8/24/2018

Zhiharevich Yury

Nokian Tyres

CR\_6 US\_Tyre+\_eHG\_6 Integration interaction of the purchase registration for direct integration

Оглавление.

[Оглавление 1](#_Toc94091776)

[1. Общее описание. 2](#_Toc94091777)

[1.1 Спецификация T+ интерфейс партнеров. 2](#_Toc94091778)

[1.2 Границы реализации. 2](#_Toc94091779)

[1.3 Особенности реализации. 2](#_Toc94091780)

[1.4 Аутентификация и доступ. 2](#_Toc94091781)

[1.5 HTTP Header. 3](#_Toc94091782)

[1.6 HTTP Response. 3](#_Toc94091783)

[2. Модель данных 3](#_Toc94091784)

[3. Функция "Регистрация новой покупки". 4](#_Toc94091785)

[3.0.1 Запрос на проверку участия шины в кампании 4](#_Toc94091786)

[3.0.2 Запрос на проверку существования покупателя. 5](#_Toc94091787)

[3.1 Запрос на создание покупателя. 6](#_Toc94091788)

[3.2 Запрос на создание набора шин. 7](#_Toc94091789)

[3.3. Запрос на создание электронного талона. 7](#_Toc94091790)

[4.0. Запрос на получение информации о текущем статусе талона по связке Клиент-Набор Шин. 9](#_Toc94091791)

[Все запросы данной функции передаются после отправки продавцом формы регистрации покупки. 9](#_Toc94091792)

[Отправка запросов происходит последовательно в указанном порядке. Для генерации и отправки последующего запроса необходимо получить ответ на предыдущий, так как в ответах содержатся значения необходимых ключей. 9](#_Toc94091793)

# 1. Общее описание.

### 1.1 Спецификация T+ интерфейс партнеров****.****

Данная документация основывается на спецификации **T+ partner interfaces.**Спецификация описывает основные принципы и форматы интеграционного взаимодействия внешних систем с глобальной интеграционной платформой **SOA Suite** в рамках реализации проекта Nokian Tyres GSP T+.

### 1.2 Границы реализации.

Данная US описывает формат интеграционного взаимодействия между системами партнеров и решением **SOA Suite**.   
В наборе функций последовательно описывается формат запросов-ответов, необходимых для реализации функционала Регистрация покупки сотрудником УХГ в своих системах.

### 1.3 Особенности реализации.

Для реализации интеграционного взаимодействия используется двухсторонняя интеграция в формате REST API между системами партнеров и решением **SOA Suite**. Взаимодействие происходит с помощью веб-сервисов с использованием API как на стороне системы партнера, так и на стороне **SOA Suite**.    
Взаимодействие с веб-сервисами осуществляются по протоколу HTTPS. Поддерживаются HTTP-запросы.     
**SOA Suite** предоставляет конечную точку веб-служб REST, расположенную на URL-адресах HTTPS. Используя эту конечную точку, система партнера может отправлять HTTP-запросы и получать синхронные HTTP-ответы.

Формат сообщений: XML payload  
Требуемая кодировка запроса и ответа: UTF-8.

### 1.4 Аутентификация и доступ.

Аутентификационные данные (логин, пароль) вносятся в header каждого запроса.   
Используется тип авторизации - Basic Authorization.   
Используется стандарт кодирования - Base64.  
Все запросы должны идти через ssl/tls.

Список сред:

|  |  |
| --- | --- |
| Environment | URI prefix |
| DEV | https://devextranet.nokiantyres.com/services/gsp |
| TEST | https://testextranet.nokiantyres.com/services/gsp |
| PROD | https://extranet.nokiantyres.com/services/gsp |

### 1.5 HTTP Header.

Все http-запросы должны иметь следующие заголовки:

|  |  |  |
| --- | --- | --- |
| Header | Значение | Описание |
| Content-type | application/xml; charset=UTF-8 | Поддерживаемый формат. |
| Authorization | Basic bXl1c2VyOm15cGFzcw== | Логин и пароль в кодировке Base64. |
| PosCode | POS123 | Партнерский ID POSа |

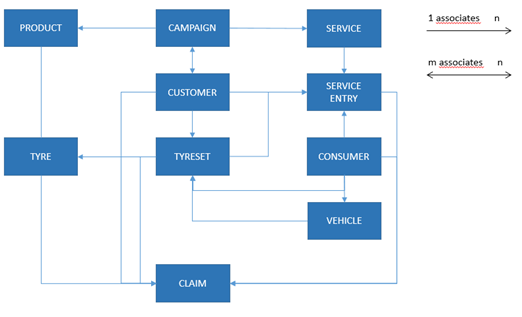
### 1.6 HTTP Response.

* 200 – “OK”
* 204 – “No Content” – В запросе не найдено подходящего параметра или объекта для обновления
* 400 – “Bad Request” – Некорректные параметрами запроса при создании / обновлении объекта
* 401 – “Unauthorized” – Некорректные логин и пароль
* 403 – “Forbidded” – Возможно был указан некорректный POSID
* 404 – “Not Found” – Некорректные параметрами запроса при создании / обновлении объекта
* 500 – “Internal Server Error” – Необрабатываемая ошибка сервера

# 2. Модель данных

T+ интерфейс партнеров предоставляет сервисы по следующим сущностям: Потребитель, Автомобиль, Комплект шин и Шина (Consumer, Vehicle, Tyre set and Tyre).   
 Потребитель имеет дочерние сущности: Автомобиль, Шины и Комплекты шин.

Диаграмма "Модель сущностей" показывает модель данных и связи.



# 3. Функция "Регистрация новой покупки".

### 3.0.1 Запрос на проверку участия шины в кампании

**Тип запроса – GET.**

Запрос требуется выполнять перед регистрацией продажи для понимания необходимости регистрировать данную покупку от данного POS по данному артикулу на текущую дату.

Требования:

itemCode (Артикул) и type=Guarantee должны быть указаны в запросе.

*Пример запроса:*

URI: /QueryCampaigns/v1/xml?itemCode=TS32204&type=Guarantee

*Пример ответа:*

<?xml version="1.0" encoding="UTF-8"?>

<ns1:Campaigns xmlns:ns1="http://www.nokiantyres.com/QueryCampaigns/1.0">

<ns1:Campaign>

<ns1:CampaignID>7013N000000BrRgQAK</ns1:CampaignID>

<ns1:CampaignType>Guarantee</ns1:CampaignType>

<ns1:ServiceCode>TYRLLG</ns1:ServiceCode>

</ns1:Campaign>

</ns1:Campaigns>

Тело ответа:

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2018 sp1 (x64) (http://www.altova.com) by Saila (Solita Oy) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nt="http://www.nokiantyres.com/QueryCampaigns/1.0" targetNamespace="http://www.nokiantyres.com/QueryCampaigns/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--Root-->

<xs:element name="Campaigns" type="nt:CampaignsDataType">

<xs:annotation>

<xs:documentation>QueryCampaigns schema for GSP.</xs:documentation>

</xs:annotation>

</xs:element>

<!--Campaigns-->

<xs:complexType name="CampaignsDataType">

<xs:sequence>

<xs:element name="Campaign" type="nt:CampaignDataType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<!--Campaign-->

<xs:complexType name="CampaignDataType">

<xs:sequence>

<xs:element name="CampaignID" type="xs:string" minOccurs="1"/>

<xs:element name="CampaignType" type="xs:string" minOccurs="0"/>

<xs:element name="ServiceCode" type="xs:string" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

### 3.0.2 Запрос на проверку существования покупателя.

**Тип запроса – GET.**

Запрос требуется выполнять перед регистрацией продажи для проверки существующего пользователя. Если пользователь не найден (404 Not Found), то необходимо выполнить запрос 3.1, если найден – то переходить к запросу 3.2, подставляя в URI ConsumerID из ответа на данный запрос.

Требования:  
phone (Телефон потребителя) должен быть указаны в запросе.

*Пример запроса:*

URI: /QueryConsumers/v1/xml?phone=72224444444

*Пример ответа с ID покупателя:*

<?xml version="1.0" encoding="UTF-8"?>

<ns1:Consumer xmlns:ns1="http://www.nokiantyres.com/QueryConsumer/1.0">

<ns1:consumerID>0016E00000bcxnsQAA</ns1:consumerID>

<ns1:email/>

<ns1:phone/>

<ns1:firstName/>

<ns1:lastName/>

<ns1:addressStreet/>

<ns1:addressPostalCode/>

<ns1:addressCity/>

<ns1:addressCountry/>

<ns1:Vehicles/>

<ns1:Tyresets/>

</ns1:Consumer>

Описание параметров тела ответа:

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2015 sp2 (x64) (http://www.altova.com) by SLe (Solita Oy) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nt="http://www.nokiantyres.com/QueryConsumer/1.0" targetNamespace="http://www.nokiantyres.com/QueryConsumer/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--Root-->

<xs:element name="Consumer" type="nt:QConsumerMsgType">

<xs:annotation>

<xs:documentation>QueryConsumer schema for GSP.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="QConsumerMsgType">

<xs:sequence>

<xs:element name="consumerID" type="xs:string"/>

<xs:element name="email" type="xs:string"/>

<xs:element name="phone" type="xs:string"/>

<xs:element name="firstName" type="xs:string"/>

<xs:element name="lastName" type="xs:string"/>

<xs:element name="addressStreet" type="xs:string"/>

<xs:element name="addressCity" type="xs:string"/>

<xs:element name="addressCountry" type="xs:string"/>

<xs:element name="Vehicles" type="nt:VehiclesDataType" minOccurs="0" maxOccurs="unbounded"/>

<xs:element name="Tyresets" type="nt:TyresetsDataType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

## 3.1 Запрос на создание покупателя.

**Тип запроса – POST.**

**Создание потребителя происходит с указанием +7 перед номером телефона:** **+79999999999**

*Пример запроса:*

**URI**: /CreateConsumer/v1/xml

<xs:Request xmlns:xs="http://www.nokiantyres.com/CreateConsumer/1.0">

<xs:phone>+71115564811</xs:phone>

<xs:lastName>+71115564811</xs:lastName>

</xs:Request>

*Пример ответа:*

<?xml version="1.0" encoding="UTF-8"?>

<ns1:Response xmlns:ns1="http://www.nokiantyres.com/CreateConsumer/1.0">

<ns1:consumerID>0016E00000b0NIXQA2</ns1:consumerID>

<ns1:email/>

<ns1:phone>**+**71115564811</ns1:phone>

<ns1:firstName/>

<ns1:lastName>**+**71115564811</ns1:lastName>

<ns1:addressStreet/>

<ns1:addressPostalCode/>

<ns1:addressCity/>

<ns1:addressCountry/>

</ns1:Response>

Описание параметров тела запроса и ответа:

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2018 sp1 (x64) (http://www.altova.com) by Saila (Solita Oy) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nt="http://www.nokiantyres.com/CreateConsumer/1.0" targetNamespace="http://www.nokiantyres.com/CreateConsumer/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--Root-->

<xs:element name="Request" type="nt:RequestConsumer">

<xs:annotation>

<xs:documentation>CreateConsumer schema for GSP. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="RequestConsumer">

<xs:sequence>

<xs:element name="consumerID" type="xs:string"/>

<xs:element name="email" type="xs:string"/>

<xs:element name="phone" type="xs:string"/>

<xs:element name="firstName" type="xs:string"/>

<xs:element name="lastName" type="xs:string"/>

<xs:element name="addressStreet" type="xs:string"/>

<xs:element name="addressPostalCode" type="xs:string"/>

<xs:element name="addressCity" type="xs:string"/>

<xs:element name="addressCountry" type="xs:string"/>

</xs:sequence>

</xs:complexType>

<xs:element name="Response" type="nt:ResponseConsumer">

<xs:annotation>

<xs:documentation>CreateConsumer schema for GSP. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="ResponseConsumer">

<xs:sequence>

<xs:element name="consumerID" type="xs:string"/>

<xs:element name="email" type="xs:string"/>

<xs:element name="phone" type="xs:string"/>

<xs:element name="firstName" type="xs:string"/>

<xs:element name="lastName" type="xs:string"/>

<xs:element name="addressStreet" type="xs:string"/>

<xs:element name="addressPostalCode" type="xs:string"/>

<xs:element name="addressCity" type="xs:string"/>

<xs:element name="addressCountry" type="xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

## 3.2 Запрос на создание набора шин.

**Тип запроса – POST.**

Запрос необходимо выполнять для создания набора шин, приобретаемых покупателем.

Требования:

Если заданный consumerID не соответствует ни одному из существующих ID, то запрос вернет ошибку 403.

*Пример запроса:*

**URI: /CreateTyreSets/v1/xml**

<xs:Request xmlns:ns="http://www.nokiantyres.com/CreateTyreSets/1.0">

<xs:consumerID>0016E00000cmnzeQAA</xs:consumerID>

<xs:itemCode>TS31872</xs:itemCode>

<xs:tyreCount>3</xs:tyreCount>

<xs:purchaseDate>2018-10-04</xs:purchaseDate>

</xs:Request>

*Пример ответа:*

<?xml version="1.0" encoding="UTF-8"?>

<ns1:Response xmlns:ns1="http://www.nokiantyres.com/CreateTyreSets/1.0">

<ns1:vehicleID/>

<ns1:tyreSetID>a1q6E000000eSbOQAU</ns1:tyreSetID>

<ns1:itemCode>TS31872</ns1:itemCode>

<ns1:tyreCount>3</ns1:tyreCount>

<ns1:purchaseDate>2018-10-04</ns1:purchaseDate>

</ns1:Response>

Описание параметров тела запроса и ответа:

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2018 sp1 (x64) (http://www.altova.com) by Saila (Solita Oy) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nt="http://www.nokiantyres.com/CreateTyreSets/1.0" targetNamespace="http://www.nokiantyres.com/CreateTyreSets/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--Root-->

<xs:element name="Request" type="nt:RequestTyreSets">

<xs:annotation>

<xs:documentation>CreateTyreSets schema for GSP. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="RequestTyreSets">

<xs:sequence>

<xs:element name="consumerID" type="xs:string" minOccurs="0"/>

<xs:element name="vehicleID" type="xs:string" minOccurs="0"/>

<xs:element name="tyreSetID" type="xs:string" minOccurs="0"/>

<xs:element name="itemCode" type="xs:string" minOccurs="0"/>

<xs:element name="tyreCount" type="xs:string" minOccurs="0"/>

<xs:element name="purchaseDate" type="xs:date" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<xs:element name="Response" type="nt:ResponseTyreSets">

<xs:annotation>

<xs:documentation>CreateTyreSets schema for GSP. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="ResponseTyreSets">

<xs:sequence>

<xs:element name="vehicleID" type="xs:string" minOccurs="0"/>

<xs:element name="tyreSetID" type="xs:string" minOccurs="0"/>

<xs:element name="itemCode" type="xs:string" minOccurs="0"/>

<xs:element name="tyreCount" type="xs:string" minOccurs="0"/>

<xs:element name="purchaseDate" type="xs:date" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

## 3.3. Запрос на создание электронного талона.

**Тип запроса – POST.**

Запрос необходим для создания в системе Nokian Tyres электронного талона и отправки смс о необходимости регистрации гарантии конечному потребителю. Необходимо передавать статус "RESERVED" в теле запроса.

*Пример запроса:*

**URI: /CreateServiceEntry/v1/xml**

***Пример запроса:***

<xs:Request xmlns:ns="http://www.nokiantyres.com/CreateServiceEntry/1.0">

<xs:consumerID>0016E00000aNAc2QAG</xs:consumerID>

<xs:tyreSetID>a1q6E000000ayWeQAI</xs:tyreSetID>

<xs:entryType>Guarantee</xs:entryType>

<xs:status>RESERVED</xs:status>

</xs:Request>

Пример ответа:

<?xml version="1.0" encoding="UTF-8"?>

<ns1:Response xmlns:ns1="http://www.nokiantyres.com/CreateServiceEntry/1.0">

<ns1:ServiceEntry>

<ns1:serviceEntryID>a2a3N000000delbQAA</ns1:serviceEntryID>

<ns1:serviceCode>TYRLLG</ns1:serviceCode>

<ns1:serviceTerms/>

<ns1:serviceInstructions/>

<ns1:validFrom>2022-03-03</ns1:validFrom>

<ns1:validTo>2032-03-03</ns1:validTo>

</ns1:ServiceEntry>

</ns1:Response>

Описание параметров тела запроса и ответа:

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2018 sp1 (x64) (http://www.altova.com) by Saila (Solita Oy) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nt="http://www.nokiantyres.com/CreateServiceEntry/1.0" targetNamespace="http://www.nokiantyres.com/CreateServiceEntry/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--Root-->

<xs:element name="Request" type="nt:RequestServiceEntry">

<xs:annotation>

<xs:documentation>CreateServiceEntry schema for GSP. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="RequestServiceEntry">

<xs:sequence>

<xs:element name="consumerID" type="xs:int" minOccurs="0"/>

<xs:element name="tyreSetID" type="xs:int" minOccurs="0"/>

<xs:element name="entryType" type="xs:string" minOccurs="0"/>

<xs:element name="serviceEntryID" type="xs:int" minOccurs="0"/>

<xs:element name="status" minOccurs="0">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="RESERVED"/>

<xs:enumeration value="REGISTERED"/>

<xs:enumeration value="CONSUMED"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

</xs:sequence>

</xs:complexType>

<xs:element name="Response" type="nt:ResponseServiceEntriesType">

<xs:annotation>

<xs:documentation>CreateServiceEntry schema for GSP. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:complexType name="ResponseServiceEntriesType">

<xs:sequence>

<xs:element name="ServiceEntry" type="nt:ServiceentryDataType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name="ServiceentryDataType">

<xs:sequence>

<xs:element name="serviceEntryID" type="xs:string " minOccurs="1"/>

<xs:element name="serviceCode" type="xs:string" minOccurs="0"/>

<xs:element name="serviceTerms" type="xs:string" minOccurs="0"/>

<xs:element name="serviceInstructions" type="xs:string" minOccurs="0"/>

<xs:element name="validFrom" type="xs:date" minOccurs="0"/>

<xs:element name="validTo" type="xs:date" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

После исполнения данного запроса покупатель получает SMS с сылкой для активации гарантии. После активации гарантии статус становится Registered. Пока статус Reserved покупатель не может воспользоваться услугами расширенной гарантии.

# 4. Запрос на получение информации о текущем статусе талона по связке Клиент-Набор Шин.

**Тип запроса – GET.**

Запрос необходим для получения актуальной информации по талонам. Необходимо передать ConsumerID, TyreSetID и entryType, который может быть только "Guarantee"

*Пример запроса:*

**URI: /QueryServiceEntries/v1/xml?consumerID=123&TyreSetID=333&entryType=123**

Описание параметров тела запроса и ответа:

<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XMLSpy v2018 sp1 (x64) (http://www.altova.com) by Saila (Solita Oy) -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nt="http://www.nokiantyres.com/QueryServiceEntries/1.0" targetNamespace="http://www.nokiantyres.com/QueryServiceEntries/1.0" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--Root-->

<xs:element name="Serviceentries" type="nt:ServiceentriesDataType">

<xs:annotation>

<xs:documentation>QueryServiceEntries schema for Tyre+. </xs:documentation>

</xs:annotation>

</xs:element>

<!--Serviceentries-->

<xs:complexType name="ServiceentriesDataType">

<xs:sequence>

<xs:element name="Serviceentry" type="nt:ServiceentryDataType" minOccurs="0" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<!--Serviceentry-->

<xs:complexType name="ServiceentryDataType">

<xs:sequence>

<xs:element name="serviceEntryID" type="xs:int" minOccurs="1"/>

<xs:element name="entryType" minOccurs="0">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="Guarantee"/>

<xs:enumeration value="Coupon"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

<xs:element name="entryStatus" type="xs:string" minOccurs="0"/>

<xs:element name="serviceCode" type="xs:string" minOccurs="0"/>

<xs:element name="serviceTerms" type="xs:string" minOccurs="0"/>

<xs:element name="serviceInstructions" type="xs:string" minOccurs="0"/>

<xs:element name="campaignID" type="xs:string" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

**Все запросы данной функции передаются после отправки продавцом формы регистрации покупки.**

**Отправка запросов происходит последовательно в указанном порядке. Для генерации и отправки последующего запроса необходимо получить ответ на предыдущий, так как в ответах содержатся значения необходимых ключей.**