Version

2.1

Payten

NestPay®

Hash Version 3

www.asseco-see.com.tr

Document Information

Project/Product Name	Hash Documentation		
Project Manager	Burak Kutlu		
Document Version No	2.1		
Document Code			
Prepared By		**Preparation Date**	
Reviewed By		**Review Date**	

Distribution List

From	**Date**	**Phone/Fax**

To	**Action***	**Due Date**	**Phone/Fax**

* Action Types: Approve, Review, Inform, File, Action Required, Attend Meeting, Other (please specify)

Version History

Ver. No.	**Ver. Date**	**Revised By**	**Description**
1.0	23.06.2015	Erdem Beğenilmiş	First version
1.1	28.08.2022	Bahar Yılmaz	General control
2.0	30.01.2024	Bahar Yılmaz	Document updated to include all hash versions and phase out of Version 1
2.1	06.02.2024	Bahar Yılmaz	Removed previous hash versions

Proprietary Notice

The information contained in all sheets of this document, proposal, or quotation constitutes trade secrets and/or information that is commercial or financial and is deemed confidential or privileged. It is furnished to prospective customer in confidence with the understanding that prospective customer will not, without the permission of Payten Teknoloji A.Ş. (from now on called Payten), use or disclose for other than evaluation purposes the information contained herein which is solely confidential and proprietary to Payten ("***Payten Confidential Information***"). In the event a contract is awarded on the basis of this document, proposal, or quotation, prospective customer shall have the right to use and disclose Asseco-See Confidential Information to the extent provided in the contract. The restriction does not limit prospective customer right to use or disclose such information if obtained from another source without restriction.

Contents

1.Genel Hash Kullanımı

Nestpay Gate işlemlerinde, Kullanıcı Doğrulaması için Hash yöntemi kullanılmaktadır. Bu dökümanda Hash Versiyon 3'ün nasıl kullanılacağı anlatılmaktadır.

Nestpay'de üç tane Hash Versiyonu desteklenmektedir. Bunlardan Hash Versiyon 1, SHA-1 algoritması ile base64 encode edilmekte iken; Hash Versiyon 2 & Hash Versiyon 3, SHA-512 algoritması ile base64 encode edilmektedir.

İşlem yapılırken, hangi hash versiyonun seçileceğinin kararı, işlem isteğinde gelen "hashAlgorithm" parametresine göre yapılmaktadır. Versiyon 1 için "hashAlgorithm" parametresi ile "ver1" değeri beklenmekteyken, Versiyon 2 için "hashAlgorithm" parametresi ile "ver2" değeri ve Versiyon 3 için "ver3" değeri beklenmektedir.

Hash değeri hesaplandıktan sonra Nestpay'e ilgili değerin, "hash" parametresi ile bildirilmesi beklenmektedir.

UYARI: PCI düzenlemeleri gereği, Hash Versiyon 1'in kullanımı sonlandırılmıştır. Eğer halen Versiyon 1'i kullanıyorsanız, Versiyon 3'e geçiş yapmanız zorunludur. Versiyon 3, Hash Versiyon 2'den çok daha üstün, esnek ve güvenli bir yapıya sahiptir. Tüm işyerlerinin hemen Versiyon 3'e geçiş yapması gerekmektedir. Bu değişikliği ihmal etmeniz durumunda ciddi güvenlik riskleri, uyumluluk sorunları ve işlem kesintisi ile karşılaşabilirsiniz

Hash değeri hesaplandıktan sonra Nestpay'e ilgili değerin, **"hash"** parametresi ile bildirilmesi beklenmektedir.

2.Hash Version 3

2.1 Hash Versiyon 3 Kullanarak Nestpay'e İstek Gönderimi

Hash Versiyon 3 için oluşturulan data'da "|" karakteri, parametreler arasında ayıraç görevi görmektedir. Hash Versyion 3 için hash'lenecek data oluşturulurken, Nestpay'e gönderilen tüm parametreler hash hesaplaması için kullanılır. Bu parametreler hash hesaplamasına sokulurken, storeKey haricinde parametre isimleri alfabetik olarak A'dan Z'ye sıralanır ve aralarına "|" ayıracı konularak ilgili alfabetik sırayla hash'lenecek data oluşturulur. Hash'lenecek data hazırlanırken, eğer bir parametre Nestpay'e boş olarak gönderiliyor olsa bile ilgili data'ya eklenilir (Boş değer için aşağıdaki örnekteki, Instalment parametresinin hash hesaplanırken ki kullanılışına bakılabilir).

Daha sonra alfabetik olarak hazırlanan datanın sonuna yine "|" ayıracı kullanılarak İşyeri Güvenli Anahtarı (storeKey) eklenir.

Önemli Not : Eğer parametrelerin değerinin içinde, "|" karakteri kullanılıyorsa, bu karakterin aynı zamanda parametreleri ayırmakta kullanılan "|" karakteri ile karıştırılmaması amacıyla, parametrenin değerinde olan "|" karakteri, hash datası oluşturulurken "\|" olarak değiştirilmektedir. Ek olarak, parametrelerin değerinde eğer "\" karakteri varsa, bu karakterinde karıştırılmaması için ilgili "\" karakterlerin de "\\" değeri ile değiştirilmesi gerekmektedir. Örneğin,

Original Value : ORDER-256712jbs\j6b|
Value used for Hash Calculation : ORDER-256712jbs\\j6b\|

Örnek Parametreler ve Hash Kullanımı:

Aşağıdaki sadece bir örnektir, kendi hesaplamanızda gönderdiğiniz tüm parametreleri eklemeniz gerekmektedir.

clientId	100200127
amount	95.93
okurl	http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler
failUrl	http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler
TranType	Auth
Instalment	
callbackUrl	http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp
currency	949
rnd	87954458746
storeType	3D
lang	tr
hashAlgorithm	ver3
BillToName	name
BillTocompany	billToCompany
refreshTime	5
storeKey	TEST1234

- **Hash**

Sample order of Used Parameters in Hash Data :

amount|BillToCompany|BillToName|callbackUrl|clientid|currency|failUrl|hashAlgorithm|Instalment|lang|okurl|refreshtime|rnd|storetype|TranType|storeKey

Plaintext:
95.93|billToCompany|name|http://localhost:8080/SampleCodeJSPTest/GateResponseControl.jsp|100200127|949|http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler|ver3||tr|http://localhost:8080/SampleCodeJSPTest/GenericVer3ResponseHandler|5|87954458746|3D|Auth|TEST1234

Hash3 = Base64(SHA512(plaintext))

Önemli Not II : "encoding" & "hash" isimli parametreler hash hesaplanmasında hesaba katılmayacaklardır.

2.2 Hash Versiyon 3 Kullanılan Bir İsteğin Yanıtında Hash Kontrolünün Yapılması

Hash Versiyon 3 kullanılarak yapılan bir isteğin sonuçlanmasının ardından, üye işyeri'ne geri bildirim yapıldığında, Üye İşyeri'nin isteğin gerçekten NestPay'den gelip gelmediğini anlayabilmesi için Hash Kontrolü yapması gerekmektedir. Bunun için NestPay, tarafından gönderilen hash değeri "HASH" parametre ismi ile Üye İşyeri sistemine yönlendirilmektedir.

Burada ilgili Hash hesaplanırken, kullanılan yaklaşım NestPay'e istekte bulunurken ki ile aynıdır. Üye İşyeri, NestPay tarafından kendisine gönderilen parametreleri alfabetik sıraya göre hesaplanacak Hash Data'sına "|" ayıracı kullanarak ekler. Üye İşyerine NestPay tarafından dönülen tüm parametreler alfabetik olarak Hash Datası'na eklendikten sonra, ilgili data'ya "|" ayıracı kullanılarak İşyeri Güvenli Anahtarı (storeKey) eklenir ve bu data SHA-512 algoritması ile base64 encode edilerek hash değeri elde edilir.

Önemli Not: "encoding" & "hash" & "countdown" isimli parametreler hash hesaplanmasında hesaba katılmayacaklardır.

2.2.1 Özelleştirilmiş Template'ların Hash Versiyon 3 ile Kullanımı

Eğer Üye İşyeri, Gate'e yapılan istek sonuçlarının bildirim öncesi için NestPay'e özelleştirilmiş bir template yüklemişse ve Hash Versiyon 3 kullanıyorsa, ilgili template'te post edilecek parametrelerin sadece NestPay'in @@**allhiddenparams**@@ bölümünde değiştireceği parametreler olması gerekmektedir. İlgili template'te kesinlike, bu parametreler dışında post edilecek parametreler bulunmamalıdır. Bulunması halinde, Gate'den gönderilecek Hash değeri ile, Üye İşyeri tarafında hesaplanan Hash değeri aynı olmayacaktır.

Bu durumda, eğer Üye İşyeri Hash Versiyon 1 ve Hash Versiyon 2 kullandığı zamanlardaki Özelleştirilmiş Template'lerini kaldırmadan, sadece Hash Versiyon 3 için yeni bir özelleştirilmiş

template'i sisteme yüklemek isterse, ilgili template'i, isminin sonu **"_ver3.htm"** olacak şekilde adlandırabilir **(Örneğin, gateresultlogosuccess_ver3.htm).**

UYARI: Hash Versiyon 1 ve Hash Versiyon 2 için aynı template'de çalışır, ancak Hash Versiyon 1 KULLANILMAMALIDIR.

3.Hash Version 3 Örnek Kodlar

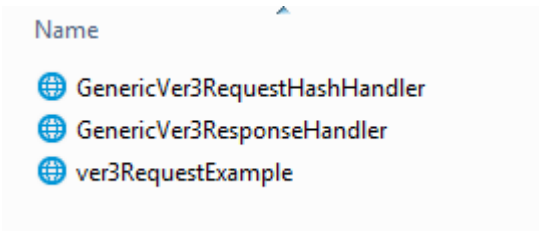
.Net C#, .Net VB, JSP ve PHP dillerinde olmak üzere, Nestpay'e Hash Versiyon 3 kullanılarak İstek'te bulunurken kullanıbilecek ve ilgili İşlem'in Sonucu bildirildiğinde yanıtın NestPay'den gelip gelmedğini doğrulayabilecek örnek kodlar **"SampleCodes.zip"** dosyası verilmiştir.

Bu kodlar, her 4 dil için de üç bölümden oluşturulmaktadır.

1. İstek Data'sının hazırlandığı Bölüm (Data Hazırlama)
2. NestPay'e iletilen parametre'lerle birlikte Hash'in Hesaplandığı Bölüm (İstek için Hash Hesaplama)
3. İşlem Sonucu Bildirildiğinde Üye İşyeri Tarafından Bildirimin NestPay'den yapılıp,yapılmadığının kontrolü (İşlem Sonuç Bildirimi Hash Kontrolü)

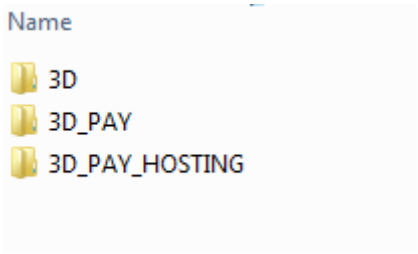
Aşağıdaki resimden görüleceği üzere, Kod dizini içinde her 4 dil için 3 adet örnek kod bulunmaktadır. Bu kodların, yukarıdaki üç bölümden hangilerine tekabül ettiğini belirtmemiz gerekirse;

- **Bölüm 1 :** ver3RequestExample
- **Bölüm 2 :** GenericVer3RequestHashHandler
- **Bölüm 3 :** GenericVer3ResponseHandler

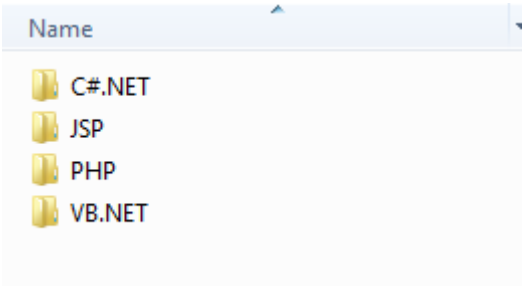


Verilen örnek kodlar, Data Hazırlanması yapıldıktan sonra ilgili isteğin NestPay'e gönderilmeden önce, tüm parametrelerin hash'inin hesaplanabilmesi için ara bir katman olarak başka bir sayfaya yönlendirilip (Bölüm 2 - İstek için Hash Hesaplama), o sayfadan NestPay'e yönlendirileceği varsayılarak hazırlanmıştır.

İlgili Örnek Kodlar, dökümanla beraber "SampleCodes.zip" dosyası ile verilmiştir. Kodlar işyeri entegrasyon tipine göre "3D", "3D_PAY", "3D_PAY_HOSTING" olarak SampleCodes klasörü içinde 3'e ayrılmıştır.



Her klasörün içinde de, kodun yazılı olduğu dili belirten 4 klasör vardır (C#.NET, JSP, PHP, VB.NET).



Entegrasyon yapacak üye işyeri, kendi entegrasyon tipine ve kullanacağı yazılım diline göre ilgili klasördeki kod örneğine bakabilir.

NOT I : Örnek kodlarda bulunan ***https://<host_address>*** adresi, ilgili kod Üye İşyeri'nin sisteminde, hangi adrese deploy edilecekse, o adresle değiştirilmelidir.

NOT II : Örnek kodlarda bulunan ***https://<host_address>/<3dgate_path>*** adresi, ilgili Üye İşyeri, hangi adresteki NestPay Gate'ine istekte bulunacaksa, onunla değiştirilmelidir.