

API !getPostPaymentStatus

Live URL:

<https://secure.trust-provider.com/products/!getPostPaymentStatus>

Version History

1.00 Original version.

This API allows web host partners to request a detailed post-payment page state for their orders. This information can be used by the web host to generate this kind of page himself.

1. Request

The request should be POSTed (rather than GETed) to the mentioned above URL.

Required variables are in **bold**.

Optional variables are in *italics*.

Variable Name (case insensitive)	Type	Max. Length	Allowed Values	Description
loginName	string	64 chars		Account Username (case sensitive)
loginPassword	string	128 chars		Account Password (case sensitive)
orderNumber	string	64 chars		The Order Number of a certificate for which it's needed to get the Post Payment page status
<i>certificateID</i>	Number	16 digits		SASP certificate ID
<i>saspSessionID</i>	String	16 chars		SASP session handle. If this parameter is specified, then loginName and loginPassword parameters should be omitted

2. Response

Response Schema: application/json; charset=UTF-8

Name of parameter is in **bold**.

Type of parameter is in *italics*.

JSON response:

errorCode	<i>integer</i>	Error Code	Possible values: see section 2.1
errorMessage	<i>string</i>	Description of error	
data	<i>Object (PostPaymentData)</i>	Contains description of statuses	

Object (PostPaymentData)

orderNumber	<i>string</i>	SASP order number	
productType	<i>integer</i>	SASP object type ID	
productName	<i>string</i>	SASP product name	
certificateDuration	<i>integer</i>	Certificate duration, in days	
productTerm	<i>integer</i>	Product term, in days	
uploadDocumentUrl	<i>string</i>	URL for uploading documents of order	See section 3

uploadDocumentToken	<i>string</i>	JWT token for upload file	See section 3
orderStatus	<i>string</i>	Order status	Enum: 'Current', 'Awaiting Payment', 'Payment Promised', 'Pending', 'Awaiting Authorization', 'Processed', 'Rejected', 'Refunded', 'ERROR!!!'
domainName	<i>string</i>	For single domain certificate contains common name. For MDC contains static text 'Multiple domains'	
primaryDomainName	<i>string</i>	Primary domain name	indicated for MDC only
maxMDCDomains	<i>integer</i>	Maximum number of domains that a certificate can contain	indicated for MDC only
maxMDCWildcardDomains	<i>integer</i>	Maximum number of wildcard domains that a certificate can contain	indicated for MDC only
csrStatus	<i>Object (csrStatus)</i>	CSR status detail	
dcvStatus	<i>Object (dcvStatus)</i>	DCV status detail	
agreementStatus	<i>Object (agreementStatus)</i>	Agreement Execution status detail	
callbackStatus	<i>Object (callbackStatus)</i>	Callback status detail	
organizationStatus	<i>Object (organizationStatus)</i>	Organization validation status detail	
secondApprovalStatus	<i>Object (secondApproval)</i>	Second Approval status for EV order	
fsupStatus	<i>Object (fsupStatus)</i>	Fast Service status	

Object (csrStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'.
description	<i>string</i>	Text of description	
availableActions	<i>Array of string</i>	Available actions on SASP side for current status	Enum: 'needProvideCSR', 'canReplaceCSR'

Object (dcvStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'.
description	<i>string</i>	Text of description	
availableActions	<i>Array of string</i>	Available actions on the SASP side for current status	Enum: 'selectDCVMethod', 'canChangeDCVMethod', 'enterDCVCode'
domains	<i>Array of object (domainDescription)</i>	List of domains	For single domain certificate the array contains one element only.
MD5	<i>string</i>	md5 CSR's hash	
SHA256	<i>string</i>	sha256 CSR's hash	

uniqueValue	<i>string</i>	unique Request Tokens	
--------------------	---------------	-----------------------	--

Object (agreementStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'.
description	<i>string</i>	Text of description	
availableActions	<i>string</i>	Available actions on the SASP side for current status	Enum: 'canUpdateEmailAddress', 'resendAgreementEmail', 'needUpdateEmailAddress'
clickThroughStatus	<i>string</i>	Click-through status	Enum: 'notExecuted', 'Executed',
clickThroughEmailAddress	<i>string</i>	EV Subscriber Agreement & Certificate Request Form Email Address	
lastEmailSent	<i>date</i>		Format: YYYY-MM-DD
lastOpenDate	<i>date</i>		Format: YYYY-MM-DD

Object (callbackStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'
description	<i>string</i>	Text of description	
phoneNumber	<i>string</i>	Callback phone number	
availableActions	<i>Array of string</i>	Available actions on the SASP side for current status	Enum: 'callMeNow', 'callMeLater', 'provideCorrectPhoneNumber', 'enterCallbackCode', 'requestManualCallback'

Object (organizationStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'
description	<i>string</i>	Text of description	
availableActions	<i>Array of string</i>	Available actions on the SASP side for current status	Enum: 'uploadDocument'
organizationName	<i>Object (requirementStatus)</i>	Status of organization name check	For OV and EV only
legalExistence	<i>Object (requirementStatus)</i>	Status of legal existence check	For EV only
assumedName	<i>Object (requirementStatus)</i>	Status of assumed name check	For EV only
physicalExistence	<i>Object (requirementStatus)</i>	Status of physical existence check	For EV only
operationalExistence	<i>Object (requirementStatus)</i>	Status of operational existence check	For EV only

trademarkCheck	<i>Object (requirementStatus)</i>	Status of brand validation check	For OV and EV only
-----------------------	-----------------------------------	----------------------------------	--------------------

Object (secondApproval)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'
description	<i>string</i>	Text of description	

Object (fsupStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	General status	Enum: 'NotApplicable', 'ActionRequired', 'Completed', 'InProgress', 'Unknown'
description	<i>string</i>	Text of description	

Object (requirementStatus)

title	<i>string</i>	Section title	
status	<i>string</i>	Validation status	Enum: 'NotApplicable', 'InProgress', 'Completed'
description	<i>string</i>	Text of description	

Object (domainDescription)

domainName	<i>string</i>	Section title	
dcvStatus	<i>string</i>	Validation status	Enum: 'NotApplicable', 'InProgress', 'Completed'
selectedMethod	<i>String</i>	Selected DCV method	
methods	<i>Array of object (dcvMethod)</i>	Available DCV methods for domain	

Object (dcvMethod)

type	<i>string</i>	Type of method	Enum: 'whoisEmail', 'email', 'csrHash', 'manual'
title	<i>string</i>	Name of method	
level	<i>string</i>	Level of email list	Applicable for type = 'email'
selectedMethod	<i>string</i>	Selected DCV method	
options	<i>Array of string</i>	If type = 'whoisEmail'	List of email available address for DCV method based on Whois
		If type = 'csrHash'	List of available DCV methods based on CSR HASH: 'CNAME CSR Hash', 'HTTP CSR Hash', 'HTTPS CSR Hash'
		If type = 'email'	List of email available address for DCV method based on email
		If type = 'manual'	List contains only one element: 'Manual'

2.1. List of errors

errorCode	errorMessage
-1	'Request was not made over https!'
-2	'<errorItem>* is an unrecognized argument!'
-3	'The <errorItem>* argument is missing!'
-4	'The value of the <errorItem>* argument is invalid!'
-14	'An unknown error occurred!'
-16	'Incorrect login details, account is locked, password has expired or your source IP is blocked.'
-17	'Request used GET rather than POST!'
-42	'Call limit reached! Please try again later'
*<errorItem> - API parameter name	

Example of response:

Failed response:

```
{
  "errorCode": "-4",
  "errorMessage": "The value of the 'orderNumber' argument is invalid!",
  "data": {}
}
```

Successful response:

```
{
  "errorCode": 0,
  "errorMessage": "",
  "data": {
    "orderNumber": 654824,
    "productType": 410,
    "productName": "COMODO EV Multi-Domain SSL Certificate",
    "certificateDuration": 365,
    "productTerm": 365,
    "uploadDocumentUrl": "https://avspartners.sectigolabs.com/avs/ws/api/v002/order",
    "uploadDocumentToken":
      "eyJhbGciOiJIUzUxMi9.eyJzdWIiOiIvbn3JkZlVnU0ODI0Iiwic2NvcGVzIjpbIlBPU1RfUEFZUUV0Vf9QQUdFbX1VTRVlXSwiaXNzIjoiaHR0cHM6Ly9hdnMuc2VjdGlnby5jb20iLCJpYXQiOiJlE1NzUyOTQxMjEsImV4cCI6MTUzNTMzNzMyMX0.UteXiv1rmu-75kgAvVjuVHW0TPTHRgu_krv4BvfrggFOGbe8dGcmt-UfgOp0QKcF4eTOKXR3gQVsy-56eeAQ",
    "orderStatus": "Pending",
    "domainName": "Multiple domains",
    "primaryDomainName": "mn.gov",
    "maxMDCDomains": 3,
    "maxMDCWildcardDomains": 0,
    "csrStatus": {
      "title": "Submit your CSR",
      "status": "Completed",
      "description": "You have already submitted your CSR.",
      "availableActions": [
        "canReplaceCSR"
      ]
    },
    "dcvStatus": {
      "title": "Domain Control Validation",
      "status": "ActionRequired",
      "MD5": "55F0233969C9C4325B1A19D4B357ED20",
      "SHA256": "549B164F1F97EC4095C3FEF212F6B86016FF66F7F9115FFEBB2AB796FF226A8",
      "uniqueValue": "mTpFcllg",
      "description": "In order to verify your ownership of the domain in the application, it is mandatory to complete domain control validation.",
      "domains": [
        {

```

```
"domainName": "mn.gov",
"dcvStatus": "notCompleted",
"selectedMethod": "Not yet set",
"methods": [
  {
    "type": "email",
    "title": "Alternative Email Addresses",
    "level": 2,
    "options": [
      "admin@mn.gov",
      "administrator@mn.gov",
      "hostmaster@mn.gov",
      "postmaster@mn.gov",
      "webmaster@mn.gov"
    ]
  },
  {
    "type": "csrHash",
    "title": "Alternative Methods",
    "options": [
      "CNAME CSR Hash",
      "HTTP CSR Hash",
      "HTTPS CSR Hash"
    ]
  },
  {
    "type": "manual",
    "title": "None of the above",
    "options": [
      "Manual"
    ]
  }
],
{
  "domainName": "mnsure.org",
  "dcvStatus": "notCompleted",
  "selectedMethod": "Not yet set",
  "methods": [
    {
      "type": "email",
      "title": "Alternative Email Addresses",
      "level": 2,
      "options": [
        "admin@mnsure.org",
        "administrator@mnsure.org",
        "hostmaster@mnsure.org",
        "postmaster@mnsure.org",
        "webmaster@mnsure.org"
      ]
    },
    {
      "type": "csrHash",
      "title": "Alternative Methods",
      "options": [
        "CNAME CSR Hash",
        "HTTP CSR Hash",
        "HTTPS CSR Hash"
      ]
    },
    {
      "type": "manual",
      "title": "None of the above",
      "options": [
        "Manual"
      ]
    }
  ]
},
{
  "domainName": "state.mn.us",
  "dcvStatus": "notCompleted",
  "selectedMethod": "Not yet set",
  "methods": [
```

```

{
  "type": "email",
  "title": "Alternative Email Addresses",
  "level": 3,
  "options": [
    "admin@state.mn.us",
    "administrator@state.mn.us",
    "hostmaster@state.mn.us",
    "postmaster@state.mn.us",
    "webmaster@state.mn.us"
  ]
},
{
  "type": "csrHash",
  "title": "Alternative Methods",
  "options": [
    "CNAME CSR Hash",
    "HTTP CSR Hash",
    "HTTPS CSR Hash"
  ]
},
{
  "type": "manual",
  "title": "None of the above",
  "options": [
    "Manual"
  ]
}
]
},
"availableActions": [
  "selectDCVMethod"
]
},
"agreementStatus": {
  "title": "Agreement Execution",
  "status": "Completed",
  "clickThroughStatus": "Executed",
  "clickThroughEmailAddress": "andrey.tomikhin@sectigo.com",
  "lastEmailSent": "2019-11-20",
  "lastOpenDate": "2019-11-20"
},
"callbackStatus": {
  "title": "Signer Verification",
  "status": "InProgress",
  "description": "Our validation staff is currently verifying the phone number used in your application.\nOnce this process is complete, you will receive an email containing further instructions."
},
"organizationStatus": {
  "title": "Organization Validation",
  "status": "InProgress",
  "description": "Our validation staff will now attempt to validate your organization by consulting various federal and business databases.\nIn most cases this is a routine process that should be completed within a couple of working days.\nIf they cannot locate certain pieces of information about your organization, they may contact you and request documentation.",
  "organizationName": {
    "title": "Organization name",
    "status": "Unknown"
  },
  "legalExistence": {
    "title": "Legal Existence",
    "status": "Unknown"
  },
  "assumedName": {
    "title": "Assumed Name",
    "status": "Unknown"
  },
  "physicalExistence": {
    "title": "Physical Existence",
    "status": "Unknown"
  },
  "operationalExistence": {
    "title": "Operational Existence",
    "status": "Unknown"
  }
}

```

```

    },
    "trademarkCheck": {
      "title": "Trademark check",
      "status": "NotApplicable"
    }
  },
  "secondApprovalStatus": {
    "title": "Second Approval",
    "status": "Unknown"
  },
  "fsupStatus": {
    "title": "Fast Service Certificate",
    "status": "NotApplicable"
  }
}
}
}

```

3. Upload document

SASP allows a customer to download electronic copies of documents of title to speed up the process of company validation.

For these purposes, a separate API is created. JSON response contains URL of this API in the attribute '**uploadDocumentUrl**'. There is also the attribute '**uploadDocumentToken**' that contains a JWT token that is used for authorization on the server side when using an upload document API.

Example of the HTML form for uploading documents:

```

<form action="<uploadDocumentUrl>" method="POST" enctype="multipart/form-data">
  <input type="hidden" name="token" value="<uploadDocumentToken>" />
  <input type="hidden" name="successURL" value="<go to URL if successful>" />
  <input type="hidden" name="errorURL" value="<go to URL if failed>" />
  <br>Document: <input type="file" name="file" multiple/>
  <br>
  <br>Comment: <input type="text" name="comment" size=64/>
  <br>
  <br>
  <input type="submit" value="Upload"/>
</form>

```