



الوكالة الوطنية للمصادقة الإلكترونية
Agence Nationale de Certification Electronique
National Agency For Digital Certification

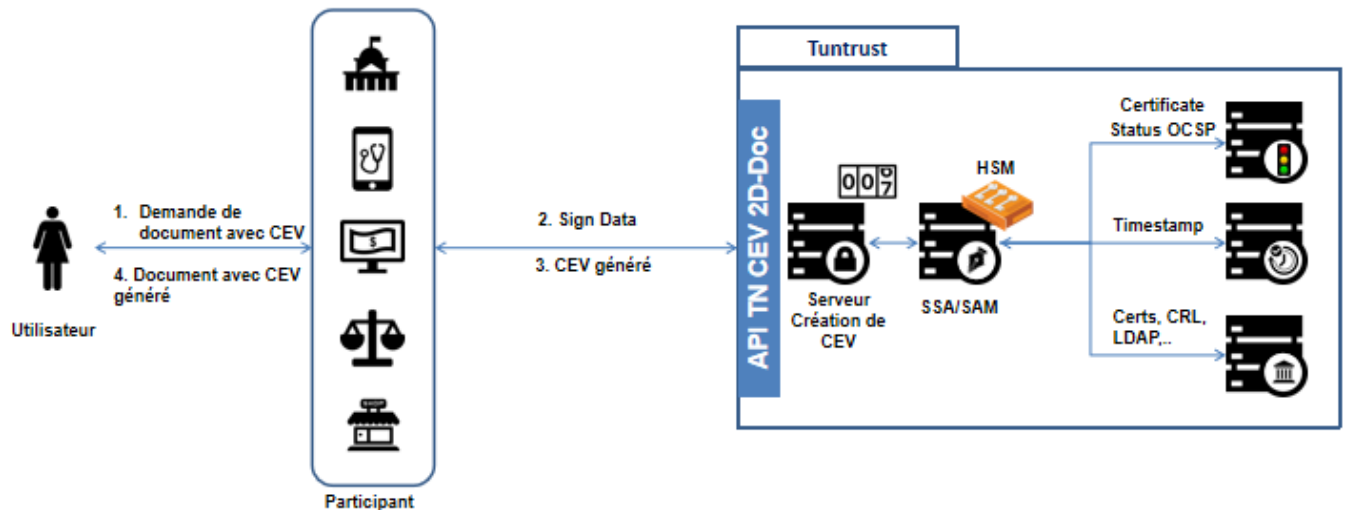
Documentation de l'API TN CEV 2D-Doc

Table des révisions

Version	Date	Nature de la révision	Section/Page
00	07/05/2021	Première rédaction	Toutes les pages
01	24/06/2021	Màj des URLs et liens hypertextes	Toutes les pages

1 Introduction

Le présent document permet de documenter la génération et la vérification d'un QR code d'un document à travers l'application QR-sign de Tuntrust selon le modèle de déploiement de l'API TN CEV 2D-Doc à consommer à partir de l'application du participant CEV selon le schéma ci dessous:



Après avoir rempli le formulaire du participant communiqué par Tuntrust, une configuration sera faite sur la plateforme CEV. Cette configuration couvre les points suivants:

- La création du Participant et de son certificat cachet électronique 2D-DOC.
- La création du document et de ses différents champs (obligatoires/ optionnels).
- L'association entre les différents éléments précédents pour pouvoir générer les cachets électroniques visibles.

2 Web Service de création de code bidimensionnel :

Cette partie décrit l'utilisation du web service SOAP permettant de créer un Cachet Électronique Visible pour un type de document donné à partir de l'Url du web service de génération de CEV :

- «<http://cev.tuntrust.tn/TuntrustCevTest/signatureWs>» : pour générer des CEV de test
- «<http://cev.tuntrust.tn/TuntrustCev/signatureWs>» : pour générer des CEV de production

Le format d'**Entrée** est de type XML (participant, codePin, alias, type de document, values)

Le format de **Sortie** est de type XML (code d'erreur, message, image, chaine cev)

Le WSDL du webservice est disponible via cette url :

- test: « http://cev.tuntrust.tn/TuntrustCevTest/signatureWs»
- Production: «<http://cev.tuntrust.tn/TuntrustCev/signatureWs>»

2.a Entrées du web service

L'élément d'Entrée du web service Soap est comme suit :

Désignation	Type	Obligatoire	Description
participant	chaine	Oui	Matricule fiscale du participant.
codePin	chaine	Oui	Code PIN du certificat cachet.
alias	chaine	Oui	Alias du certificat cachet.
sessionId	chaine	Non	Session Id: ce paramètre est utilisé lorsque le client se propose d'activer la signature par cachet électronique par une personne physique à travers son certificat Digigo. Le paramètre sessionId résulte d'une authentification par certificat Digigo de la personne responsable en saisissant son e-mail, son code pin et le code de vérification reçu par sms (otp)).
typeDoc	chaine	Oui	Code du type du document.
values	chaine	Non	Les champs du document (code du champ & valeur).

=> Pour le cas d'un certificat cachet autorisé par une personne physique ayant un certificat Digigo, une documentation de web services supplémentaires permettant l'obtention du paramètre «sessionId» sera communiquée ultérieurement.

L'élément sessionId fait référence à un numéro de session valide généré suite à une authentification avec le certificat électronique DigiGO. Cet élément est optionnel et ne sera utilisé que si la demande de génération du QR-code se fait suite à une intervention humaine.

Cet élément permet de garder une trace sur la personne qui est entrain de lancer une requête de génération de QR.

Si la demande est générée automatiquement par le serveur alors l'élément <sessionId> ne sera pas utilisé.

Exemple d'entrée :

Cet exemple traite le cas du participant ayant le numéro de MF 147147, signant avec le certificat ayant l'alias AnceCevTest14 et le code PIN 833811, il va générer un document Diplôme (Code L4) possédant 14 champs avec des codes (code LJ, code 03, .. , code LO) et les valeurs (1, Mohamed Ben Foulen.....,1)

--> Cette requête permettra de créer un QR code signant les valeurs envoyées relatives aux 14 champs du diplôme.



2.b Sorties du web service

L'élément de retour du web service Soap est comme suit :

Désignation	Type	Taille	Description
Code Error	chaîne	2	Code d'erreur
Message	chaîne	255	Message explicatif

Cev	chaîne	255	Chaîne CEV (si l'opération a réussi)
Image Cev	chaîne	255	Image CEV encodée en base 64 (si l'opération a réussi)

Exemple de réponse (QR code généré avec succès) :

```
<?xml version='1.0' encoding='UTF-8'?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body><ns2:signatureWsResponse
xmlns:ns2="http://ws.soap.progress.com/">
<return>
<cev>DC04000101611E4A1E4AL01TN4E3....</cev>
<codeError>00</codeError>
<imageCev>iVBORw0KGgoAAAANSUhEUgAAAJAAACoCAIAAACADfAvAAAFo
kLEQVR42u2b21LEMAxD+f+fhldmdiiW.....D=</imageCev>
<message>Success</message>
</return>
</ns2:signatureWsResponse>
</S:Body>
</S:Envelope>
```

2.c Codes d'erreur :

La liste ci-dessous donne tous les codes d'erreurs possibles suite à l'exécution du web service soap:

- 1 → If participant null → code : 07 , message : Participant is required !
- 2 → If participant non trouvé dans la BD → code : 06 , message : Participant not found !
- 3 → If participant n'a pas des cachets → code : 09 , message : Participant has no stamp !
- 4 → If typeDoc null → code : 05 , message : Type document is required !
- 5 → If typeDoc non trouvé dans la BD → code : 04 , message : Document type not found !
- 6 → If participant n'a pas l'autorisation pour ce type de document → code : 10 , message : Document type not authorized to the participant!
- 7 → If alias null → code : 12 , message : Alias is required !
- 8 → If codePin non trouvé dans la BD → code : 13 , message : Stamp not found !
- 9 → if cachet n'est pas affecté a cet participant → code : 14 , message : The stamp is not assigned to this participant !
- 10 → If codePin (cachet) est désactivé → code : 15 , message : The stamp is deactivated !
- 11 → If les quotas sont épuisés → code : 16 , message : The quota of the stamp is exhausted!

12	If la validité du document > date d'expiration du cachet → code : 17 , message : The validity of the stamp must be greater than the validity of the document !
13	If pas d'association entre document et le cachet → code : 03 , message : No stamp available for this type of document !
14	Check <u>Donnees</u> (nombre des champs entrées , taille min , max , ..) → code : 01 , message : Non-compliant data ! !
15	If il ya un erreur au moment de <u>generation</u> du CEV → code : 08 , message : Error in Web Service !
16	If alias ou code pin incorrecte (appel <u>ws</u> signature) → code : 403 , message : The alias is not valid or the password is wrong !
18	If <u>error</u> in <u>ws</u> signature → code : 404 , message : Error while calling Signature Service !
19	If <u>success</u> → code : 00 , message : Success , <u>cev</u> + image

2.d Exemples de retour du WS

Exemple 1 : Cachet non valide

```

Raw XML
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:signatureWsResponse xmlns:ns2="http://ws.soap.progress.com/">
      <return>
        <codeError>03</codeError>
        <message>No stamp available for this type of document !</message>
      </return>
    </ns2:signatureWsResponse>
  </S:Body>
</S:Envelope>

```

Exemple 2 : Code Document non trouvé

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:signatureWsResponse xmlns:ns2="http://ws.soap.progress.com/">
      <return>
        <codeError>04</codeError>
        <message>Document type not found !</message>
      </return>
    </ns2:signatureWsResponse>
  </S:Body>

```

```
</S:Envelope>
```

Exemple 3 : Champ vide

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:signatureWsResponse xmlns:ns2="http://ws.soap.progress.com/">
      <return>
        <codeError>01</codeError>
        <message>Non-compliant data!</message>
      </return>
    </ns2:signatureWsResponse>
  </S:Body>
</S:Envelope>
```

Exemple 4 : Cachet valide

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:signatureWsResponse xmlns:ns2="http://ws.soap.progress.com/">
      <return>
        <cev>
          DC04000100241C6C1C6C0701TN37310854726962756E616C37320333235373305313235343837340331343237350A53616D6920536173736937360E4E
          6F6D20636F6D6E6726369616C3737094E6F6D206C6174696E637380861646472657373653739013138300454657374CMEQCIGnSq9dpTLMdCyn
          792p2tKb8NKEvpD9YTgIQ/igi2FFpAiBLtmvDBTLuVA2sQEswK0r5WIZKFvRpmJQEpgdhTrwEQ==
        </cev>
        <codeError>00</codeError>
        <imageCev>
          iVBORw0KGgoAAAANSUHuUgAAAJAAACoCAIAAACADfAvAAAFq01EQVR42u2d0W7rMAxD+/8/vb0OKNqJpJzE8tFT0LVp4uMqFKWL+/ohtooXSvCw2hf/iU+vf3p
          P5Zz1z9bf710VwAB2B6r81eT86uvJ9QAMYLc/SOM17sLwNDwAAxjAAAaw5wJL5MD3wgBgADsb2PvSI+sBBBrBuU+o9re1iTR2UEgG2vawnAAYwAmBEJ7BPspvja4
          4Bdh6wRChXLKXv60I7vxqK6bX9/PKPlu3EQA2HVhi0kQ//8KN1VONutwemPrqmWU+wEYBU22YZExMRVg3t7yt8wqisqW8NQfYLGDFU1N1abz19tKaKmlUPBWp4
          j0sFhbOAHs0ME9QeA/buuRVGzTeVlPPnH+qQdYDbDNgaJg9bLm5b1aXCeb12N4AJsFrGuUTE2DnrDOzS1vK6j1cCJGADYRmPpz9h6tatHt2am59E9KYLWUBth0
          YMnPB04XRZqvBd8f76t1BXBmAtGeV1rGo7qW12FdN5sawmzFzQA2wiMDX1eU1Cr2xfUW7nRpp6hQt1PcA2AObdamId1Qvh/FP1x0sN5FRQLZnpANjGwBIM6mN
          c3S5eCyZPkiEh1kAAAGwgsC7b10u5yThCHWpuKSVF9JJ+GMA2AEbcFQA7AxjHdx0DjH8f9q9NGRg8SQJRk4xaequmndeEAAtgZwOrmZt62zwdy8qVRhwDUVo43eA
          Ow6cC60kKSBtWS3Cuuk1aLL/C8oQ4D2IOAeWZPYgKZoylWqzC5d0/UqCY7wM4Alg+fqLJbNW2ToSClmaJecz6OB7C5wFSBng/zdF1DBZWX/FXR74kXgJ0BrL64i
          R1VKTzVNmBXO7G32KjLM4CdCkwtQpORgnzMPi7ivVZI7+a7SNYD7HHAktQDeLcuvUWcYU5p5pnS6wpgG0JzEtuXur7CvPJkuJ9c7jbV+AzQXmFafqpa8owLte
          UG/rx9tqADsDWF5iJ0nPu6okjXt3raZW+b4ANhCYKsRz0e9Z2R/mS5YKoDkO1JgA2F5j6xetS1jUDDUn7JikSSkYawAYCI64PgJ0Bj007jgF2ErDkIa9aRMkQgDe
          U4I3feEVIvaEKsFOBJeap2phFw0e9SWkpvK28Bq2ADsJmLqsatGat9vrpaua+ryUWG/ByAkZYKOA1ZdeLgCbGvN1ePKrVx8BXmI0UzHABgLR1SR1G1RFRgoBdS
          QiGc6p/xiWjAgA7KHA1Id5PtKTS3wvXKuFRF7qLPQSAbYNNMNVKUUVtYhQ191LyV1X0q1I1YGcAqxeH3oM6L81Vc2GM3KhLrxpgzXUYwDYAlieLyXW/+qVrmohr
          74zb4B7AxgXULWg+Q1K9Wi7Sok83dbE0BbANGassuaV16R1RSJqvFw4qEL6dTgI0C1rWU6oPKK5DzFqVqfanGnjo6ALDzgOXGqDqG5o0FrDPAugzoilIiwB4N
          LH1QJ6VxV8pKQHbB8Ax0gJ0BjLgrAHYGMi7vOgYY/3+YClhtYdQtV29AQW4hNo3pmTYFwEYB8xoT9YXz1smT9UkbSC0GvMGFNlkPsM2AJUZt3jxUE1eysdQtmI/
          ryHYEwIYDS9r8udGVj6c1UqW3UX1RSgTYBsCS4RzVgPFEv9ovVO03r2GrN1eb2T3ANgDW+4hWRU3SmPaa86pwT8Yg1owIAGxLYIMUVwVuUmwmo3Delk02nFm0AG
          wUMM+wSYr13vPn2b2nZNeFlYoCYLOAg2LUk8sry15VdCSNfK8Z2N41wAYCU0fVkmLbawzW2x+e0F/xjUusKYBtBswTC3mSVJfAs60TAzcp81V7D2DTgUaM3lq6r
          KkksaN2jrxhnmMh1YrMmW6ErGarjC94V5iNAF816gD0UWN2mTG7bKyGS9r9qKKtbJ0nAdSDGHFAGw6MOK2ATCAEQAJAAYwAmAEwABGAIwAGMAIgBEAAxgBMA
          JgACMARgDsiPgF4NGAAFU00wUAAAAAUVORK5CYII=
        </imageCev>
        <message>Success</message>
      </return>
    </ns2:signatureWsResponse>
  </S:Body>
</S:Envelope>
```

2.e Décodage du QR code généré

Le décodage de la chaîne CEV ou bien de l'image CEV retournées par le web service de génération peut se faire en utilisant par exemple les liens web:

- Plateforme de test : <http://cev.tuntrust.tn/TuntrustCevTest/verify.jsf>
- Plateforme de production : <http://cev.tuntrust.tn/TuntrustCev/verify.jsf>

La plateforme de vérification web fournira les informations suivantes :



Diplome

Signer : AnceCevTest14

Date emission du cachet : 2020-11-04

Date Signature : 2020-11-04

Type diplome : Diplôme National d'Ingénieur

Nom et Prénom :

CIN/Passeport : 0721

Date de naissance : 10/10/1995

Lieu de naissance : Tunis

Nationalite : TUNISIENNE

Specialite : Télécommunications

Session : juin 2020

Institution universitaire : E

Numéro diplôme : 0

3 Web service de vérification du code bidimensionnel

Le web service de vérification permet de décoder et vérifier les cachets électroniques visibles (QR Code signé).

L'Url du web service de vérification :

- Plateforme de test: <http://cev.tuntrust.tn/TuntrustCevTest/rest/validService/validateDoc>
- Plateforme de production: <http://cev.tuntrust.tn/TuntrustCev/rest/validService/validateDoc>

3.a Format d'Entrée : JSON (valeur de la chaine CEV)

L'élément d'Entrée est comme suit :

Désignation	Type	Obligatoire	Description
value	chaîne	oui	Valeur de la chaîne CEV obtenue lors de la génération du QR code. C'est la valeur de chaîne DC à valider

Exemple d'entrée:

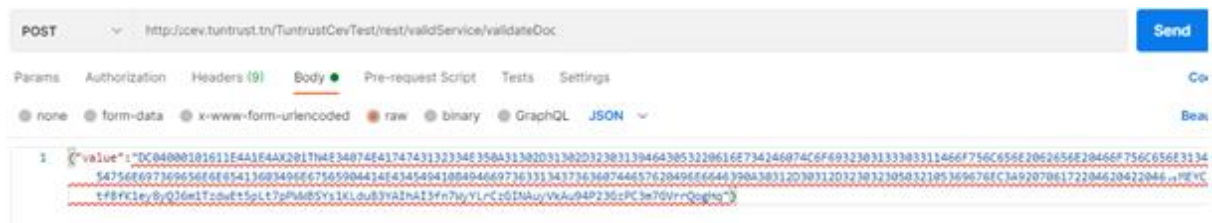
POST /TuntrustCevTest/rest/validService/validateDoc HTTP/1.1

Host: cev.tuntrust.tn

Content-Type: application/json

Content-Length: 415

```
{"value": "DC04000101611E4A1E4AX201TN4E34074E4174743132334E350A31302D31302D323031394643053220616E734246074C6F6932303133303311466F756C656E2062656E20466F756C656E31340A54756E697369656E6E65413603496E67565904414E4345494108494669736331343736360744657620496E6646390A30312D30312D323032305032105369676EC3A920706172204620422046-MEYCIQCtF8fK1ey8yQJ6m1TzdwEt5plT7pPWBSYs1KLdu83YAIhAI3fn7WyYLRcZGINAuyVKAu94P23GzPC3m7GVrrQogHq"}
```



3.b Format de sortie : JSON

Le résultat de l'exécution du web service donne :

- Le résultat global
- La décomposition de la chaîne du QR code
- Le résultat de vérification du cachet

L'élément de retour est comme suit :

Désignation	Type	Description
Résultat	Boolean	Résultat de décomposition de la chaine et de vérification de la signature du document
errorId	entier	Code d'erreur
Error	Chaine	Message d'erreur
verifCertif	Boolean	Résultat de vérification du cachet de signature
Details	Array de json	Détails du document, les différents champs du QR code en spécifiant pour chaque champ :id, label, value, type
dateSign	json	Date de signature (id, label, value)
Signer	json	Le signataire (id=identifiant du certificat, label, value)
dateCert	json	Date de création du QR code
document	json	Détail sur le document (id, label, value)
docstyle	json	Langue du document (ar, fr)

Exemple de sortie

```
{
  "verifCertif": true, "result": true, "errorId": "0",
  "docstyle": { "id": "ZZ", "label": "language", "value": "ar" },
  "document": { "id": "X2", "label": "الوثيقة نوع", "value": "شغل عقد لتأشيرة الخضوع عدم في شهادة",
  "dateCert": { "id": "00", "value": "2021-03-25", "label": "إصدار تاريخ الختم الإلكتروني",
  "signer": { "id": "0161", "label": "إمضاء", "value": "FormatCevTest9" },
  "dateSign": { "id": "00", "label": "الإمضاء تاريخ", "value": "2021-03-25" },
  "titre1": { "id": "00", "label": "titre 1", "value": "" },
  "titre2": { "id": "00", "label": "titre 2", "value": "" },
  "details": [ { "id": "N4", "label": "الشهادة رقم", "value": "NAtt123", "type": "String"}, { "id": "N5",
  "label": "الشهادة تاريخ", "value": "10-10-2019", "type": "String" }, { "id": "FC", "label": "الصلاحية مدة",
  "value": "2 ans", "type": "String"},
  { "id": "BF", "label": "القانون", "value": "Loi2013", "type": "String"}]
```

3.c Codes d'erreurs

Les codes d'erreurs sont les suivants :

ErrorId	Message de l'erreur	Description
101	Invalid input !!	La valeur de la chaine entrée est vide
102	Invalid input DC !!	La chaine du QR code ne commence pas par DC
103	Invalid input parsing !!	Problème de décomposition
501	chaine invalide !!	La longueur de la chaine est très courte (< 24 caractères)
502	version 2d doc invalide : version	La version de 2D-DOC n'est pas correcte.
503	type document invalide	Le code du document n'est pas correct.
504	date certificat invalide	Le format de la date du certificat n'est pas correct.
505	date signature invalide	Le format de la date de signature du QR code n'est pas correct.
506	Signer invalide	Si le champs signataire est incorrect