

JSON API voor PartnerHub

Auteur: PETER BOGMAN | ESET NEDERLAND

Datum: 12-2-2021

Inhoudsopgave

Beveiliging.....	3
Functies	4
LicenseLetter	4
Verzoek.....	4
Antwoord.....	4
Renewals.....	4
Verzoek.....	4
Antwoord.....	4
LeakedLicenses	5
Verzoek.....	5
Antwoord.....	5
License	5
Verzoek.....	5
Antwoord.....	5
Price.....	7
Verzoek.....	7
Antwoord.....	8
Order	8
Verzoek.....	8
Antwoord.....	9
GetLicenseStatus	11
Verzoek.....	11
Antwoord.....	11
OrderDetails	11
Verzoek.....	11
Antwoord.....	12
OrderHistory.....	13
Verzoek.....	13
Antwoord.....	13
Lijsten.....	14
Kortingscodes	14
Producten	14
Producten upgraden.....	16

Voorbeeldcode	17
C#	17
Voorbeeld aanroep	17
Web aanroep helper	18
Serialisatie en deserialisatie classes	21
PHP	24
Licentie gegevens ophalen	24
Bestelling plaatsen	25

Aanroepen van de functies

De functies van de API kunnen aangeroepen worden doormiddel van het aanroepen van een URL over HTTPS. In de POST data van de aanroep komt JSON informatie met het verzoek. Bij het verzoek moeten ook twee headers toegevoegd worden. De headers zijn:

X-ESET-Guid

X-ESET-Signature

In deze headers komt een GUID die dient om de partner te identificeren, en een handtekening om bevestigen dat het bericht van die partner komt en ongewijzigd is. De handtekening wordt verder beschreven in het volgende hoofdstuk. De GUID is te vinden op het partner portaal (<https://partnerhub.eset.nl/api>).

De url die aangeroepen moet worden is <https://partnerhub.eset.nl/api>. Achter deze url komt de functienaam zoals in het hoofdstuk "Functies" beschreven staat. Bijvoorbeeld: <https://partnerhub.eset.nl/api/order>

Het antwoord van de aanroep bevat ook JSON data. Deze heeft standaard altijd twee variabelen:

- Result – Een boolean die aangeeft of het verzoek succesvol was of niet
- Messages – Een array/lijst aan strings met berichten over het verzoek. Als het verzoek niet succesvol is staan hierin de foutmeldingen.

Voorbeeld:

```
{
  "Result":false,
  "Messages":
  [
    "License not found",
    "Another message"
  ]
}
```

Beveiliging

Zoals in het vorige hoofdstuk beschreven is gaat de communicatie over HTTPS. Dit encrypteert de communicatie en voorkomt afluisteren. Daarnaast wordt een digitale handtekening toegevoegd om te bepalen dat het bericht ook van de afzender komt en dat het bericht ongewijzigd is. Deze handtekening is een SHA256 hash van de data (de JSON tekst) de GUID en een geheime sleutel.

Deze sleutel is te genereren via het partnerportaal. Bij vermoeden van misbruik van een sleutel kan met een druk op de knop een nieuwe sleutel genereerd worden.

Voorbeeldcode C#:

```
public static string Sha256Hash(string phrase)
{
    UTF8Encoding encoder = new UTF8Encoding();
    SHA256CryptoServiceProvider hasher = new SHA256CryptoServiceProvider();
    byte[] hashedDataBytes = null;
    hashedDataBytes = hasher.ComputeHash(encoder.GetBytes(phrase));
    return ByteArrayToString(ref hashedDataBytes);
}

public static string ByteArrayToString(ref byte[] inputArray)
{
    StringBuilder output = new StringBuilder(inputArray.Length);
    for (int i = 0; i <= inputArray.Length - 1; i++)
    {
        output.Append(inputArray[i].ToString("x2"));
    }
    return output.ToString();
}

var signature = Sha256Hash(data + guid + secretkey);
```

Voorbeeldcode PHP:

```
$signature = hash('sha256', $data . $guid . $secretkey);
```

Functies

LicenseLetter

Deze functie stuurt de licentiebrief van de meegegeven licentie op naar het e-mail adres waarop de licentie staat.

Url: <https://partnerhub.eset.nl/api/licenseletter>

Verzoek

```
{
  "LicenseId": "33C-P2S-8DN",
  "LicenseKey": "BE9W-X82D-HH7P-6HV5-DPVH"
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
LicenseId	string	nee*	Dit is de publieke code van de licentie (de EAV-code of public license id)
LicenseKey	string	nee*	Dit is de privé code van de licentie (password of licensekey)

*Één van de twee moet meegeleverd worden

Antwoord

```
{
  "Result":true,
  "Messages":
  []
}
```

Waarden:

Naast de standaard waarden heeft dit verzoek geen andere waarden.

Renewals

De functie haalt een lijst op van alle aflopende licenties die gekoppeld zijn aan de partner.

Url: <https://partnerhub.eset.nl/api/renewals>

Verzoek

Dit verzoek bevat geen JSON

Antwoord

```
{
  "Result":true,
  "Messages":[]
}
```

Waarden:

Naast de standaard waarden heeft dit verzoek geen ander waarden.

LeakedLicenses

Deze functie haalt een lijst op van alle gelekte licenties die gekoppeld zijn aan de partner.

Url: <https://partnerhub.eset.nl/api/leakedlicenses>

Verzoek

Dit verzoek bevat geen JSON

Antwoord

```
{
  "Result": true,
  "Messages": []
}
```

Waarden:

Naast de standaard waarden heeft dit verzoek geen andere waarden.

License

Deze functie haalt de details van een licentie op.

Url: <https://partnerhub.eset.nl/api/license>

Verzoek

```
{
  "LicenseId": "33C-P2S-8DN",
  "LicenseKey": "BE9W-X82D-HH7P-6HV5-DPVH"
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
LicenseId	string	ja	Dit is de publieke code van de licentie (de EAV-code of public license id)
LicenseKey	string	nee*	Dit is de prive code van de licentie (password of licensekey)

*De prive code van de licentie is niet verplicht mits de licentie aan uw debiteurennummer is gekoppeld.

Antwoord

```
{
  "License": {
    "LicenseKey": "BE9W-X82D-HH7P-6HV5-DPVH",
    "LicenseId": "33C-P2S-8DN",
    "Username": "EAV-0135537076",
    "Password": "s44x223b7m",
    "ProductCode": "321",
    "ProductName": "",
    "Quantity": 29,
    "PurchaseType": "Renew",
    "CreatedDate": "28-07-2015",
    "ExpirationDate": "27-07-2017",
    "CustomerName": "Peter",
    "CustomerCompany": "",
    "CustomerEmail": "peter@nod32.nl",
    "DiscountCode": 0,
  }
}
```

```

        "Status": "Active",
        "BundleProductCode": "2001",
        "BundleProductName": "ESET Endpoint Protection Advanced",
        "BundleQuantity": 29,
        "ELAPassword": "",
        "CountryId": 60,
        "RenewProductcode": "E54-020107AA"
    },
    "Result": true,
    "Messages": []
}

```

Waarden:

Naam	Datatype	Beschrijving
LicenseKey	string	De licentiesleutel. Als de licentie van een serienummer afkomstig is, dan is dit het serienummer.
Username	string	De licentie gebruikersnaam.
Password	string	Het licentie wachtwoord.
ProductCode	string	De ESET Productcode van dit product. Als een bundel is besteld is dit de code van het product IN de bundel. Zie hoofdstuk Lijsten, tabel Producten
ProductName	string	De naam van dit product. Als een bundel is besteld is dit de naam van het product IN de bundel.
Quantity	int	Het aantal van de licentie. Als een bundel is besteld is dit het aantal van het product IN de bundel.
PurchaseType	string	Het type aanschaf van deze licentie. Dit kan zijn: New, Renew, Upgrade, Enlarge, NFR, Trial, Downgrade.
CreatedDate	string	De besteldatum van deze licentie. Let op, in het geval van een verlenging is het de aanschafdatum van de verlenging en niet van de originele nieuwe licentie.
ExpirationDate	string	De afloopdatum van de licentie
CustomerName	string	De naam van de klant.
CustomerCompany	string	De bedrijfsnaam van de klant.
CustomerEmail	string	Het e-mail adres van de klant.
DiscountCode	int	De kortingscode. Zie hoofdstuk Lijsten, tabel Kortingscodes.
Status	string	Active, Canceled, Inactive of Suspended.
BundleProductCode	string	De ESET productcode van de bundel die besteld is. Deze is leeg als er geen bundel besteld is.
BundleProductName	string	De naam van de bundel die besteld is. Deze is leeg als er geen bundel besteld is.
BundleQuantity	int	Het aantal van de bundel die besteld is. Deze is 0 (nul) als er geen bundel besteld is.
ELAPassword	string	Het ESET License Administrator wachtwoord.
CountryId	int	Het land nummer van waar de licentie is aangeschaft. Als dit 60 is, is het een Nederlandse licentie.
RenewProductcode	string	De productcode voor een standaard verlenging van de licentie voor 1 jaar

Price

Met deze functie kan de prijs van een product berekend worden. Dit geldt zowel voor nieuwe verkoop als verlengingen, upgrades, enlarges etc.

Url: <https://partnerhub.eset.nl/api/price>

Verzoek

```
{
  "LicenseKey": "BE9W-X82D-HH7P-6HV5-DPVH",
  "LicenseId": "33C-P2S-8DN",
  "NewProductCode": "2002",
  "NewQuantity": 30,
  "Period": 1,
  "TaxRate": 1.21,
  "DiscountCode": 4
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
LicenseId	string	ja*	Dit is de publieke code van de licentie (de EAV-code of public license id)
LicenseKey	string	nee	Dit is de privé code van de licentie (password of licensekey/serienummer)
NewProductCode	string	ja	De ESET productcode van het nieuwe product. Als deze hetzelfde is als het huidige product, wordt het een verlenging of enlarge. Zie hoofdstuk Lijsten, tabel Producten
NewQuantity	int	ja	Het nieuwe aantal van de licentie. Als deze hetzelfde is als de huidige licentie, wordt het een verlenging. Bij een bundel product wordt dit het bundel aantal.
Period	int	ja	De verleng periode. 1 = 1 jaar 2 = 2 jaar 5 = 3 jaar 6 = zonder verlenging
TaxRate	decimal	nee	Indien gewenst kan hier een belastinghoeveelheid worden meegegeven. Voor standaard BTW wordt dit 1.21. De prijs wordt vermenigvuldigd met dit aantal.
Discountcode	Int	nee	Het type korting dat toegepast moet worden op de prijs. Bijvoorbeeld non-profit (4). Zie hoofdstuk Lijsten tabel Kortingscodes voor een lijst aan opties.

*Deze code is niet verplicht op het moment dat u een prijs opvraagt voor nieuwe verkoop.

Antwoord

```
{
  "Price": 950.2493,
  "DiscountPrice": 950.2493,
  "ProductCode": "E54-020104AA",
  "Result": true,
  "Messages": []
}
```

Waarden:

Naam	Datatype	Beschrijving
Price	decimal	De prijs zonder korting. Als deze 0 is, is deze set aan waarden niet mogelijk om te bestellen.
DiscountPrice	decimal	De prijs met korting. Als deze 0 is dan is er een actie/korting waardoor we het product gratis weggeven.
ProductCode	string	De productcode die gebruikt kan worden om de bestelling te plaatsen. Deze code heeft geen staffeltoevoeging is gebaseerd op de opties die met dit verzoek zijn meegestuurd.

Order

Met deze functie kan een of meer licenties besteld worden.

Url: <https://partnerhub.eset.nl/api/order>

Verzoek

```
{
  "TestOrder": false,
  "CustomerName": "Peter",
  "CustomerEmail": "peter@eset.nl",
  "CustomerCompany": "ESET",
  "LicenseLetterCC": "peter@nod32.nl",
  "OrderReference": "Api-01235",
  "OrderItems":
  [
    {
      "ProductCode": E01-010107AA,
      "Quantity": 1,
      "LicenseKey": "",
      "LicenceId": ""
    }
  ]
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
TestOrder	boolean	nee	Als TestOrder true is, wordt de licentie direct na aanmaken geannuleerd en wordt de bestelling niet gefactureerd.
CustomerName	string	ja	De naam van de klant.
CustomerEmail	string	Ja	Het e-mailadres van de klant.
CustomerCompany	string	nee	De bedrijfsnaam van de klant.
LicenseLetterCC	string	nee	Het e-mailadres waar een kopie van de licentiebrief naar toe moet.

OrderReference	string	nee	Een vrij in te vullen referentie voor deze order. Bijvoorbeeld een ordernummer uit uw systeem.
OrderItems	lijst	ja	Een lijst/array aan producten die besteld moeten worden.
ProductCode	string	ja	De productcode uit onze prijslijst minus het laatste karakter (dit is het karakter dat de staffel aangeeft maar omdat we een aantal meegeven is die niet nodig).
Quantity	int	ja	Het aantal apparaten/gebruikers waarvoor de licentie is.
LicenseKey	string	nee*/**	Dit is de privé code van de licentie (password of licensekey/serienummer).
Licenceld	string	ja**	Dit is de publieke code van de licentie (de EAV-code of public license id).

*De privé code van de licentie is niet verplicht mits de licentie aan uw debiteurennummer is gekoppeld.

** Alleen invullen bij een verlenging, upgrade of enlarge etc.

Antwoord

```
{
  "Ordernumber": 62002680,
  "Licenses":
  [
    {
      "LicenseKey": "AGT9-XB64-W7UP-TC2N-AVB5",
      "LicenseId": "333-3VD-3F4",
      "Username": "DEV-0000341388",
      "Password": "brfdbfrhej",
      "ProductCode": "106",
      "ProductName": "ESET NOD32 Antivirus",
      "Quantity": 1,
      "PurchaseType": "New",
      "CreatedDate": "06-10-2015",
      "ExpirationDate": "06-10-2016",
      "CustomerName": "Peter",
      "CustomerCompany": "ESET",
      "CustomerEmail": "peter@nod32.nl",
      "DiscountCode": 7,
      "Status": "Active",
      "BundleProductCode": "",
      "BundleProductName": "",
      "BundleQuantity": 0,
      "ELAPassword": "",
      "CountryId": 60
    }
  ],
  "Result": true,
  "Messages": []
}
```

Waarden:

Naam	Datatype	Beschrijving
Ordernumber	int	Het ordernummer uit ons systeem. Het is voor de communicatie gemakkelijk om deze op te slaan.
Licenses	lijst	Een lijst/array aan licenties die besteld zijn.
LicenseKey	string	De licentiesleutel. Als de licentie van een serienummer afkomstig is, dan is dit het serienummer.
Username	string	De licentie gebruikersnaam.
Password	string	Het licentie wachtwoord.
ProductCode	string	De ESET Productcode van dit product. Als een bundel is besteld is dit de code van het product IN de bundel. Zie hoofdstuk Lijsten, tabel Producten
ProductName	string	De naam van dit product. Als een bundel is besteld is dit de naam van het product IN de bundel.
Quantity	int	Het aantal van de licentie. Als een bundel is besteld is dit het aantal van het product IN de bundel.
PurchaseType	string	Het type aanschaf van deze licentie. Dit kan zijn: New, Renew, Upgrade, Enlarge, NFR, Trial, Downgrade.
CreatedDate	string	De besteldatum van deze licentie. Let op, in het geval van een verlenging is het de aanschafdatum van de verlenging en niet van de originele nieuwe licentie.
ExpirationDate	string	De aflooptdatum van de licentie
CustomerName	string	De naam van de klant.
CustomerCompany	string	De bedrijfsnaam van de klant.
CustomerEmail	string	Het e-mail adres van de klant.
DiscountCode	int	De kortingscode. Zie hoofdstuk Lijsten, tabel Kortingscodes.
Status	string	Active, Canceled, Inactive of Suspended.
BundleProductCode	string	De ESET productcode van de bundel die besteld is. Deze is leeg als er geen bundel besteld is.
BundleProductName	string	De naam van de bundel die besteld is. Deze is leeg als er geen bundel besteld is.
BundleQuantity	int	Het aantal van de bundel die besteld is. Deze is 0 (nul) als er geen bundel besteld is.
ELAPassword	string	Het ESET License Administrator wachtwoord.
CountryId	int	Het land nummer van waar de licentie is aangeschaft. Als dit 60 is, is het een Nederlandse licentie.

GetLicenseStatus

Deze functie haalt de status van een licentie op (status, besteldatum en aflooptdatum).

Url: <https://partnerhub.eset.nl/api/getlicensestatus>

Verzoek

```
{
  "LicenseId": "",
  "LicenseKey": "NLA2-2B4U-22F8-7E2V-AHAW"
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
LicenseId	string	nee*	Dit is de publieke code van de licentie (de EAV-code of public license id)
LicenseKey	string	nee*	Dit is de privé code van de licentie (password of licensekey/serial)

*Één van de twee moet meegeleverd worden

Antwoord

```
{
  "Status": "Canceled",
  "OrderDate": "/Date(1439986169403)/",
  "ExpirationDate": "/Date(1442527200000)/",
  "Result": true,
  "Messages": []
}
```

Waarden:

Naam	Datatype	Beschrijving
Status	string	Active, Inactive, Canceled of Suspended
OrderDate	datetime	De datum dat de laatste bestelling was voor deze licentie. Als de licentie bijvoorbeeld gisteren verlengd is, is dit de datum van gisteren.
ExpirationDate	datetime	De huidige aflooptdatum van de licentie.

OrderDetails

Deze functie haalt de details van een bestelling op.

Url: <https://partnerhub.eset.nl/api/orderdetails>

Verzoek

```
{
  "OrderNr": 62017592,
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
OrderNr	int	ja	Het ordernummer van de bestelling

Antwoord

```

{
  "Order": {
    "OrderID": 62017592,
    "OrderSubtotalInclTax": 100.01,
    "OrderSubtotalExclTax": 82.65,
    "OrderSubTotalDiscountInclTax": 0.0,
    "OrderSubTotalDiscountExclTax": 0.0,
    "OrderTax": 17.36,
    "OrderTotal": 100.01,
    "OrderDiscount": 0.0,
    "PurchaseOrderNumber": "",
    "BillingFirstName": "Peter",
    "BillingLastName": "Bogman",
    "BillingPhoneNumber": "061234567",
    "BillingEmail": "peter@eset.nl",
    "BillingCompany": "ESET Nederland",
    "BillingAddress1": "Trapezium 304",
    "BillingAddress2": "",
    "BillingCity": "Sliedrecht",
    "BillingZipPostalCode": "3364DL",
    "CreatedOn": "/Date(1479821174627)/",
    "OrderItems": [
      {
        "Name": "ESET Smart Security Premium - 1 Jaar",
        "UnitPriceInclTax": 20.0013,
        "UnitPriceExclTax": 16.53,
        "PriceInclTax": 100.0065,
        "PriceExclTax": 82.65,
        "License": "3AA-XXX-XXX",
        "Quantity": 5,
        "DiscountAmountInclTax": 0.0,
        "DiscountAmountExclTax": 0.0,
        "SKU": "E54-010104AA",
        "AttributeDescription": "Product: 141 - Licentie: 3AA- XXX-XXX\\u003cbr
        /\u003eKlantnaam: Peter\u003cbr /\u003eBedrijfsnaam: ESET Nederland\u003cbr
        /\u003eLicentiebrief: peter@eset.nl\u003cbr /\u003eLicentiebrief CC: info@eset.nl"
      }
    ]
  },
  "Result": true,
  "Messages": []
}

```

OrderHistory

Deze functie haalt de details van een bestelling op.

Url: <https://partnerhub.eset.nl/api/orderhistory>

Verzoek

```
{
  "FromDate": "2021-01-01",
  "ToDate": "2021-01-31"
}
```

Waarden:

Naam	Datatype	Verplicht	Beschrijving
FromDate	string	ja	De vanaf datum van wanneer de orders geplaatst zijn
ToDate	string	ja	De tot datum tot wanneer de orders geplaatst zijn. Let op de orders die op die datum geplaatst zijn worden niet meegenomen.

Antwoord

Het antwoord is vergelijkbaar als die van OrderDetails. Alleen bevat dit verzoek een lijst aan orders onder de naam "Orders":

```
{
  "Orders": [
    {
      "OrderID": 62017592,
      etc
    },
    {
      "OrderID": 62017593,
      etc
    }
  ],
  "Result": true,
  "Messages": []
}
```

Lijsten

Kortingscodes

Discount type	Code
Government	1
Education	2
Student	3
Non-profit / NGO	4
No discount	7

Producten

Code	Naam
106	ESET NOD32 Antivirus,
107	ESET Endpoint Antivirus for Windows,
206	ESET Mail Security for Microsoft Exchange Server,
207	ESET Mail Security for IBM Lotus Domino,
402	ESET NOD32 Antivirus for Kerio Control,
208	ESET NOD32 Antivirus for Kerio Connect,
308	ESET Mail Security for Linux/BSD/Solaris,
309	ESET File Security for Linux/BSD/Solaris,
110	ESET Smart Security,
111	ESET Endpoint Security for Windows,
307	ESET Gateway Security for Linux/BSD/Solaris,
211	ESET File Security for Microsoft Windows Server,
403	ESET NOD32 for Safe-Box,
602	ESET NOD32 Antivirus for Dell PowerVault Storage,
108	ESET Cyber Security,
109	ESET NOD32 Antivirus for Linux Desktop,
709	ESET Mobile Security,
710	ESET Endpoint Security for Android,
112	ESET Endpoint Antivirus for Mac OS X,
113	ESET Endpoint Antivirus for Linux,
114	ESET NOD32 Antivirus Business Edition for Mac OS X (Harvard),
713	M+ Guardian,
212	ESET Gateway Security for Microsoft Forefront Threat Management Gateway,
715	ESET Mobile Security for Android,
116	ESET Cyber Security Pro,
716	ESET Mobile Security for Nokia Store,
2000	ESET Endpoint Protection Standard,
310	ESET File Security,
311	ESET Mail Security,
312	ESET Gateway Security,
2001	ESET Endpoint Protection Advanced,

2002	ESET Secure Business,
2003	ESET Secure Enterprise,
1500	ESET Home Office Security Pack,
1501	ESET Small Business Security Pack,
250	ESET Secure Authentication,
718	ESET Mobile Security
313	ESET Security for Microsoft SharePoint Server (Per User),
600	SMS for ESET Secure Authentication,
719	ESET Mobile Security For Free,
720	ESET Parental Control For Google Play,
721	ESET Parental Control for Android,
722	ESET Parental Control For Free,
315	ESET Security for Kerio,
251	ESET Endpoint Encryption Essential Edition,
252	ESET Endpoint Encryption Standard Edition,
253	ESET Endpoint Encryption Pro,
254	ESET Endpoint Encryption Server,
316	ESET Security for Microsoft SharePoint Server (Per Server),
1600	(N) ESET Multi-Device Security Pack,
124	ESET Multi-Device Security Pack,
117	ESET Endpoint Security for Mac OS X,
255	ESET Endpoint Encryption Mobile,
256	DESlock+ Essential Edition / Promo,
320	ESET Endpoint Antivirus + File Security,
321	ESET Endpoint Security + File Security,
2011	ESET Endpoint Antivirus Suite,
132	ESET Virtualization Security (per VM),
133	ESET Virtualization Security (per Host),
134	ESET Virtualization Security (per Processor),
249	ESET Secure Authentication for ERA,
1605	(N) ESET Multi-Device Security Pack Youfone,
128	ESET Multi-Device Security Youfone,
220	ESET File Security for Terminal Server,
141	ESET Smart Security Premium,
140	ESET Internet Security,
150	ESET Threat Intelligence Automated Sample Analysis,
151	ESET Threat Intelligence Early Warning,
152	ESET Threat Intelligence Botnet Feed,
2150	ESET Threat Intelligence (ETI),
322	ESET Enterprise Inspector,
2020	ESET Enterprise Inspector and Endpoint Security,
2013	ESET Dynamic Threat Defense,
2014	ESET Endpoint Protection Standard Cloud,
2015	ESET PROTECT Entry (ESET Endpoint Protection Advanced Cloud),

2016	ESET Secure Business Cloud,
154	ESET Cloud Office Security,
756	ESET Full Disk Encryption,
757	ESET Full Disk Encryption for ECA (Discontinued),
999	ESET Security Management Center Cloud,
330	ESET Dynamic Endpoint Protection,
332	ESET Targeted Attack Protection,
2017	ESET PROTECT Advanced (ESET Remote Workforce Offer),
333	ESET Dynamic Endpoint Protection – Antivirus Level,
2018	ESET PROTECT Complete,
2019	ESET PROTECT Enterprise,
2021	ESET PROTECT Mail Plus

[Producten upgraden](#)

Om te zien welke upgrades er mogelijk zijn kunt u in PartnerHub bij het product kijken naar de producten die daarbij staan.

Voorbeeldcode

C#

Voorbeeld aanroep

```
//This example code uses Newtonsoft.Json
string url = "https://partnerhub.eset.nl/api/";
string key = "";
string guid = "";

public APIResult Order()
{
    var request = new APIOrderRequest()
    {
        CustomerCompany = "ESET",
        CustomerEmail = "peter@eset.nl",
        CustomerName = "Peter",
        OrderReference = "Api-01235",
        LicenseLetterCC = "peter@nod32.nl",
        TestOrder = false
    };
    request.OrderItems.Add(new APIOrderItem()
    {
        Period = 1,
        ProductCode = 106,
        Quantity = 1
    });
    var json = JsonConvert.SerializeObject(request);
    var response = DoRequest("Order", json);
    var orderresult = JsonConvert.DeserializeObject<APIOrderResult>(response);
}

private string DoRequest(string resource, string data, string method = "POST")
{
    var signature = Sha256Hash(data + guid + key);

    WebRequestHelper helper = new WebRequestHelper();
    helper.AddHeader("X-ESET-Guid", guid);
    helper.AddHeader("X-ESET-Signature", signature);
    var response = helper.DoRequest(url, resource, data, "application/json", method);

    helper.Clear();
    return response;
}

//Hash method
public static string Sha256Hash(string phrase)
{
    UTF8Encoding encoder = new UTF8Encoding();
    SHA256CryptoServiceProvider hasher = new SHA256CryptoServiceProvider();
    byte[] hashedDataBytes = null;
    hashedDataBytes = hasher.ComputeHash(encoder.GetBytes(phrase));
    return ByteArrayToString(ref hashedDataBytes);
}
```

```
public static string ByteArrayToString(ref byte[] inputArray)
{
    StringBuilder output = new StringBuilder(inputArray.Length);
    for (int i = 0; i <= inputArray.Length - 1; i++)
    {
        output.Append(inputArray[i].ToString("x2"));
    }
    return output.ToString();
}
```

Web aanroep helper

```
//This class helps making a request to an url for example for making calls to
webservices
public class WebRequestHelper : IDisposable
{
    private List<KeyValuePair<string, string>> parameters;
    private List<KeyValuePair<string, string>> headers;

    public WebRequestHelper()
    {
        parameters = new List<KeyValuePair<string, string>>();
        headers = new List<KeyValuePair<string, string>>();
    }

    public void AddParameter(string key, object value)
    {
        if (!string.IsNullOrEmpty(key) && value != null)
            parameters.Add(new KeyValuePair<string, string>(key, string.Format("{0}",
                value)));
    }

    public void AddHeader(string key, string value)
    {
        if (!string.IsNullOrEmpty(key) && value != null)
            headers.Add(new KeyValuePair<string, string>(key, value));
    }
}
```

```

public string DoRequest(string url, string resource, string data = "", string
    contentType = "application/x-www-form-urlencoded", string method =
    "POST", int timeout = 300000, Encoding encoding = null)
{
    string result = "";
    Encoding enc = encoding == null ? Encoding.UTF8 : encoding;
    string requestUrl = url.EndsWith("/") ? url : url + "/";
    requestUrl += resource;
    HttpRequest request = (HttpRequest)WebRequest.Create(requestUrl);
    request.ReadWriteTimeout = timeout;
    request.Method = method;
    request.ContentType = contentType;

    foreach (var pair in headers)
    {
        request.Headers.Add(pair.Key + ":" + pair.Value);
    }

    if (request.Method == "POST" || request.Method == "PUT")
    {
        byte[] bytes;
        if (string.IsNullOrEmpty(data))
        {
            StringBuilder builder = new StringBuilder();
            foreach (var pair in parameters)
            {
                builder.Append(pair.Key + "=" + pair.Value + "&");
            }
            bytes = enc.GetBytes(builder.ToString());
        }
        else
        {
            bytes = enc.GetBytes(data);
        }
        request.ContentLength = bytes.Length;
        Stream os = request.GetRequestStream();
        os.Write(bytes, 0, bytes.Length);
        os.Close();
    }
    HttpResponse response;
    try
    {
        response = (HttpResponse)request.GetResponse();
        StreamReader reader = new StreamReader(response.GetResponseStream());
        result = reader.ReadToEnd();
    }
    catch (WebException e)
    {
        if (((HttpResponse)e.Response).StatusCode == HttpStatusCode.BadRequest)
        {
            StreamReader reader = new
                StreamReader(e.Response.GetResponseStream());
            result = reader.ReadToEnd();
        }
        else
        {
            result = Enum.GetName(typeof(HttpStatusCode),
                ((HttpResponse)e.Response).StatusCode);
        }
    }
    return result;
}

```

```
    }  
  
    public List<KeyValuePair<string, string>> GetParameters()  
    {  
        return parameters;  
    }  
  
    public List<KeyValuePair<string, string>> GetHeaders()  
    {  
        return headers;  
    }  
  
    public void Dispose()  
    {  
        Clear();  
    }  
  
    public void Clear()  
    {  
        parameters.Clear();  
        headers.Clear();  
    }  
}
```

Serialisatie en deserialisatie classes

```
//classes used to serialize to json for orders.
public class APIOrderRequest
{
    public APIOrderRequest()
    {
        if (OrderItems == null) OrderItems = new List<APIOrderItem>();
    }
    public bool TestOrder { get; set; }
    public string CustomerName { get; set; }
    public string CustomerEmail { get; set; }
    public string CustomerCompany { get; set; }

    public string LicenseLetterCC { get; set; }
    public string OrderReference { get; set; }

    public List<APIOrderItem> OrderItems { get; set; }
}

public class APIOrderItem
{
    public string ProductCode { get; set; }
    public int Quantity { get; set; }
    public string LicenseKey { get; set; }
    public string LicenceID { get; set; }
}

//class used to serialise a price request
public class APIPriceRequest
{
    public string LicenseKey { get; set; }
    public string LicenseId { get; set; }
    public string NewProductCode { get; set; }
    public int NewQuantity { get; set; }
    public int Period { get; set; }
    public decimal TaxRate { get; set; }
}

//class used to serialize a license status.
public class APILicenseRequest
{
    public string LicenseKey { get; set; }
    public string LicenseId { get; set; }
}
```

```

//class to deserialize a basic API result
public class APIResult
{
    public APIResult()
    {
        if (Messages == null) Messages = new List<string>();
    }
    public bool Result { get; set; }
    public List<string> Messages { get; set; }
}

public class APILeakedLicenseResult : APIResult
{
    public APILeakedLicenseResult()
    {
        if (LeakedLicenses == null) LeakedLicenses = new List<APILeakedLicense>();
    }
    public List<APILeakedLicense> LeakedLicenses { get; set; }
}

public class APILeakedLicense
{
    public string Type { get; set; }
    public string EAV { get; set; }
    public string ProductNaam { get; set; }
    public int Aantal { get; set; }
    public string Klant { get; set; }
    public string Email { get; set; }
    public int IPs { get; set; }
    public int VSDB_Downloads { get; set; }
    public int Downloads { get; set; }
}

//class used to deserialize a renewal result
public class APIRenewalResult : APIResult
{
    public APIRenewalResult()
    {
        if (Renewals == null) Renewals = new List<APIRenewal>();
    }
    public List<APIRenewal> Renewals { get; set; }
}

public class APIRenewal
{
    public string Username { get; set;}
    public string LicenseKey { get; set; }
    public string Email { get; set; }
    public string Name { get; set; }
    public DateTime ExpirationDate { get; set; }
    public string Product { get; set; }
    public string ProductCode { get; set; }
    public int Quantity { get; set; }
    public string OrderID { get; set; }
    public string PurchaseType { get; set; }
}

```



```
//class used to deserialise a license status result
public class APILicenseStatusResult : APIResult
{
    public string Status { get; set; }
    public DateTime OrderDate { get; set; }
    public DateTime ExpirationDate { get; set; }
}

//classes used to deserialize the result of an order
public class APIOrderResult : APIResult
{
    public APIOrderResult()
    {
        if (Licenses == null) Licenses = new List<APILicense>();
    }
    public int Ordernumber { get; set; }
    public List<APILicense> Licenses { get; set; }
}

public class APILicense
{
    public string LicenseKey { get; set; }
    public string LicenseId { get; set; }
    public string Username { get; set; }
    public string Password { get; set; }
    public string ProductCode { get; set; }
    public string ProductName { get; set; }
    public int Quantity { get; set; }
    public string PurchaseType { get; set; }
    public string CreatedDate { get; set; }
    public string ExpirationDate { get; set; }
    public string CustomerName { get; set; }
    public string CustomerCompany { get; set; }
    public string CustomerEmail { get; set; }
    public int DiscountCode { get; set; }
    public string Status { get; set; }
    public string BundleProductCode { get; set; }
    public string BundleProductName { get; set; }
    public int BundleQuantity { get; set; }
    public string ELAPassword { get; set; }
}

//class used to deserialize a license result
public class APILicenseResult : APIResult
{
    public APILicense License { get; set; }
}
```

PHP

Licentie gegevens ophalen

```
<?php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$LicenseID = "3AH-4MR-7RX";
$LicenseKey = "9WC3-XNP8-JUC7-V5D2-VXXG";

$json = array(
    'LicenseId' => $LicenseID,
    'LicenseKey' => $LicenseKey,
);

$url = "https://partnerhub.eset.nl/api/license";
$secretkey = ""; //Get this from https://partnerhub.eset.nl/api
$guid = ""; //Get this from https://partnerhub.eset.nl/api

$jsonstring = json_encode($json);

$signature = hash( 'sha256', $jsonstring . $guid . $secretkey);

$headers = array(
    'X-ESET-Guid: ' . $guid,
    'X-ESET-Signature: ' . $signature,
    'Cache-Control: no-cache',
    'Content-Type: application/json'
);

$request = curl_init();
$options = array(CURLOPT_URL => $url,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_POST => 1,
    CURLOPT_POSTFIELDS => $jsonstring,
    CURLOPT_HTTPHEADER => $headers
);

curl_setopt_array($request, $options);
$result = curl_exec($request);

if($result === False)
{
    print(curl_error($request));
}
else
{
    print($result);
}
curl_close($request);
?>
```

Bestelling plaatsen

```

<?php

$baseUrl = "https://partnerhub.eset.nl/api/"; //Alter this url in case the
partnerportal changes url.

$guid = ""; //Your guid as found on https://partner.eset.nl/api
$secretkey = ""; //Your secretkey as found on
https://partner.eset.nl/api

$testOrder = true; //Set to false to place a real order
$customerName = ""; //The name of the customer
$customerEmail = ""; //The email address of the customer
$customerCompany = ""; //Optional name of the company of the customer
$licenseLetterCC = ""; //Optional email address of the partner
$orderReference = ""; //Optional reference for the order in your system, for
example your ordernumber
$productCode = "E01-010107AA"; //Productcode as per the pricelist minus the last
character
$quantity = 1; //Quantity of the product

//Place the order (only one product is supported in the example, alter the function to
order more products in a single order)
$response = orderEsetLicense($guid, $secretkey, $testOrder, $customerName,
$customerEmail, $customerCompany, $licenseLetterCC, $orderReference, $productCode,
$quantity);

//Output information of the response. In an actual implementation you would use these
values instead of displaying them like this.
if($response['Result'] === false) //Something went wrong, read/log the messages
{
    echo "<div>";
    print_r(($response['Messages']));
    echo "</div>";
}
else //Read from the response
{
    $license = $response['Licenses'];
    echo "<div>";
    echo "<br />Ordernummer= ". $response['Ordernumber'];
    echo "<br />LicenseKey= ". $license['LicenseKey'];
    echo "<br />LicenseId= ". $license['LicenseId'];
    echo "<br />Username= ". $license['Username'];
    echo "<br />Password= " . $license['Password'];
    echo "<br />ProductCode= " . $license['ProductCode'];
    echo "<br />ProductName= " . $license['ProductName'];
    echo "<br />Quantity= " . $license['Quantity'];
    echo "<br />PurchaseType= " . $license['PurchaseType'];
    echo "<br />ExpirationDate= " . $license['ExpirationDate'];
    echo "<br />CustomerName= " . $license['CustomerName'];
    echo "<br />CustomerCompany= " . $license['CustomerCompany'];
    echo "<br />CustomerEmail= " . $license['CustomerEmail'];
    echo "<br />Status= " . $license['Status'];
    echo "</div>";
}
}

```

```

//Function to order a single product using the ESET NL partner API
function orderEsetLicense($guid, $secretkey, $testorder, $orderReference,
$customerName, $customerEmail, $customerCompany, $licenseLetterCC, $productCode,
$quantity)
{
    $url = $baseUrl . "order";

    //Create order object
    $order = [
        'TestOrder'      => $testOrder,
        'CustomerName'   => $customerName,
        'CustomerEmail'  => $customerEmail,
        'CustomerCompany' => $customerCompany,
        'LicenseLetterCC' => $licenseLetterCC,
        'OrderReference' => $orderReference,
        'OrderItems'     => array()
    ];

    //Add product to order
    $OrderItems = array(
        'ProductCode' => $productCode,
        'Quantity'    => $quantity
    );
    array_push($order['OrderItems'], $OrderItems);

    //Convert array to a json string
    $jsonstring = json_encode($order);

    $response = apiRequest($url, $jsonstring, $guid, $secretkey, $testOrder);
    return $response;
}

function apiRequest($url, $jsonstring, $guid, $secretkey, $showDebugOutput)
{
    //Calculate digital signature
    $signature = hash('sha256', $jsonstring . $guid . $secretkey);

    //Create headers
    $headers = array(
        'X-ESET-Guid: ' . $guid,
        'X-ESET-Signature: ' . $signature,
        'Cache-Control: no-cache',
        'Content-Type: application/json'
    );

    //Create the request with curl
    $request = curl_init();
    $options = array(
        CURLOPT_URL          => $url,
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_POST         => 1,
        CURLOPT_POSTFIELDS   => $jsonstring,
        CURLOPT_HTTPHEADER   => $headers
    );
    curl_setopt_array($request, $options);
}

```

```
//Perform the request and store the result
$result = curl_exec($request);
curl_close($request);

//Some debugging output incase curl fails. Only when doing a testorder.
if($result === False && $showDebugOutput === true)
{
    print(curl_error($request));
}

//Convert json string to object and return the object
$response = json_decode($result, true);
return $response;
}
?>
```