeDisplayed> <standard> <url1> <url2> <url3...>**

Pré-condições do exemplo:

- Já possuir um certificado cadastrado no mobile;
- Saber o identificador do mobile que o certificado está cadastrado;

Argumentos que devem ser passados para sua execução:

1. UniqueIdentifier: identificador que permite que o CSS se comunique com o aplicativo mobile. Email em formato padrão. Ex: **usuario@esec.com.br**
2. DataToBeDisplayed: texto que deverá apresentado para o usuário.

3. Standard: padrão de assinatura: adobepdf ou cades;

4. Url: Url de destino dos arquivos a serem assinados. Ex:
http://certilion.com/example.pdf

*Ao final do exemplo, a assinatura é salva no diretório local no formato **signature-<transactionId>.csr**.*

7.2.3 Exemplo de assinatura por template

Classe do fluxo de solicitação síncrona:

br.com.esec.icpm.sample.ap.SignatureByTemplateSynchronousSample.java

Classe do fluxo de solicitação assíncrona cliente-servidor:

br.com.esec.icpm.sample.ap.SignatureByTemplateAsynchronousSample.java

Comando para execução dos exemplos:

java -jar icpm-sample-ap.jar signature-by-template-sync <UniqueIdentifier> <Text>

java -jar icpm-sample-ap.jar signature-by-template-async <UniqueIdentifier> <Text>

Pré-condições do exemplo:

- Já possuir um certificado cadastrado no mobile;
- Saber o identificador do mobile que o certificado está cadastrado;

Argumentos que devem ser passados para sua execução:

1. UniqueIdentifier: identificador que permite que o CSS se comunique com o aplicativo mobile. Email em formato padrão. Ex: **usuario@esec.com.br**
2. Text: texto que será passado como parâmetro do template número 1.

Ao final do exemplo, a assinatura bem como o texto a ser assinado são salvos no diretório local nos formatos **signature-<transactionId>.csr** e **data-<transactionId>.txt**.

7.3 Exemplo de validação de assinatura

Classe do fluxo de solicitação síncrona:

br.com.esec.icpm.sample.ap.ValidateSignatureSynchronousSample.java

Comando para execução do exemplo:

java -jar icpm-sample-ap.jar signature-validate <SignatureFile> <dataFile>

Pré-condições do exemplo:

- Possuir um arquivo no formato .p7s que contém uma ou mais assinaturas;
- Possuir um arquivo de dataToBeSigned, que contém o conteúdo assinado;

Argumentos que devem ser passados para sua execução:

1. SignatureFile: Nome do arquivo que contém as assinaturas a serem validadas;
2. dataFile: Nome do arquivo que contém o conteúdo assinado;