

# COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME

CIP-ICT-PSP-2013-7 Pilot Type B



WP4 – Data analysis, modelling and synthesis

D4.3.2 Event-based notification services

Deliverable Lead: ATOS

Deliverable due date: 29/02/2016

Actual submission date: 29/02/2016

Version: 1.4



**Document Control Page**

<b>Title</b>	D4.3.2 Event-based notification services
<b>Creator</b>	Miguel Ángel Esbrí (ATOS)
<b>Description</b>	This deliverable, D4.3.2 Event based notification services, is the result of task T4.4 Reporting and alerts, which aims at defining a set of components that enable FOODIE platform to generate - in an automatic manner - diverse notifications and reports for its users (principally for the farmers) based on the data collected and stored in its various repositories of information.
<b>Publisher</b>	FOODIE Consortium
<b>Contributors</b>	Raúl Palma (PSNC)
<b>Creation date</b>	23/09/2015
<b>Type</b>	Text
<b>Language</b>	en-GB
<b>Rights</b>	copyright "FOODIE Consortium"
<b>Audience</b>	<input type="checkbox"/> internal <input checked="" type="checkbox"/> public <input type="checkbox"/> restricted
<b>Review status</b>	<input type="checkbox"/> Draft <input checked="" type="checkbox"/> WP leader accepted <input checked="" type="checkbox"/> Technical Manager accepted <input checked="" type="checkbox"/> Coordinator accepted
<b>Action requested</b>	<input type="checkbox"/> to be revised by Partners <input type="checkbox"/> for approval by the WP leader <input type="checkbox"/> for approval by the Technical Committee <input type="checkbox"/> for approval by the Project Coordinator
<b>Requested deadline</b>	

**STATEMENT FOR OPEN DOCUMENTS**

(c) 2016 FOODIE Consortium

The *FOODIE* Consortium (<http://www.foodie-project.eu>) grants third parties the right to use and distribute all or parts of this document, provided that the *FOODIE* project and the document are properly referenced.

THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. EXCEPT WHAT SET FORTH BY MANDATORY PROVISIONS OF LAW IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**About the project**

FOODIE project aims at creating a platform hub on the cloud where spatial and non-spatial data related to agricultural sector is available for agri-food stakeholders groups and interoperable. It will offer: an infrastructure for the building of an interacting and collaborative network; the integration of existing open datasets related to agriculture; data publication and data linking of external agriculture data sources, providing specific and high-value applications and services for the support of planning and decision-making processes.

FOODIE project is addressed to four basic groups of users: a) stakeholders from the agriculture sector as end-users of final applications, b) public sector for communication with farmers about taxation, subsidies and regulation, c) researchers for large scale experimentation on real data and d) ICT companies for the development of new applications for agriculture and food sector, mainly using implemented tools

FOODIE specifically works on three pilots:

- Pilot 1: Precision Viticulture (Spain) will focus on the appropriate management of the inherent variability of crops,
- Pilot 2: Open Data for Strategic and Tactical Planning (Czech Republic) will focus on improving future management of agricultural companies (farms) by introducing new tools and management methods,
- Pilot 3: Technology allows integration of logistics via service providers and farm management including traceability (Germany).

**Contact information**

Miguel Angel Esbri

Project Coordinator

Atos Spain, Madrid, Spain

E-mail: [miguel.esbri@atos.net](mailto:miguel.esbri@atos.net)

URL: <http://www.foodie-project.eu>

Twitter: [https://twitter.com/FOODIE\\_Project](https://twitter.com/FOODIE_Project)

## Table of Contents

Glossary.....	5
Abbreviations and Acronyms.....	6
Executive Summary .....	7
<b>1 Introduction.....</b>	<b>8</b>
<b>2 Notification Broker Service .....</b>	<b>9</b>
2.1 Service resource objects .....	10
2.1.1 Capabilities resource .....	10
2.1.2 Consumer resource.....	12
2.1.3 Subscription resource .....	12
2.1.4 Notification resource .....	13
2.2 Service operations .....	16
2.2.1 Capabilities API .....	16
2.2.2 Consumer API .....	17
2.2.3 Producer API .....	18
2.2.4 Admin API .....	18
<b>3 Service feed extension .....</b>	<b>19</b>
3.1 Service resources and links.....	19
3.1.1 Service description.....	19
3.1.2 Notification resource .....	19
3.1.3 Service operations .....	19
<b>4 Conclusions.....</b>	<b>21</b>
References .....	22
Annex A – API documentation with Swagger .....	23
Annex B – REST operations examples .....	24

## Index of Figures

Figure 1: Notification Broker Service - High-level view .....	9
---	---

## Index of Tables

Table 1: Abbreviations and Acronyms .....	6
<b>Table 2:</b> Capabilities resource description .....	11
<b>Table 3:</b> Consumer resource description.....	12
<b>Table 4:</b> Subscription resource description .....	13
<b>Table 5:</b> Common notification resource attributes .....	15
<b>Table 6:</b> Producer notification resource attributes .....	15
<b>Table 7:</b> Consumer notification resource attributes .....	16
<b>Table 8:</b> Capabilities resource operations .....	17
Table 9: Consumer API operations .....	18
<b>Table 10:</b> Producer API operations.....	18
<b>Table 11:</b> Admin API operations .....	18

## Glossary

The glossary of terms used in this deliverable can be found in the public document “FOODIE\_Glossary.pdf” available at: <http://www.foodie-project.eu>

## Abbreviations and Acronyms

Abbreviation / Acronym	Description
API	Application Programming Interface
EU	European Union
FOODIE	Farm Oriented Open Data in Europe
FP	Framework Program
GeoJSON	Geographical JavaScript Object Notation
JSON	JavaScript Object Notation
REST	In computing, representational state transfer (REST) is the software architectural style of the World Wide Web.
SMS	Short Message Service, a form of text messaging communication on phones and mobile phones

**Table 1:** Abbreviations and Acronyms

## Executive Summary

This deliverable defines a set of components that enable FOODIE platform to generate - in an automatic manner - diverse notifications and reports for its users (principally for the farmers) based on the data collected and stored in its various repositories of information.

The second version of the document comprises three main sections. The first section contains a brief introduction into the main focus of the document for this first version of the deliverable that is the specification and implementation of the main component in this task: the notification broker service. The second section defines the high-level view of the notification broker as well as all the resources that it manages (Capabilities, Consumer, Subscription, Producer and Consumer notifications) and its REST interfaces. Please, note that Annexes A and B provide a detailed list of supported REST operations and usage examples.

Further iterations of this deliverable will be aimed at describing the new features and enhancements presented in the conclusions section (fourth section).

## 1 Introduction

This deliverable, D4.3.2 Event based notification services, is the result of task T4.4 Reporting and alerts, which aims at defining a set of components that enable FOODIE platform to generate - in an automatic manner - diverse notifications and reports for its users (principally for the farmers) based on the data collected and stored in its various repositories of information.

To that end, the deliverable focuses on the description and refinement (in an incremental manner along its three releases) of the business logic, data structure formats as well as the detailed APIs of the notification and report generation software components/services.

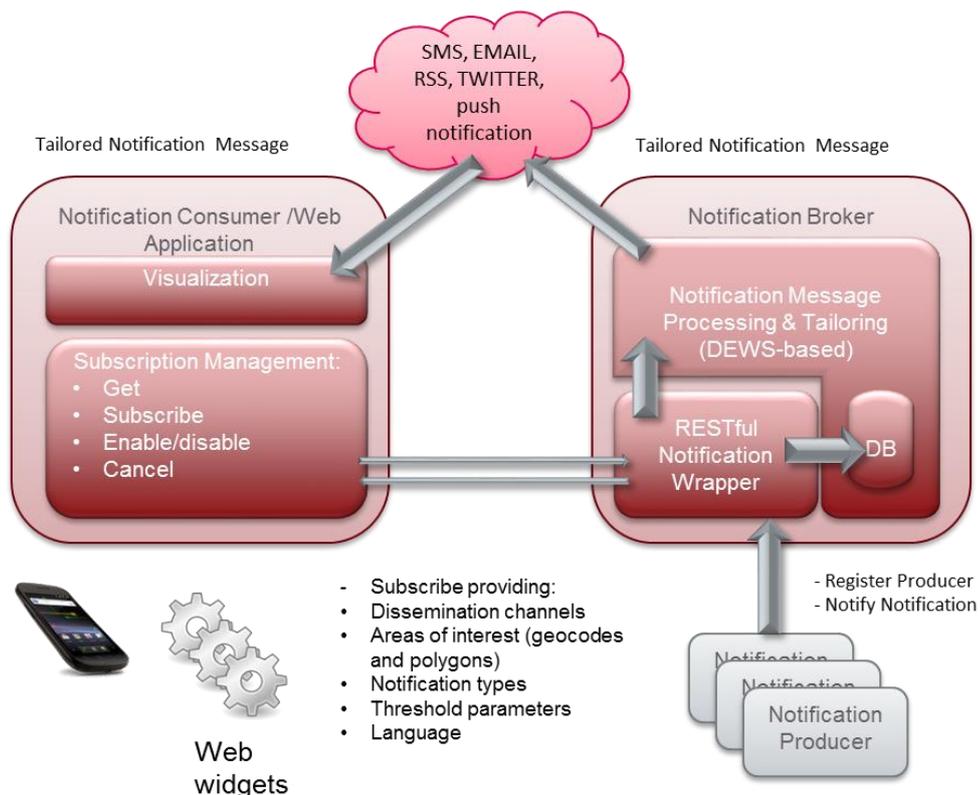
The first deliverable version presented in detail the Notification Broker Service (see section 2), which is an adaptation and enhancement of the notification component developed in FP6 DEWS (Distant Early Warning System) project [1]. This second version provides new information regarding the updates made in the REST interfaces of the service.

## 2 Notification Broker Service

FOODIE Notification Broker Service builds on the adaptation and enhancement of the notification component developed in FP6 DEWS (Distant Early Warning System) project. This notification component was initially developed for issuing hazard-related warnings (e.g., earthquake and tsunami alerts) but has been adapted to be application domain independent, therefore allowing for instance to issue notifications with agriculture-based information that is relevant for FOODIE platform subscribers.

In a general manner, the service allows that registered users in FOODIE platform subscribe to different subscription “topics” defined by the service (e.g., meteorological alerts, specific crop plague alerts, monthly reporting notifications, etc.). The generation of these notifications is typically initiated by some data pushed to the notification service (referred to an existing topic) by some other component located in FOODIE platform (e.g., recommenders, sensors, processing services, etc.), although in the future it is planned that external applications registered in the platform as “notification producers” are able to define their own notification topics and thus enable them to push notifications to their topics’ subscribers.

Notifications are customized and delivered to topic subscribers through different communication channels (according to the users’ subscription preferences such as language, communication channel or topic specific options that might be defined in the future (e.g., include as PDF, XLS or CSV attachment the monthly generated report), being currently supported email and SMS message notifications. For future releases it is also planned to support Google Cloud Messaging (notifications delivered to Android devices) as well as social sites such as Twitter, Facebook and RSS feeds.



**Figure 1:** Notification Broker Service - High-level view

Next subsections describe in further detail the data structures, formats and service APIs that support all these features.

## 2.1 Service resource objects

The Notification Broker Service defines the following resources:

- Capabilities:** describes the features supported by the service, such as what topics are available for subscription, what attributes are defined by the topic (which in turn can be used by the subscriber to define restrictions. E.g., receive notifications from topic “meteorological alerts” if \$windSpeed > 14.0 km/h), in which languages can be received the notifications and what communication channels can be used to disseminate the notifications.
- Consumers:** describes the notification recipient (consumer) general attributes, such as the username, category (e.g., farmer, farmer pro, etc.), country code, preferred language of the notification messages and addresses to where the notification broker should send the notifications (currently SMS and email messages are supported).
- Subscriptions:** describes the subscription preferences of the user for a concrete topic.
- Notifications:** the notification broker distinguishes between two types of notifications, the original notification - and all the payload data that it contains- pushed by the producer (**ProducerNotification**) to the notification broker and the final notification customized by the notification broker and delivered to the user (**ConsumerNotification**).
- Producers** (in the future): describes information related to external FOODIE platform notification producers such as their producer id and topic definitions.

Next subsections describe in detail the structure of each of these resources.

### 2.1.1 Capabilities resource

Capabilities resource object and its sub-elements are described in the following table:

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
<b>Capabilities</b>		The container for all component parts of the Capabilities object (REQUIRED).	
<b>TopicDescriptors</b>		The container for all information related to the subscription topics supported by the Notification Broker (REQUIRED).	List of “TopicDescriptor” objects. The list <b>MUST</b> contain at least one TopicDescriptor
<b>Name</b>	Capabilities. TopicDescriptor	Identifier of the topic (REQUIRED), which will be used later on for subscribing to notifications related to that topic name.	Topic name identifiers <b>MUST</b> be URLs (not necessarily existing ones) in order to prevent the possibility of having two topics with the same name (from different notification producers). E.g.,: <a href="http://www.foodie-cloud.com/notification-broker/topic/meteorological-alert">http://www.foodie-cloud.com/notification-broker/topic/meteorological-alert</a>
<b>Description</b>	Capabilities TopicDescriptor	Textual information describing what is about this concrete topic (OPTIONAL)	
<b>ParameterDescriptors</b>	Capabilities. TopicDescriptor	A list of topic-specific parameter description that this topic supports. (OPTIONAL)	Typically topics may provide notifications with information
<b>Name</b>	Capabilities TopicDescription ParameterDescriptor	Name of the parameter supported by the topic (REQUIRED)	

<b>Description</b>	Capabilities. TopicDescriptor ParameterDescriptor	(OPTIONAL)	
<b>Type</b>	Capabilities TopicDescriptor ParameterDescriptor	Value type for this parameter (REQUIRED)	Currently supported parameter types are: Integer, Long, Float, Double, String, Boolean, Date, CodeList (enumerations)
<b>DefaultValue</b>	Capabilities TopicDescriptor ParameterDescriptor	Default value for this parameter in case no value is specified (OPTIONAL)	
<b>MinimumValue</b>	Capabilities TopicDescriptor ParameterDescriptor	Minimum value that can have this parameter (OPTIONAL)	
<b>MaximumValue</b>	Capabilities TopicDescriptor ParameterDescriptor	Maximum value that can have this parameter (OPTIONAL)	
<b>ValidValues</b>	Capabilities TopicDescriptor ParameterDescriptor	Finite list of values that may have this parameter (OPTIONAL)	
<b>Unit</b>	Capabilities TopicDescriptor ParameterDescriptor	Units of measurement for this parameter (OPTIONAL)	If any, it is commonly used with parameters of type Integer, Long, Float or Double.
<b>Required</b>	Capabilities TopicDescriptor ParameterDescriptor	Indicates whether this parameter must be included in a notification referred to this topic (OPTIONAL)	Default is FALSE
<b>FeatureDescriptor</b>	Capabilities. TopicDescriptor	Descriptor for features supported by this specific topic (OPTIONAL). See ParameterDescriptor for list of object properties.	Users subscribing to a topic might indicate whether for instance they want that the notification includes reports in pdf, csv, excel, et.
<b>IsoLanguageCode</b>	Capabilities	List of languages in which the notification message can be sent to the notification recipient (REQUIRED)	At least one language code must be supported by the notification broker service. Language codes are indicated according to the 2-digits ISO standard
<b>ChannelType</b>	Capabilities	List of supported dissemination channels that can be used by the notification broker to send the notification message to the recipient user (REQUIRED).	At least one dissemination channel must be supported by the notification broker service (typically, EMAIL).
<b>GeocodeType</b>	Capabilities	List of supported geocode area types that can be used by the user to indicate what his geographical areas of interest are for receiving a notification (OPTIONAL).	

**Table 2:** Capabilities resource description

### 2.1.2 Consumer resource

Consumer resource object and its sub-elements are described in the following table:

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
<b>Consumer</b>		The container for all component parts of the Consumer object that define the general aspects of the notification recipient (REQUIRED)	
<b>Id</b>	Consumer	The identifier of this consumer object (REQUIRED)	The identifier is UNIQUE and it is automatically assigned by the system
<b>Username</b>	Consumer	The username identifier of this consumer object used to facilitate the retrieval of consumer info (consumer object as well as related subscriptions and notifications) in contraposition to the "id" field. (REQUIRED)	The username is UNIQUE across the whole service (normally, the email address is used). Typically the main email address is used as username.
<b>Category</b>	Consumer	Consumer type of the notifications (REQUIRED)	E.g., Farmer, Farmer Pro, etc.
<b>CountryCode</b>	Consumer	3-letters identifier for a country according to ISO 3166-1 alpha-3 (REQUIRED)	E.g., ESP (for Spain)
<b>LanguageCode</b>	Consumer n	3-letters identifier for the language according to ISO 639-3 used for determining in which language the final notification message must be sent to the recipient (REQUIRED)	E.g., SPA (for Spain)
<b>TimeZone</b>	Consumer	String name of the time zone where the notification recipient is located. If not provided, default is UTC (OPTIONAL)	E.g., Europe/Madrid
<b>Addresses</b>	Consumer	List of addresses where the notification message must be sent to. (REQUIRED)	One or more addresses are allowed. The subscriber will receive the notification through ALL specified addresses. Currently, mobile phone numbers and email addresses are supported for the reception of SMS messages At least one email address MUST be specified.

**Table 3:** Consumer resource description

### 2.1.3 Subscription resource

Subscription resource object and its sub-elements are described in the following table:

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
<b>Subscription</b>		The container for all component parts of the Subscription object that define a user subscription to a topic (REQUIRED)	
<b>Id</b>	Subscription	The identifier of this subscription object (REQUIRED)	The identifier is UNIQUE and it is automatically assigned by the system
<b>Topic</b>	Subscription	Topic name for which the user has subscribed in order to receive related notifications. (REQUIRED)	See the definition of the Topic element in the Capabilities resource section. Please, also note that only ONE subscription to a concrete topic

			is allowed by user account
<b>Enabled</b>	Subscription	Indicates whether this concrete Subscription is enabled. If not, it will be skipped when checking what subscriptions should be used for sending the notifications. (REQUIRED)	Possible values: true or false. New subscriptions are enabled by default.
<b>Restrictions</b>	Subscription	Set of filtering options used to indicate that the user is interesting in receiving the notification message only if ALL conditions are matched. (OPTIONAL)	Restrictions that can be applied to filter the notifications are:
<b>Expression</b>	Subscription Restriction	Filter expression combining one or more topic specific parameters (OPTIONAL)	Only ONE expression is allowed by topic subscription. - Expressions: based on the topic parameters (see TopicDescriptor explanation). Supported expressions are: Comparisons (e.g., parameter > 5), Logical (e.g., parameter1 > 5 && parameter2 < 79; )
<b>Area</b>	Subscription Restriction	Geographic area for which the user is interested in (OPTIONAL)	One or more areas are allowed. The subscriber will only receive the notification if the message contains ANY of the areas here defined. Areas can be defined according to their either a Geocode and/or a valid geometry type.
<b>Geocode</b>	Subscription Restriction Area	Geocode area (OPTIONAL)	Zero or more geocode area identifiers can be used per Area object. The Geocode object is composed of a GeocodeType property (it must be one of the geocode types defined in the Capabilities resource) and the area code (e.g., GeocodeType = HASC1 and value = ES.CV)
<b>Geometry</b>	Subscription Restriction Area	Polygon delimiting the area of interest (OPTIONAL)	Only ONE geometry is allowed in order to define the area of interest. The geometry must be declared according to GeoJson standard [3]. Allowed geometry types are (polygon, multipolygon, point, multipoint, linestring, multilinestring, geometrycollection)
<b>TopicFeatures</b>	Subscription	List of selected topic features and their values (OPTIONAL)	Zero or more TopicFeature objects are allowed (they must be defined in the topic descriptor and cannot be repeated). (E.g., name = includeReport, value = false)

**Table 4:** Subscription resource description

#### 2.1.4 Notification resource

The notification broker distinguishes between two types of notifications. On one hand, the notification messages sent by the producers (NotificationProducer message) which contains the information (i.e., topic name, topic parameters, attached resources such as maps, documents, etc., and additional payload data), to be notified to the subscribers. On the other hand, the notifications that are sent to the subscribers/recipients (ConsumerNotification), including most of the original information sent by the producer plus tailored information according to the subscription options (i.e., text messages in the recipient language, automatically generated reports based on the topic parameters, etc.). It must be noticed that most of the time, the notification recipients will not receive the consumer notification object itself but a channel specific message such as an email or SMS message (ConsumerNotification resources are converted into channel specific formats, and in some cases the ConsumerNotification JSON [2] representation can be attached to it, as it is the case of the email).

As mentioned before, Producer and Consumer notifications have in common several attributes, which are described in the following table:

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
<b>Notification</b>		The common component parts of the Producer and Consumer Notification objects (REQUIRED)	
<b>Id</b>	Notification	The identifier of this (Consumer or Producer) notification object (REQUIRED)	The identifier is UNIQUE and it is automatically assigned by the system
<b>Topic</b>	Notification	Topic name to which this notification is related to (REQUIRED)	See the definition of the Topic element in the Capabilities resource section. Please, also note that a notification can only refer to ONE topic.
<b>MessageType</b>	Notification	A specialization of the type of message sent for a concrete topic (sort of subtopic) (REQUIRED)	E.g., A user may subscribe for topic "meteorological-alerts", however message types can be about "Storm Alert", "Heavy rain Alert", "Snow Alert", etc.
<b>ProducerId</b>	Notification	Identifier of the producer who is "producing" the notification (REQUIRED).	e.g., producer001
<b>ReferenceIds</b>	Notification	List of identifiers of previous notifications related to the current one.(OPTIONAL)	
<b>CommonPayload</b>	Notification	Dictionary of additional data included in the notification (OPTIONAL). This object attribute is usually used to add information to the notification that is common to all InfoElements so it has not to be repeated for each of them. (OPTIONAL)	In the form of "attribute-name": "value". E.g., "textMessage": "this is an example message";
<b>InfoElements</b>	Notification	Set of individual pieces of information normally referred to specific geographical areas (although not necessarily) (OPTIONAL)	Zero or more occurrences are allowed
<b>Parameters</b>	Notification InfoElement	Set of parameter values for this info element. Normally, these parameter values relate to the possible parameters defined in the topic descriptor (OPTIONAL)	Zero or more occurrences of parameters are allowed
<b>Resources</b>	Notification InfoElement	Set of resources (e.g., maps, documents, etc.) attached to the notification (OPTIONAL)	Zero or more occurrences of resources are allowed
<b>Name</b>	Notification InfoElement Resource	Name of the resource (REQUIRED)	
<b>Url</b>	Notification InfoElement Resource	URL of the resource (in case it is provided as an online resource)(CONDITIONAL)	Note: either a URL or actual content data must be provided in order to be a valid resource. It cannot be empty.
<b>MimeType</b>	Notification InfoElement Resource	Mime type of the resource (REQUIRED)	E.g., image/jpg, text/plain; etc.
<b>Content</b>	Notification InfoElement Resource	Actual content of the resource (in case it has been provided as a dereferenced URL) (CONDITIONAL)	Binary information is encoded as Base 64. Note: either a URL or actual content data must be provided in or-

			der to be a valid resource. It cannot be empty.
<b>Areas</b>	Notification InfoElement	Geographic area to which this info element applied (OPTIONAL)	One or more areas are allowed. See explanation in Subscription resource for further details
<b>Payload</b>	Notification InfoElement	Payload data specific for this info element (OPTIONAL)	

**Table 5:** Common notification resource attributes

**2.1.4.1 Producer notification**

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
ProducerNotification		The component parts of the Producer Notification object (REQUIRED)	Producer notification objects have exactly the same attributes as described in Table 5: Common notification resource attributes.

**Table 6:** Producer notification resource attributes

**2.1.4.2 Consumer notification**

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
ConsumerNotification		The component parts of the Consumer Notification object (REQUIRED)	
address	ConsumerNotification	Address to which the notification was sent (REQUIRED)	Only one endpoint address per consumer notification is allowed. In case the user indicated more addresses (of one or more channel types) the system generates one ConsumerNotification object per address endpoint. E.g., <a href="mailto:email-address@email.com">email-address@email.com</a> ; +3468789098
languageCode	ConsumerNotification	Main language of the contents contained in the consumer notification (REQUIRED)	Only one language code is allowed per consumer notification object. Normally it refers to the language of the textual information within the message. Note: ProducerNotification objects (in principle) should contain language agnostic information (such as parameter values, geographic geocode areas or polygons, etc.). The notification broker then composes tailored messages that may contain textual information in the language that

			the user indicated when he was registered in the system.
TopicFeatures	ConsumerNotification	The topic features that were selected by the user when subscribed for the notification topic and that are used by the service to further customize the notification (e.g., include some auto-generated reports, etc.) (OPTIONAL)	
Status	ConsumerNotification	Current status of the delivery of the consumer notification (REQUIRED)	Allowed values are:
MessageSent	ConsumerNotification	Pointer to the final notification message that has been generated, sent to the user and stored in the system. See description of Resource object for further details. (REQUIRED)	The purpose is to be able to recover past notifications in case it is necessary.

**Table 7:** Consumer notification resource attributes

## 2.2 Service operations

The Notification Broker exposes the aforementioned resources by means of RESTful operations (GET, POST, PUT and DELETE), enabling thus consulting the service capabilities; creating, retrieving, updating and deleting user's subscriptions; and pushing ("notifying") notifications to the notification broker for it to customize and deliver them to the final recipients, as well as accessing to the pushed and generated notifications.

In order to better manage the different operations and the authorisation and access over the aforementioned resources, the interfaces and operations have been organized in 4 categories according to the following roles in the service:

- **capabilities-api:** operations requesting the different features ("capabilities") supported by the service which don't require any special permission for accessing to them.
- **consumer-api:** operations that a notification consumer can perform such as for instance creating/deleting/updating a consumer resource in the service, creating/deleting/updating subscriptions for that consumer and accessing (read only) to the notifications that were sent to that consumer.
- **producer-api:** operations that a notification producer can perform such as sending a producer notification to the service for its customization and distribution to the different consumers and accessing (read only) to the producer notifications that the producer sent to the service.
- **admin-api:** typically related to maintenance operations (performed by the service administrator) such as mass creation, delete or update of the resources exposed (e.g., massively creation of consumers, subscriptions, etc.).

### 2.2.1 Capabilities API

The following table summarizes the supported (and planned) service operations that can be performed by any user over these Capabilities resource.

URL path	POST (Create)	GET (Read)	PUT (Update)	DELETE (Delete)
/capabilities	N/A	Lists all supported features by the notification service	N/A	N/A
/capabilities/languageCode	N/A	Lists all supported language codes that can be used to disseminate a notification	N/A	N/A
/capabilities/geocodeTypes	N/A	Lists all supported geocode types that can be used in order to indicate areas of interest	N/A	N/A
/capabilities/channelTypes	N/A	Lists all supported channel types that can be used for disseminating a notification	N/A	N/A

/capabilities/topicDescriptors	N/A	Lists all supported topics to which a user can be subscribed.	N/A	N/A
--------------------------------	-----	---	-----	-----

**Table 8:** Capabilities resource operations

## 2.2.2 Consumer API

The following table summarizes the supported (and planned) service operations that can be performed by a consumer over the Consumer, and its related Subscription and ConsumerNotification resources.

URL path	POST (Create)	GET (Read)	PUT (Update)	DELETE (Delete)
/consumers	Creates a new consumer	N/A	N/A	N/A
/consumers/{consumerId}	N/A	Returns a consumer by its associated consumerId	PLANNED	Deletes a consumer by its associated consumerId
/consumers/{consumerId}/languageCode Note: for the PUT operation use /consumers/{consumerId}/languageCode={languageCode}	N/A	Returns a consumer iso language code by its associated consumerId	Updates a consumer iso language code by its associated consumerId.	N/A
/consumers/{consumerId}/addresses	N/A	Returns a consumer's set of notification addresses by its associated consumerId	Updates a consumer list of contact addresses by its associated consumerId	Removes a consumer list of contact addresses by its associated consumerId
/consumers/{consumerId}/subscriptions?QUERY_STRING_PARAMETERS, which can be one or more from: topics=[list of topic names] & enabled=(true false)	Creates a consumer subscription by its associated consumerId	Lists all consumer subscriptions by its associated consumerId and possibly additional query string parameters	N/A	Removes all consumer subscriptions by its associated consumerId and possibly additional query string parameters
/consumers/{consumerId}/subscriptions/{subscriptionId}	N/A	Lists a consumer's subscription by its associated consumerId and subscriptionId	Validates a consumer's subscription by its associated consumerId and subscriptionId	Removes a consumer's subscription by its associated consumerId and subscriptionId
/consumers/{consumerId}/subscriptions/{subscriptionId}/enabled	N/A	Gets consumer's subscription enabled status by its associated consumerId and subscriptionId	Updates a consumer notification subscription status according to its consumerId and subscriptionId	N/A
/subscriptions & channelType=(one of the channel types) & addresses=[list of addresses]	N/A	Lists all users' topic subscriptions matching the query string parameter values	N/A	Deletes all users' topic subscriptions matching the query string parameter values
/consumers/{consumerId}/notifications?QUERY_STRING_PARAMETERS, which can be one or more from: topic=topicName & status=[UNKNOWN, SENT, DELIVERED, ERROR] & channelType=(one of the channel types) & addresses=[list of addresses] &	N/A	Lists all consumer's notification by its associated consumerId and possibly additional query string parameters	N/A	N/A

producerIds=[list of producer ids] & referenceIds=[list of reference ids]				
Note1: probably to add more: spatial search				
/consumers/{consumerId}/notifications/{consumerNotificationId}	N/A		Gets a consumer notification by its associated consumerId and consumer consumerNotificationId	N/A

**Table 9: Consumer API operations**

### 2.2.3 Producer API

The following table summarizes the supported (and planned) service operations that can be performed by a producer over the Producer and related ProducerNotification resources.

URL path	POST (Create)	GET (Read)	PUT (Update)	DELETE (Delete)
/producers/{producerId}/notifications?QUERY_STRING which can be one or more of: topics=[list of topic names] & referenceIds=[list of reference ids]	N/A	List all notifications (ProducerNotification) pushed to the service by the notification producer that match with ALL the specified parameters in the QUERY_STRING. If no query string is specified all producer notifications for that producerid are returned.	N/A	N/A
/producers/{producerId}/notifications/{producerNotificationId}	N/A	List the notification (ProducerNotification) pushed to the service by the notification producer	N/A	N/A
/producers/{producerId}/notify	N/A	Pushes a NotifyRequest to the notification service. The request contains a ProducerNotification message, which can be notified to the users registered in the system and/or to the complementary list of addresses specified in the request. If “includeSubscribers” == true then registered users will also be notified.	N/A	N/A

**Table 10: Producer API operations**

### 2.2.4 Admin API

The following table summarizes the supported service operations that can be performed by an administrator over ALL the resources managed in the service.

URL path	POST (Create)	GET (Read)	PUT (Update)	DELETE (Delete)
/admin/consumers	N/A	Lists all consumer in the database	N/A	Delete all consumers from the database
/admin/consumers/subscriptions	N/A	Lists all subscriptions available for consumer in the database	N/A	Deletes all consumer’s subscriptions available in the database.

**Table 11: Admin API operations**

### 3 Service feed extension

For the next period, we plan to provide an extension for the notification broker service in order to generate Web feeds to which users can subscribe and receive notifications from any compatible reader (e.g., browser, email client, blogs, etc.)

The idea is that the notifications generated by the producers will also become available as a feeds.

This extension will be a simple component on top of the existing service, which will connect either to the database of notifications, or subscribe through the available channels, and automatically transform the notification into a feed that will be available through the service extension endpoint. The component will be based and adapted on the Notification API<sup>1</sup> and its implementation<sup>2</sup> from the Wf4Ever project.

#### 3.1 Service resources and links

##### 3.1.1 Service description

The service description is an RDF file that contains the URI template for the notification resource. The RDF syntax used may be content negotiated. In the absence of content negotiation, RDF/XML should be returned. The client should create the notification URI by building the template with the parameters described in the previous sections.

##### 3.1.2 Notification resource

The notifications will be generated as Atom feeds, and they will have the following attributes:

- Title: name of the notification type
- Subject: topic of the notification
- Source: URI of the producer of the notification
- Id: an internal entry id, consistent across different feeds
- Published: time and date of the notification
- Content: notification message

##### 3.1.2.1 Links

- <http://purl.org/dc/terms/source>: to specify URI of the producer of the notification
- <http://purl.org/dc/terms/subject>: to specify the topic of the notification
- [http://rdfs.org/sioc/spec/#term\\_id](http://rdfs.org/sioc/spec/#term_id): to specify the user Id.

##### 3.1.3 Service operations

GET operation is used to get the notifications. Notifications are read only, and no other method is available.

The client should create the notification URI using the following template:

/atomservice?QUERY\_STRING\_PARAMETERS, which can be the following:

- userid – unique identification of the subscriber (mandatory)
- topic - name of the topic of the notifications to receive
- from - the timestamp of the oldest notification that should be returned.
- to - the timestamp of the most recent notification that should be returned.
- source - URI of the producer of notifications.
- limit - the maximum number of results to be returned.

If neither topic nor source are specified, notifications from all topics to which user is subscribed should be re-

<sup>1</sup> <https://github.com/wf4ever/apis/wiki/Notification-API>

<sup>2</sup> <http://sandbox.wf4ever-project.org/rodl/atomnotifications>

turned. If neither "from", "to" nor "limit" are set, the service should return no more than 10 notifications.

## 4 Conclusions

This deliverable has presented the first results achieved concerning the task T4.4 Reporting and alerts, focusing in first place on the adaptation and enhancement of an existing notification service - developed within the context of FP6 DEWS project – that allows to the FOODIE platform users to subscribe and receive agriculture based notifications.

In this sense, the work presented in this deliverable shows the main logical resources managed and generated by the notification broker (i.e., Subscriptions and Notifications resources) as well as the main REST operations that allow interacting with the service. Besides, in its current state, the notification broker is already able to generate notifications with dummy data and disseminate it via EMAIL and SMS channels.

The service includes Swagger Framework libraries [4] which allows it to auto-document its REST interfaces, facilitating thus to future programmers and companies to easily understand the interfaces and data structures of the service.

In this second version of the deliverable, efforts have been focused on redefining and refactoring several REST operations of the service, separating them into four categories: capabilities, consumer, producer and administration related operations.

In the next months, the work to be carried out contemplates the improvement of existing features and enhancement of existing ones such as:

- Adoption of security mechanisms provided by FOODIE platform in order to prevent unauthorised use of the notification broker API by external users
- Support of additional notification communication channels such as RSS feed and Google Cloud Messaging, being the later of special relevance in order to allow the system to send directly the notifications to the mobile application developed by CTIC.
- Possibility to use the parcel identifier as geocode area, enabling in an easy manner to the farmers to receive notifications directly affecting their farms.
- Implementation of additional API interfaces, such as the registration of notification producers. This would allow for instance that external systems authorised by FOODIE may publish their own topics to which FOODIE users can subscribe in order to receive notifications from them as well. This may include wide variety of applications, ranging from advanced recommendation systems to announcement of new products and services that the users may be willing to acquire or pay for.
- Linking with other FOODIE platform components such as the recommenders and report generators defined in deliverable D4.3.1. Advanced Decision Support Tools.

## References

- [1] FP6 DEWS (Distant Early Warning System) project. Link: <http://www.dews-online.org>
- [2] JSON standard. Link: <http://www.json.org>
- [3] GEOJSON standard. Link: <http://geojson.org>
- [4] SWAGGER Framework. Link: <http://swagger.io>

## Annex A – API documentation with Swagger

➔ swagger
full-notification-broker-api (v2/api-docs?group=full-notification-broker-api)

Explore

**Notification Broker API**

Notification Broker supported operations

Created by Miguel Angel Esbri <miguel.esbri@atos.net>  
[Apache License Version 2.0](#)

**administration-controller : Administration Controller** Show/Hide | List Operations | Expand Operations

DELETE	/admin/consumers	removeAllConsumers
GET	/admin/consumers	getAllConsumers
DELETE	/admin/consumers/subscriptions	removeAllSubscriptions
GET	/admin/consumers/subscriptions	getAllSubscriptions

**capabilities-controller : Capabilities Controller** Show/Hide | List Operations | Expand Operations

GET	/capabilities	getCapabilities
GET	/capabilities/channelTypes	getChannelTypeDescriptors
GET	/capabilities/geocodeTypes	getGeocodeTypeDescriptors
GET	/capabilities/languageCodes	getLanguageCodeDescriptors
GET	/capabilities/topicDescriptors	getTopicDescriptors

**consumer-controller : Consumer Controller** Show/Hide | List Operations | Expand Operations

POST	/consumers	createConsumer
DELETE	/consumers/{consumerId}	removeConsumer
GET	/consumers/{consumerId}	getConsumer
DELETE	/consumers/{consumerId}/addresses	removeConsumerAddresses
GET	/consumers/{consumerId}/addresses	getConsumerAddresses
PUT	/consumers/{consumerId}/addresses	updateConsumerAddresses
GET	/consumers/{consumerId}/languageCode	getConsumerLanguageCode
PUT	/consumers/{consumerId}/languageCode={languageCode}	updateConsumerLanguageCode
GET	/consumers/{consumerId}/notifications	getConsumerNotifications
GET	/consumers/{consumerId}/notifications/{consumerNotificationId}	getConsumerNotification
DELETE	/consumers/{consumerId}/subscriptions	removeConsumerSubscriptions
GET	/consumers/{consumerId}/subscriptions	getConsumerSubscriptions
POST	/consumers/{consumerId}/subscriptions	createConsumerSubscription
DELETE	/consumers/{consumerId}/subscriptions/{subscriptionId}	removeConsumerSubscription
GET	/consumers/{consumerId}/subscriptions/{subscriptionId}	getConsumerSubscription
PUT	/consumers/{consumerId}/subscriptions/{subscriptionId}	updateConsumerSubscription
GET	/consumers/{consumerId}/subscriptions/{subscriptionId}/enabled	getConsumerSubscriptionEnabledStatus
PUT	/consumers/{consumerId}/subscriptions/{subscriptionId}/enabled={enabledStatus}	updateConsumerSubscriptionEnabledStatus

**producer-controller : Producer Controller** Show/Hide | List Operations | Expand Operations

POST	/producers/notify	notify
GET	/producers/{producerId}/notifications	getProducerNotifications
GET	/producers/{producerId}/notifications/{producerNotificationId}	getProducerNotification

[ BASE URL: / , API VERSION: 2.0 ]

## Annex B – REST operations examples

### B.1 Capabilities

GET /notification-broker/v1.0/capabilities

Response:

```
{
  "isoLanguageCodeDescriptors": [ {
    "iso639_1_code": "EN",
    "iso639_2_code": "ENG",
    "longName": "English"
  }, {
    "iso639_1_code": "CS",
    "iso639_2_code": "CES",
    "longName": "Czech"
  }, {
    "iso639_1_code": "ES",
    "iso639_2_code": "SPA",
    "longName": "Spanish"
  } ],
  "geocodeTypeDescriptors": [ {
    "type": "ISO2",
    "description": "ISO 3166-1 alpha-2 codes are two-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest.",
    "reference": "https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2"
  }, {
    "type": "HASC2",
    "description": "Hierarchical administrative subdivision codes (HASC) are codes to represent names of country subdivisions, such as states, province, regions.",
    "reference": "http://www.statoids.com/ihasc.html"
  }, {
    "type": "ISO3",
    "description": "ISO 3166-1 alpha-3 codes are three-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest.",
    "reference": "https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3"
  }, {
    "type": "HASC1",
    "description": "Hierarchical administrative subdivision codes (HASC) are codes to represent names of country subdivisions, such as states, province, regions.",
    "reference": "http://www.statoids.com/ihasc.html"
  } ],
  "channelTypeDescriptors": [ {
    "type": "SMS",
    "pattern": "^\\+(?:[0-9]\\x20?){6,14}[0-9]$"
  }, {
    "type": "EMAIL",
    "pattern": "[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]$"
  }, {
    "type": "FAX",
    "pattern": "^\\+(?:[0-9]\\x20?){6,14}[0-9]$"
  } ],
  "topicDescriptors": [ {
    "name": "http://www.foodie-project.eu/topic/meteorological-alert",
    "description": "meteorological related alerts",
    "parameterDescriptors": [ {
      "name": "windSpeed",
      "type": "java.lang.Double",
      "required": false,
      "description": "wind-speed"
    }, {
      "name": "windDirection",
      "type": "java.lang.Double",
      "required": false,
      "description": "wind-direction"
    } ]
  } ]
}
```

```

    }}
  }, {
    "name": "http://www.foodie-project.eu/topic/crop-plague-alert",
    "description": "alerts concerning plagues and diseases affecting the crops",
    "parameterDescriptors": [ {
      "name": "plagueType",
      "type": "java.lang.String",
      "required": false,
      "description": "plague type affecting the crop"
    } ], {
      "name": "affectedCropPercentage",
      "type": "java.lang.Float",
      "required": false,
      "description": "percentage of the crop affected by the plague/disease"
    } ], {
      "name": "cropType",
      "type": "java.lang.String",
      "required": false,
      "description": "crop type affected by the plague/disease"
    } ]
  }, {
    "name": "http://www.driver.eu/topic/flood-alert",
    "description": "flood related alerts",
    "parameterDescriptors": [ {
      "name": "windSpeed",
      "type": "java.lang.Double",
      "required": false,
      "description": "wind-speed"
    } ], {
      "name": "windDirection",
      "type": "java.lang.Double",
      "required": false,
      "description": "wind-direction"
    } ] },
    "featureDescriptors": [ {
      "name": "messageType",
      "type": "java.lang.String",
      "required": false
    } ], {
      "name": "includeCap",
      "type": "java.lang.Boolean",
      "required": false
    } ]
  }, {
    "name": "http://www.driver.eu/topic/storm-alert",
    "description": "meteorological related alerts",
    "parameterDescriptors": [ {
      "name": "windDirection",
      "type": "java.lang.Double",
      "required": false,
      "description": "wind-direction"
    } ], {
      "name": "windSpeed",
      "type": "java.lang.Double",
      "required": false,
      "description": "wind-speed"
    } ]
  } ]
}

```

**B.1.2. Channel Types**

GET /notification-broker/v1.0/capabilities/channelTypes

Response:

```

[ {
  "type": "SMS",

```

```
"pattern" : "^\\+(?:[0-9]\\x20?){6,14}[0-9]$"
}, {
  "type" : "EMAIL",
  "pattern" : "[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+"
}, {
  "type" : "FAX",
  "pattern" : "^\\+(?:[0-9]\\x20?){6,14}[0-9]$"
}]
```

### B.1.3 ISO Language Codes

GET /notification-broker/v1.0/capabilities/isoLanguageCodes

Response:

```
[ {
  "iso639_1_code" : "EN",
  "iso639_2_code" : "ENG",
  "longName" : "English"
}, {
  "iso639_1_code" : "CS",
  "iso639_2_code" : "CES",
  "longName" : "Czech"
}, {
  "iso639_1_code" : "ES",
  "iso639_2_code" : "SPA",
  "longName" : "Spanish"
}]
```

### B.1.4 Geocode Types

GET /notification-broker/v1.0/capabilities/geocodeTypes

Response:

```
[ {
  "type" : "ISO2",
  "description" : "ISO 3166-1 alpha-2 codes are two-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest.",
  "reference" : "https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2"
}, {
  "type" : "HASC2",
  "description" : "Hierarchical administrative subdivision codes (HASC) are codes to represent names of country subdivisions, such as states, province, regions.",
  "reference" : "http://www.statoids.com/ihasc.html"
}, {
  "type" : "ISO3",
  "description" : "ISO 3166-1 alpha-3 codes are three-letter country codes defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest.",
  "reference" : "https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3"
}, {
  "type" : "HASC1",
  "description" : "Hierarchical administrative subdivision codes (HASC) are codes to represent names of country subdivisions, such as states, province, regions.",
  "reference" : "http://www.statoids.com/ihasc.html"
}]
```

### B.1.5 Topic Descriptors

GET /notification-broker/v1.0/capabilities/topicDescriptors

Response:

```
[ {
  "name" : "http://www.foodie-project.eu/topic/meteorological-alert",
  "description" : "meteorological related alerts",
  "parameterDescriptors" : [ {
    "name" : "windSpeed",
    "type" : "java.lang.Double",
```

```

    "required" : false,
    "description" : "wind-speed"
  }, {
    "name" : "windDirection",
    "type" : "java.lang.Double",
    "required" : false,
    "description" : "wind-direction"
  }
], {
  "name" : "http://www.foodie-project.eu/topic/crop-plague-alert",
  "description" : "alerts concerning plagues and diseases affecting the crops",
  "parameterDescriptors" : [ {
    "name" : "plagueType",
    "type" : "java.lang.String",
    "required" : false,
    "description" : "plague type affecting the crop"
  }, {
    "name" : "affectedCropPercentage",
    "type" : "java.lang.Float",
    "required" : false,
    "description" : "percentage of the crop affected by the plague/disease"
  }, {
    "name" : "cropType",
    "type" : "java.lang.String",
    "required" : false,
    "description" : "crop type affected by the plague/disease"
  }
], {
  "name" : "http://www.driver.eu/topic/flood-alert",
  "description" : "flood related alerts",
  "parameterDescriptors" : [ {
    "name" : "windSpeed",
    "type" : "java.lang.Double",
    "required" : false,
    "description" : "wind-speed"
  }, {
    "name" : "windDirection",
    "type" : "java.lang.Double",
    "required" : false,
    "description" : "wind-direction"
  }
],
  "featureDescriptors" : [ {
    "name" : "messageType",
    "type" : "java.lang.String",
    "required" : false
  }, {
    "name" : "includeCap",
    "type" : "java.lang.Boolean",
    "required" : false
  }
], {
  "name" : "http://www.driver.eu/topic/storm-alert",
  "description" : "meteorological related alerts",
  "parameterDescriptors" : [ {
    "name" : "windDirection",
    "type" : "java.lang.Double",
    "required" : false,
    "description" : "wind-direction"
  }, {
    "name" : "windSpeed",
    "type" : "java.lang.Double",
    "required" : false,
    "description" : "wind-speed"
  }
]
}]

```





```
}]  
}  
}
```

Response: (subscriptionId) 5698e0e502c7da382bb4a005

#### **B.2.4 Update subscription “enabled” status**

PUT /notification-broker/v1.0/subscriptions/{subscriptionId}/enabled=false

#### **B.2.5 Update subscription dissemination channels**

PUT /notification-broker/v1.0/subscriptions/{subscriptionId}/channelTypes

RequestBody: [EMAIL, SMS]

### B.3 Notifications

Notification operations are split into producer and consumer notification operations

#### B.3.1 Producer Notifications

##### B.3.1.1 Get all producer notifications

GET /notification-broker/v1.0/notifications/producer

Response:

```
[{
  "topic": "http://www.driver.eu/topic/flood-alert",
  "producerId": "producer:002",
  "referenceId": "0",
  "id": "5698e67202c7da34394519b1",
  "createdAt": "2016-01-05T15:39:36+0000",
  "infoElements": [{
    "parameters": [{
      "name": "windSpeed",
      "value": 20.0
    }],
    "resources": [{
      "filename": "Birch_bark_document_210.jpg",
      "url": "https://upload.wikimedia.org/wikipedia/commons/2/2a/Birch_bark_document_210.jpg",
      "mediaType": "image/jpeg"
    }, {
      "filename": "test.pdf",
      "url": "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType": "application/pdf"
    }],
    "areas": [{
      "geometry": {
        "type": "Polygon",
        "coordinates": [[[ 35.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 35.0, 10.0 ]], [ [ 20.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 20.0, 30.0 ]]]
      }
    }
  ]
}]
}
```

##### B.3.1.2 Get a subset of producer notifications based on certain parameters

GET /notification-broker/v1.0/notifications/producer?producerId=producer:002

Response:

```
[{
  "topic": "http://www.driver.eu/topic/flood-alert",
  "producerId": "producer:002",
  "referenceId": "0",
  "id": "5698e67202c7da34394519b1",
  "createdAt": "2016-01-05T15:39:36+0000",
  "infoElements": [{
    "parameters": [{
      "name": "windSpeed",
```





```

]]
  }
}, {
  "geocode" : {
    "type" : "ISO2",
    "value" : "NL"
  },
  "geometry" : {
    "type" : "Polygon",
    "coordinates" : [[ [ 45.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 45.0, 10.0 ] ], [ [ 40.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 40.0, 30.0 ]
]]
  }
}]
}, {
  "parameters" : [ {
    "name" : "windSpeed",
    "value" : 4.0
  } ],
  "resources" : [ {
    "filename" : "test.pdf",
    "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
    "mediaType" : "application/pdf"
  } ],
  "endpointAddress" : {
    "type" : "SMS",
    "value" : "+34691091744"
  },
  "isoLanguageCode" : "EN",
  "topicFeatures" : [ {
    "name" : "includeCap",
    "value" : true
  } ],
  "status" : "UNKNOWN_STATUS"
}, {
  "id" : "5698e67302c7da34394519b5",
  "createdAt" : "2016-01-15T12:30:43+0000",
  "topic" : "http://www.driver.eu/topic/flood-alert",
  "producerId" : "producer:002",
  "referenceId" : "0",
  "infoElements" : [ {
    "parameters" : [ {
      "name" : "windSpeed",
      "value" : 20.0
    } ],
    "resources" : [ {
      "filename" : "Birch_bark_document_210.jpg",
      "url" : "https://upload.wikimedia.org/wikipedia/commons/2/2a/Birch_bark_document_210.jpg",
      "mediaType" : "image/jpeg"
    } ],
    {
      "filename" : "test.pdf",
      "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType" : "application/pdf"
    } ],
    "areas" : [ {
      "geocode" : {
        "type" : "HASC1",
        "value" : "NL.AS"
      },
      "geometry" : {
        "type" : "Polygon",
        "coordinates" : [[ [ 35.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 35.0, 10.0 ] ], [ [ 20.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 20.0, 30.0 ]
]]
      }
    } ],
    {
      "geocode" : {
        "type" : "ISO2",
        "value" : "NL"
      }
    }
  }
}

```

```

    },
    "geometry" : {
      "type" : "Polygon",
      "coordinates" : [[ [ 45.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 45.0, 10.0 ] ], [ [ 40.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 40.0, 30.0 ] ] ] ] ]
  }
}
}, {
  "parameters" : [ {
    "name" : "windSpeed",
    "value" : 4.0
  } ],
  "resources" : [ {
    "filename" : "test.pdf",
    "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
    "mediaType" : "application/pdf"
  } ]
}, {
  "endpointAddress" : {
    "type" : "EMAIL",
    "value" : "maesbri@gmail.com"
  },
  "isoLanguageCode" : "EN",
  "topicFeatures" : [ {
    "name" : "includeCap",
    "value" : true
  } ],
  "status" : "UNKNOWN_STATUS"
}, {
  "id" : "5698e67302c7da34394519b4",
  "createdAt" : "2016-01-15T12:30:43+0000",
  "topic" : "http://www.driver.eu/topic/flood-alert",
  "producerId" : "producer:002",
  "referenceId" : "0",
  "infoElements" : [ {
    "parameters" : [ {
      "name" : "windSpeed",
      "value" : 20.0
    } ],
    "resources" : [ {
      "filename" : "Birch_bark_document_210.jpg",
      "url" : "https://upload.wikimedia.org/wikipedia/commons/2/2a/Birch_bark_document_210.jpg",
      "mediaType" : "image/jpeg"
    } ],
    {
      "filename" : "test.pdf",
      "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType" : "application/pdf"
    } ],
    "areas" : [ {
      "geocode" : {
        "type" : "HASC1",
        "value" : "NL.AS"
      },
      "geometry" : {
        "type" : "Polygon",
        "coordinates" : [[ [ 35.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 35.0, 10.0 ] ], [ [ 20.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 20.0, 30.0 ] ] ] ] ]
    } ] ] ] ] ]
}
}, {
  "geocode" : {
    "type" : "ISO2",
    "value" : "NL"
  },
  "geometry" : {
    "type" : "Polygon",
    "coordinates" : [[ [ 45.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 45.0, 10.0 ] ], [ [ 40.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 40.0, 30.0 ] ] ] ] ]
}
}
}

```



```

    "resources" : [ {
      "filename" : "test.pdf",
      "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType" : "application/pdf"
    } ]
  },
  "endpointAddress" : {
    "type" : "EMAIL",
    "value" : "user2.2@iriss-platform.com"
  },
  "isoLanguageCode" : "EN",
  "topicFeatures" : [ {
    "name" : "includeCap",
    "value" : true
  } ],
  "status" : "UNKNOWN_STATUS"
}, {
  "id" : "5698e67302c7da34394519b2",
  "createdAt" : "2016-01-15T12:30:43+0000",
  "topic" : "http://www.driver.eu/topic/flood-alert",
  "producerId" : "producer:002",
  "referenceId" : "0",
  "infoElements" : [ {
    "parameters" : [ {
      "name" : "windSpeed",
      "value" : 20.0
    } ],
    "resources" : [ {
      "filename" : "Birch_bark_document_210.jpg",
      "url" : "https://upload.wikimedia.org/wikipedia/commons/2/2a/Birch_bark_document_210.jpg",
      "mediaType" : "image/jpeg"
    } ],
    {
      "filename" : "test.pdf",
      "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType" : "application/pdf"
    }
  ] },
  "areas" : [ {
    "geometry" : {
      "type" : "Polygon",
      "coordinates" : [ [ [ 35.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 35.0, 10.0 ] ], [ [ 20.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 20.0, 30.0 ] ] ] ]
    }
  } ]
} ],
{
  "parameters" : [ {
    "name" : "windSpeed",
    "value" : 4.0
  } ],
  "resources" : [ {
    "filename" : "test.pdf",
    "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
    "mediaType" : "application/pdf"
  } ]
},
"endpointAddress" : {
  "type" : "EMAIL",
  "value" : "user1@iriss-platform.com"
},
"isoLanguageCode" : "ES",
"status" : "UNKNOWN_STATUS"
}]

```

### B.3.2.2 Get a subset of consumer notifications based on certain parameters

GET /notification-broker/v1.0/notifications/consumer?channelType=SMS

Response:

```
[{
  "id" : "5698e67302c7da34394519b6",
  "createdAt" : "2016-01-15T12:30:43+0000",
  "topic" : "http://www.driver.eu/topic/flood-alert",
  "producerId" : "producer:002",
  "referenceId" : "0",
  "infoElements" : [{
    "parameters" : [{
      "name" : "windSpeed",
      "value" : 20.0
    }],
    "resources" : [{
      "filename" : "Birch_bark_document_210.jpg",
      "url" : "https://upload.wikimedia.org/wikipedia/commons/2/2a/Birch_bark_document_210.jpg",
      "mediaType" : "image/jpeg"
    }, {
      "filename" : "test.pdf",
      "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType" : "application/pdf"
    }],
    "areas" : [{
      "geocode" : {
        "type" : "HASC1",
        "value" : "NL.AS"
      },
      "geometry" : {
        "type" : "Polygon",
        "coordinates" : [[ [ 35.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 35.0, 10.0 ] ], [ [ 20.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 20.0, 30.0 ] ] ] ] ]
    }],
    "geocode" : {
      "type" : "ISO2",
      "value" : "NL"
    },
    "geometry" : {
      "type" : "Polygon",
      "coordinates" : [[ [ 45.0, 10.0 ], [ 10.0, 20.0 ], [ 15.0, 40.0 ], [ 45.0, 45.0 ], [ 45.0, 10.0 ] ], [ [ 40.0, 30.0 ], [ 35.0, 35.0 ], [ 30.0, 20.0 ], [ 40.0, 30.0 ] ] ] ] ]
    }],
    "parameters" : [{
      "name" : "windSpeed",
      "value" : 4.0
    }],
    "resources" : [{
      "filename" : "test.pdf",
      "url" : "https://s2.q4cdn.com/024715446/files/doc_downloads/test.pdf",
      "mediaType" : "application/pdf"
    }],
    "endpointAddress" : {
      "type" : "SMS",
      "value" : "+34691091744"
    },
    "isoLanguageCode" : "EN",
    "topicFeatures" : [{
      "name" : "includeCap",
      "value" : true
    }],
    "status" : "UNKNOWN_STATUS"
  }]
```

**B.3.2.3 Delete all consumer notifications**

DELETE /notification-broker/v1.0/notifications/consumer

**B.3.2.4 Delete consumer notifications that match certain parameters**

DELETE /notification-broker/v1.0/notifications/consumer?channelType=EMAIL&status=DELIVERED