Country Code Base Registry

*Oracle Service Bus Interface*

*Version 1.0*

Document History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Comments** |
| 0.7 | 30/04/2013 | First Issue |
| 0.8 | 23/07/2013 | Minor corrections, updated error codes, added notes, latest versions of wsdl |
| 0.9 | 16/09/2013 | Included users groups and access rights |
| 0.10 | 02/05/2014 | Add new web methods to return subsctibed information, add the latest version of the wsdl. |
|  |  |  |

Approval

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Date** | **Signature** |
| Prepared by: | Nuno Barruncho | 30/04/2013 |  |
| Visa by: |  |  |  |
| Approved by: |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Company** | **Name** | **E-Mail Address** |
| EMSA |  |  |
| EMSA |  |  |
| EMSA |  |  |
| EMSA |  |  |
| EMSA |  |  |
| EMSA |  |  |
| EMSA |  |  |

Table of Contents

[1 Objective 5](#_Toc386793370)

[2 CBR / OSB Web Services Interface 5](#_Toc386793371)

[2.1 Subscription Web Service 10](#_Toc386793372)

[2.1.1 SignSubscription 11](#_Toc386793373)

[2.1.2 updateSubscription 12](#_Toc386793374)

[2.1.3 revokeSubscription 13](#_Toc386793375)

[2.1.4 subscribe 14](#_Toc386793376)

[2.1.5 unsubscribe 17](#_Toc386793377)

[2.1.6 getServiceSubscribers 19](#_Toc386793378)

[2.1.7 getSubscriptionOperations 20](#_Toc386793379)

[2.1.8 getServiceSubscribersList 21](#_Toc386793380)

[2.1.9 updateService announcementStatus 22](#_Toc386793381)

[2.2 Manage Countries Web Service 24](#_Toc386793382)

[2.2.1 insertCountry 24](#_Toc386793383)

[2.2.2 updateCountry 25](#_Toc386793384)

[2.2.3 deleteCountry 26](#_Toc386793385)

[2.2.4 insertMID 27](#_Toc386793386)

[2.2.5 updateMID 28](#_Toc386793387)

[2.2.6 deleteMID 29](#_Toc386793388)

[2.2.7 insertCallsign 30](#_Toc386793389)

[2.2.8 updateCallsign 31](#_Toc386793390)

[2.2.9 deleteCallsign 32](#_Toc386793391)

[2.3 Country Information Web Service 33](#_Toc386793392)

[2.3.1 getCountryList 33](#_Toc386793393)

[2.3.2 getCountryFullList 35](#_Toc386793394)

[2.3.3 getSubscribedCountryList 36](#_Toc386793395)

[2.3.4 getCountry 37](#_Toc386793396)

[2.3.5 getParentCountry 39](#_Toc386793397)

[2.3.6 getChildCountryList 40](#_Toc386793398)

[2.3.7 getCountryMIDList 41](#_Toc386793399)

[2.3.8 getCountryCallsignList 42](#_Toc386793400)

[2.3.9 getCountryFlagImageList 43](#_Toc386793401)

[2.3.10 getCountryFlagImages 44](#_Toc386793402)

[2.3.11 getCountryFlagImage 45](#_Toc386793403)

[2.3.12 getCountryFlagImageLocation 46](#_Toc386793404)

[2.3.13 getCountryFlagImageLocationContext 47](#_Toc386793405)

[2.3.14 getCountryHistory 48](#_Toc386793406)

[2.3.15 getSubscribedCountries 49](#_Toc386793407)

[2.3.16 getSubscribedCountryTypes 50](#_Toc386793408)

[2.3.17 getCountryCategories 51](#_Toc386793409)

[2.3.18 getCountryTypes 52](#_Toc386793410)

[2.3.19 getSubscribedRegionalAgreements 53](#_Toc386793411)

[2.4 Announcement Web Service 54](#_Toc386793412)

[2.4.1 notify 54](#_Toc386793413)

[2.5 User Groups and Access Rights 55](#_Toc386793414)

[3 Annex I 57](#_Toc386793415)

[SubscriptionService 57](#_Toc386793416)

[ManageCountriesService 76](#_Toc386793417)

[CountryInformationService 95](#_Toc386793418)

[NotificationService 130](#_Toc386793419)

[4 Annex II 131](#_Toc386793420)

# Objective

The objective of this document is to describe the services made available (through web services) on the Oracle Service Bus platform to be used by the Country Base Registry clients. By using those web services, the clients (the EMSA applications) will be able to get the most up-to-date information about Countries, MID’s and Callsign’s.

# CBR / OSB Web Services Interface

The Oracle Service Bus platform will make available four web services with the following operations to be used by the Country Code Base Registry clients:

|  |  |
| --- | --- |
| **SubscriptionService** | |
| Service Operation | Service Description |
| signSubscription(String appAcronym, String announcementEndpoint) | By calling this WebMethod an external system (Application identified by appAcronym) signs for the subscription service. |
| updateSubscription(String appAcronym, String announcementEndpoint) | By calling this WebMethod an external system (Application identified by appAcronym) updates his web service announcement endpoint. |
| revokeSubscription(String appAcronym) | By calling this WebMethod an external system (Application identified by appAcronym) revokes the subscription service. |
| Subscribe(String appAcronym , String subscriptionType, String subscriptionCode) | By calling this WebMethod an Application subscribes to the announcement service for the:   * Country identified by alpha2code. * Regional Agreement identified by the Regional Agreement Code. * Country Type identified by the country Type Code. |
| Unsubscribe(String appAcronym , String unsubscriptionType, String unsubscriptionCode) | By calling this WebMethod an Application unsubscribes to the announcement service for the:   * Country identified by alpha2code. * Regional Agreement identified by the Regional Agreement Code. * Country Type identified by the country Type Code. |
| getServiceSubscribers() | Returns the list of subscribed external systems. |
| getSubscriptionOperations(String appAcronym, javax.xml.datatype.XMLGregorianCalendar startDate, javax.xml.datatype.XMLGregorianCalendar endDate) | Returns the list of the operations performed by an external system (Application identified by appAcronym) matching the search criteria (between the startDate and endDate including boundaries).  If no dates are provided, this WebMethod should return the full list of operations. |
| getServiceSubscribersList (String alpha2code) | Returns the list of subscribed external systems for a specific country. The service subscribers presented in this list will receive an announcement about an operation (insert, update or delete) performed over a country. |
| updateServiceAnnouncementStatus(String appAcronym, String alpha2code, String announcementType, javax.xml.datatype.XMLGregorianCalendar operationDate) | Since the service announcement is handled by the CCBR OSB Services, this method is used to store in the database information about the announcement sent to the ES. |

|  |  |
| --- | --- |
| **ManageCountriesService** | |
| Service Operation | Service Description |
| insertCountry(Country cntr) | Insert a new country in the central database. |
| updateCountry(Country cntr) | Updates the data of an existent country in the central database. |
| deleteCountry(String alpha2Code) | Deletes an existent country from the central database |
| insertMID(MID mid) | Insert a new MID in the central database |
| updateMID(MID mid) | Updates the data of an existent MID in the central database. |
| deleteMID(String midID) | Deletes an existent MID from the central database |
| insertCallsign(Callsign callsign) | Insert a new Callsign in the central database |
| updateCallsign(Callsign callsign) | Updates the data of an existent Callsign in the central database. |
| deleteCallsign(String callsignIdD) | Deletes an existent Callsign from the central database |

|  |  |
| --- | --- |
| **CountryInformationService** | |
| Service Operation | Service Description |
| getCountryList(CountryDetailedSearchCriteria searchMsg) | Returns the list of countries matching the search criteria and respective details. |
| getCountryFullList() | Returns the full list of countries and respective details. |
| getSubscribedCountryList(String appAcronym) | Returns the details of the Countries subscribed by the requestor. |
| getCountry(CountrySearchCriteria searchMsg) | Returns the details of the Country identified by a given search parameter.  Inside this message, one of the following search parameters must be provided in order to perform the search:   |  |  | | --- | --- | | **Parameter** | **Meaning** | | NumericCode | Returns the country identified by a given ‘numericCode’. | | Alpha2Code | Returns the country identified by a given ‘alpha2Code’. | | Alpha3Code | Returns the country identified by a given ‘alpha3Code’. | | Callsign | Returns the country identified by a given ‘callsign’. | | MID | Returns the country identified by a given ‘MID’. | | LRIT\_ID | Returns the country identified by a given ‘lritId’. | |
| getParentCountry(String alpha2Code) | Returns the details of the Parent Country of the Country identified by a given ‘alpha2Code’. |
| getChildCountryList (String alpha2Code) | Returns the details of the Child Countries of a Country identified by a given ‘alpha2Code’. |
| getCountryMIDList(String alpha2Code) | Returns all the MID’s associated with a Country identified by a given ‘alpha2Code’. |
| getCountryCallsignList(String alpha2Code) | Returns all the Callsign’s associated with Country identified by a given ‘alpha2Code’. |
| getCountryFlagImageList() | Returns the flag images (small, medium, large) of all Countries. |
| getCountryFlagImages(String alpha2Code) | Returns the flag images (small, medium, large) of a Country identified by the given ‘alpha2Code’. |
| getCountryFlagImage(String alpha2Code, Integer imageType) | Returns a specific flag image size of a Country identified by the given ‘alpha2Code’. |
| getCountryFlagImageLocation(String alpha2Code, Integer imageType) | Returns the full location (URL) of the flag image with a specific size (small, medium or large) of a specific Country.  Ex:  *If the provided imageType is 0:*   * *http://CCBR/images/countryFlags/PT\_S.png*   *If the provided imageType is 1:*   * *http://CCBR/images/countryFlags/PT\_M.png*   *If the provided imageType is 2:*   * *http://CCBR/images/countryFlags/PT\_L.png* |
| getCountryFlagImagesLocationContext() | Returns the context (URL) of the country flag images location.  *Ex:*  *The context is : http://CCBR/images/countryFlags/*  *The client should add to the URL the flag image name in order to retrieve the image itself. The flag image name is composed by : “<alpha2Code>"+ ”\_” +”<imageSize>”+”.png”*  ***Note****: the ImageSize may assume the following values: “S” “M” or “L”.* |
| getCountryHistory(HistorySearchCriteria searchMsg) | Returns the history country information in a specific time or a period of time start and end dates are included in the boundaries when present |
| getCountryTypes | Returns a list with the existent Country Types. |
| getCountryCategories | Returns a list with the existent Country Categories. |
| getSubscribedCountries | Returns only Countries subscribed by a specific external system. |
| getSubscribedRegionalAgreements | Returns only Regional Agreements subscribed by a specific external system. |
| getSubscribedCountryTypes | Returns only the Country Types subscribed by a specific external system. |

|  |  |
| --- | --- |
| **AnnouncementService** | |
| Service Operation | Service Description |
| notify(SubscriberAnnouncement announcement) | Web Method to be called by the Country Base Registry in order to deliver a change announcement to an External System. |

## Subscription Web Service

This web service is used to handle subscriptions made by external systems in order to be informed about modifications (whenever a subscribed item was updated/deleted or new items were created) in the country base registry data.

The list of external systems already defined is presented in the Table 1 – External Systems. This list can be updated when a new external system wants to start using the CCBR.

|  |  |
| --- | --- |
| **Application** | **appAcronym** |
| THETIS | THETIS |
| SafeSeaNet | SSN |
| CleanSeaNet | CSN |
| STCW-IS | STCW |
| RuleCheck | RCHECK |
| MaKCs | MAKCS |
| LRIT DC | LRITDC |
| LRIT I&B | LRITIB |
| LRIT IDE | LRITIDE |
| LTIT Ship DB | LRITSDB |
| IMDatE | IMDATE |
| MARSURV | MSURV |
| MARSURV 2 | MSURV2 |

**Table 1 – External Systems**

### SignSubscription

This method is used by clients for signing for the subscription service.

* **Request Parameter(s):**

This web method will receive two input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | String | This parameter identifies the external application subscribing the service and must contain one of the values presented in Table 1 . |
| AnnouncementEndpoint | String | This parameter is the web service end-point provided by the external application and is used by the Country Base Registry to send announcements. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 40 | NOK – NOK, the received appAccronym is already signed up. | | 100 | NOK – Internal Error. | |

### updateSubscription

This method is used by clients for updating the announcement endpoint.

* **Request Parameter(s):**

This web method will receive two input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | String | This parameter identifies the external application subscribing the service and may contain one of the values presented in Table 1 . |
| announcementEndpoint | String | This parameter is the web service end-point provided by the external application and is used by the Country Base Registry to send announcements. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 50 | NOK - The received appAccronym has no subscription service | | 100 | NOK – Internal Error | |

### revokeSubscription

This method is used by clients for revoking the subscription service.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | String | This parameter identifies the external application subscribing the service and may contain one of the values presented in Table 1 . |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 50 | NOK – The received appAcronym has no subscription service | | 100 | NOK – Internal Error | |

### subscribe

This method is used by clients for subscribing the announcement service for one specific country, for all countries, or a group of pre-defined countries according to a set subscription criteria: belonging to a list of regional agreements or a list of country types.

* **Request Parameter(s):**

This web method will receive three input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | String | This parameter identifies the external application subscribing the service and must contain one of the values presented in Table 1 . |
|  | SubscibeCriterias | The input parameter of type “SubscibeCriterias” is defined by the schema CBR\_ SubscibeCriterias\_1\_0.xsd.  Inside this message, at least one of the following parameters must be provided in order to perform the subscribe operation.  Parameters are combined using AND while several occurances of the same parameter are combined using OR.   |  |  |  | | --- | --- | --- | | **Name** | **Type** | **Meaning** | | Country | String | This parameter represents the Country (through alpha2code) you want to subscribe for.  (NOTE: If this parameter contains the keyword “AA”, the Country Base Registry will execute the subscription of the service for all existing countries.) | | Regional Agreement | List of Strings | This parameter may be used to subscribe all countries belonging to a specific Regional Agreement.  A list of Regional agreements may be used in this parameter  The folloing values are valid for this parameter:   * XH - Helcom * XP – ParisMoU * XW - Wetrep * XR - Barcelona * XA – Bonn Agreement * XB - Black Sea Commission   be added in the future. | | County Type | List of Strings | This parameter may be used to subscribe all countries belonging to a specific country Tyoe.  A list of Country types may be used in this parameter  The folloing values are valid for this parameter:   * MEMBER\_STATE – EU Member State * OVERSEAS\_TERRITORY - Overseas Territory * EU\_ACCEDING\_COUNTRY - EU Acceding Country * EU\_CANDIDATE\_COUNTRY – EU Candidate Country * EFTA - EFTA * EU\_ECONOMIC\_AREA - European Economic Area * FLAG\_STATE – Flag State * EUROPEAN\_UNION – European Union   **Note**: other types may be added in the future. | |

**Note:** No ERROR is returned when a parameter refers to a (group of) already subscribed country (/countries).

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### unsubscribe

This method is used by clients for unsubscribing the announcement service for one specific country or for all countries.

* **Request Parameter(s):**

This web method will receive three input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | AppAcronym | This parameter identifies the external application unsubscribing the service and must contain one of the values presented in Table 1 . |
|  | UnSubscibeCriterias | The input parameter of type “SubscibeCriterias” is defined by the schema CBR\_ SubscibeCriterias\_1\_0.xsd.  Inside this message, at least one of the following parameters must be provided in order to perform the subscribe operation:   |  |  |  | | --- | --- | --- | | **Name** | **Type** | **Meaning** | | Country | String | This parameter represents the Country (through alpha2code) you want to unsubscribe for.  (NOTE: If this parameter contains the keyword “AA”, the Country Base Registry will unsubscribe the external application of the service for all existing countries.) | | Regional Agreement | List of Strings | This parameter may be used to unsubscribe all countries belonging to a specific Regional Agreement.  A list of Regional agreements may be used in this parameter  The folloing values are valid for this parameter:   * XH - Helcom * XP – ParisMoU * XW - Wetrep * XR - Barcelona * XA – Bonn Agreement * XB - Black Sea Commission   be added in the future. | | County Type | List of Strings | This parameter may be used to unsubscribe all countries belonging to a specific country Tyoe.  A list of Country types may be used in this parameter  The folloing values are valid for this parameter:   * MEMBER\_STATE – EU Member State * OVERSEAS\_TERRITORY - Overseas Territory * EU\_ACCEDING\_COUNTRY - EU Acceding Country * EU\_CANDIDATE\_COUNTRY – EU Candidate Country * EFTA - EFTA * EU\_ECONOMIC\_AREA - European Economic Area * FLAG\_STATE – Flag State * EUROPEAN\_UNION – European Union   **Note**: other types may be added in the future. | |

**Note:** No ERROR is returned when a parameter refers to a (group of) already unsubscribed country(/countries).

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### getServiceSubscribers

This method is used to return the list of external systems with subscribed services.

* **Request Parameter(s):**

This web method will not receive input parameters.

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| SubscriberDetails | The response of type “SubscriberDetails” is defined by the schema CBR\_SubscriberDetails\_1\_0.xsd.  Inside the SubscriberDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK - No data to return | | 100 | NOK – Internal Error | |

### getSubscriptionOperations

This method is used to return the list of operations performed by external systems and has in consideration the provided search criteria (start and end dates, both included in boundaries when mentioned). If no dates are provided, this method will return the full list of operations by made by the external system.

* **Request Parameter(s):**

This web method will receive three input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | String | This parameter identifies the external application for which is requested the list of operations performed. This paramemet may contain one of the values presented in Table 1 . |
| startDate | javax.xml.datatype.XMLGregorianCalendar | This parameter is mandatory and represents the search criteria start date. |
| endDate | javax.xml.datatype.XMLGregorianCalendar | This parameter is mandatory and represents the search criteria end date. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| SubscriptionOperations | The response of type “SubscriptionOperations” is defined by the schema CBR\_SubscriptionOperations\_1\_0.xsd.  Inside the SubscriptionOperations message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 30 | NOK – Received input parameters are not valid. | | 100 | NOK – Internal Error | |

### getServiceSubscribersList

This method is used to return the list of subscribed external systems for a specific country. The service subscribers presented in this list will receive a announcement about an operation (insert, update or delete) performed over a country. The announcement is send by the CCBR OSB services when it receives a request for execute a create, update or delete request.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2code | String | This parameter uniquely identifies a Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| ServiceSubscribers | The response of type “ServiceSubscribers” is defined by the schema CBR\_ ServiceSubscribers\_1\_0.xsd.  Inside the SubscriptionOperations message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK - No data to return | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### updateService announcementStatus

This method is used to store in the CCBR database information about the announcements sent to the ES by the CCBR OSB services.

* **Request Parameter(s):**

This web method will receive four input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| AppAcronym | AppAcronym | This parameter identifies uniquely a ES, In this case it represents the ES receiving the announcement. |
| Alpha2Code | Alpha2Code | This parameter identifies uniquely a Country. In this case it represents the country created, updated or deleted. |
| AnnouncementType | AnnouncementType | This parameter identifies the type of announcement sent.  There are 3 (three) possible types of announcement:   * C – Announcement of the ES about a Country creation * U – Announcement of the ES about an update in a subscribed Country. * D – Announcement of the ES about a deletion of a subscribed Country. |
| AnnouncementStatus | AnnouncementStatus | This parameter defines wether the announcement was successful (1) or not (0). |
| AnnouncementTimestamp | AnnouncementTimestamp (javax.xml.datatype.XMLGregorianCalendar) | The date/time when the announcement was sent to the ES. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| AnnouncementStatus | The response of type “AnnouncementStatus” is defined by the schema CBR\_ ServiceSubscribers\_1\_0.xsd.  Inside the SubscriptionOperations message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received parameter(s) do(e)s not exist in the reference list. | | 100 | NOK – Internal Error | |

## Manage Countries Web Service

This web service is used to manage (add, update or delete) information about countries and associated MID’s and Callsign’s.

### insertCountry

This method is used to insert a new country in the database.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| Country | The input parameter of type “Country” is defined by the schema CBR\_Country\_1\_0.xsd.  **Note:** the values of the RecordTimestamp and the CreationTimestamp parameters sent are ignored. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 60 | NOK – Unable to insert Country. | | 100 | NOK – Internal Error | |

### updateCountry

This method is used to update the data of an existent country in the database.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| Country | The input parameter of type “Country” is defined by the schema CBR\_Country\_1\_0.xsd.  **Note:** the values of the RecordTimestamp and the CreationTimestamp parameters sent are ignored. |

* **Response**

This web method returns the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 70 | NOK – Unable to update Country. | | 100 | NOK – Internal Error | |

### deleteCountry

This method is used to delete a country from the database.

* **Request Parameter:**

This web method accepts one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter identifies uniquely the Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK, the received parameter(s) do(es) not exist in the reference database | | 100 | NOK – Internal Error | |

### insertMID

This method is used to insert a new MID in the database.

* **Request Parameter:**

This web method accepts one input parameter:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| MID | The input parameter of type “MID” is defined by the schema CBR\_MID\_1\_0.xsd.  **Note:** the values of the RecordTimestamp and the CreationTimestamp parameters sent are ignored. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 60 | NOK – Unable to insert MID. | | 100 | NOK – Internal Error | |

### updateMID

This method is used to update the data of an existent MID in the database.

* **Request Parameter(s):**

This web method will receive one input parameters:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| MID | The input parameter of type “MID” is defined by the schema CBR\_MID\_1\_0.xsd.  **Note:** the values of the RecordTimestamp and the CreationTimestamp parameters sent are ignored. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 70 | NOK – Unable to update MID. | | 100 | NOK – Internal Error | |

### deleteMID

This method is used to delete an existent MID from the database.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| midID | MidID | This parameter uniquely identifies the MID |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK - the received parameter(s) do(es) not exist in the reference database | | 100 | NOK – Internal Error | |

### insertCallsign

This method is used to insert a new Callsign in the database.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| Callsign | The input parameter of type “Callsign” is defined by the schema CBR\_Callsign\_1\_0.xsd.  **Note:** the values of the RecordTimestamp and the CreationTimestamp parameters sent are ignored. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 60 | NOK – Unable to insert Callsign. | | 100 | NOK – Internal Error | |

### updateCallsign

This method is used to update the data of an existent Callsign in the database.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| Callsign | The input parameter of type “Callsign” is defined by the schema CBR\_Callsign\_1\_0.xsd.  **Note:** the values of the RecordTimestamp and the CreationTimestamp parameters sent are ignored. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 70 | NOK – Unable to update Callsign. | | 100 | NOK – Internal Error | |

### deleteCallsign

This method is used to update the data of an existent callsign in the database.

* **Request Parameter(s):**

This web method will receive one input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| callsignID | String | This parameter identifies uniquely the Callsign |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK - the received parameter(s) do(es) not exist in the reference database | | 100 | NOK – Internal Error | |

## Country Information Web Service

This web service is used to request information about countries and associated MID’s and Callsign’s.

### getCountryList

This method is used to return the list of countries and respective details and has in consideration the provided search criteria. The search criteria will work as an “AND” operator, meaning that this method will return only the countries matching with all the provided input parameters.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |
| --- | --- |
| Parameter Type | Parameter Description |
| CountryDetailedSearchCriteria | The input parameter of type “CountryDetailedSearchCriteria” is defined by the schema CBR\_CountryDetailedSearchCriteria\_1\_0.xsd.  Inside this message, at least one of the following parameters must be provided in order to perform the search.  Parameters are combined using AND while several occurances of the same parameter are combined using OR.   |  |  |  | | --- | --- | --- | | **Name** | **Type** | **Meaning** | | CountryTypesList | CountryType | This parameter may be used to pass a list of country types to be considered during the search.  The folloing values are valid for this parameter:   * MEMBER\_STATE – EU Member State * OVERSEAS\_TERRITORY - Overseas Territory * EU\_ACCEDING\_COUNTRY - EU Acceding Country * EU\_CANDIDATE\_COUNTRY – EU Candidate Country * EFTA - EFTA * EU\_ECONOMIC\_AREA - European Economic Area * FLAG\_STATE – Flag State * EUROPEAN\_UNION – European Union   **Note**: other types may be added in the future. | | RegionalAgreementsList | RegionalAgreement | This parameter may be used to pass a list of regional agreements to be considered during the search.  The following values are valid for this parameter:   * XH - Helcom * XP – ParisMoU * XW - Wetrep * XR - Barcelona * XA – Bonn Agreement * XB - Black Sea Commission   **Note**: other agreements may be added in the future. | | CategoryTppe | CategoryCode | This parameter may be used to pass a list of Category type to be considered during the search.  The following values are valid for this parameter:   * COUNTRY - Country * VIRTUAL\_COUNTRY - Virtual Country * REGIONAL\_AGREEMENT - Regional Agreement * INSTITUTION - Institution * COMPANY – Company | |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return. | | 20 | NOK - the received parameter(s) do(es) not exist in the reference database. | | 100 | NOK – Internal Error. | |

### getCountryFullList

This method is used to return the full list of countries and respective details.

* **Request Parameter(s):**

This web method will not receive input parameters.

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 100 | NOK – Internal Error | |

### getSubscribedCountryList

This method is used to return the list of subscribed countries and respective details for a specific external application. It includes also all countries added because they are part of a Regional Agreement or Country Type.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | AppAcronym | This parameter identifies the external application requesting the service and may contain one of the values presented in Table 1 . |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 50 | NOK – The received appAcronym has no subscription service | | 100 | NOK – Internal Error | |

### getCountry

This method is used to return the country with respective details.

* **Request Parameter(s):**

This web method will receive one input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| searchMsg | CountrySearchCriteria | The input parameter of type “CountrySearchCriteria” is defined by the schema CBR\_CountrySearchCriteria\_1\_0.xsd.  Inside this message, one of the following parameters must be provided in order to perform the search:   |  |  | | --- | --- | | **Parameter** | **Meaning** | | NumericCode | Returns the country identified by a given ‘numericCode’. | | Alpha2Code | Returns the country identified by a given ‘alpha2Code’. | | Alpha3Code | Returns the country identified by a given ‘alpha3Code’. | | Callsign | Returns the country identified by a given ‘callsign’. | | MID | Returns the country identified by a given ‘MID’. | | LRIT\_ID | Returns the country identified by a given ‘lritId’. | |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK - No data to return | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 30 | NOK – Received input parameters are not valid. | | 100 | NOK – Internal Error | |

### getParentCountry

This method is used to return the details of the parent country. The child country is identified by the input parameter.

* **Request Parameter(s):**

This web method will receive one input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | String | This parameter identifies uniquely a child Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails \_1.0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK - No data to return | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### getChildCountryList

This method is used to return the details of the child countries of a parent country. The parent country is identified by the input parameter.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter uniquely identifies a parent Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails \_1.0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK - No data to return | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### getCountryMIDList

This method is used to return all the MID’s (Maritime Identification Digits) associated with a specific country. The country is identified by the input parameter.

* **Request Parameter(s):**

This web method will receive one input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter identifies uniquely a Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryMIDs | The response of type “CountryMIDs” is defined by the schema CBR\_CountryMIDs\_1\_0.xsd.  Inside the CountryMIDs message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### getCountryCallsignList

This method is used to return all the Callsign’s associated with a specific country. The country is identified by the input parameter.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter uniquely identifies a Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryCallSigns | The response of type “CountryCallSigns” is defined by the schema CBR\_CountryCallSigns\_1\_0.xsd.  Inside the CountryCallSigns message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### getCountryFlagImageList

This method is used to return all the flag images (small, medium, large) of all existing Countries.

* **Request Parameter(s):**

This web method will not receive input parameters.

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryImages | The response of type “CountryImages” is defined by the schema CBR\_CountryImages\_1\_0.xsd.  Inside the CountryImages message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 100 | NOK – Internal Error | |

### getCountryFlagImages

This method is used to return the flag images (small, medium, large) of a specific Country.

* **Request Parameter:**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter uniquely identifies the Country |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryImages | The response of type “CountryImages” is defined by the schema CBR\_CountryImages\_1\_0.xsd.  Inside the CountryImages message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

### getCountryFlagImage

This method is used to return a flag image with a specific size (small, medium or large) of a specific Country.

* **Request Parameter(s):**

This web method will receive two input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter uniquely identifies the Country |
| imageType | ImageType | This parameter is used to return the flag image having in consideration the size requested.  Three values are valid for this parameter:   * **S** – Returns the small flag image. * **M** – Returns the medium flag image. * **L** – Returns the large flag image. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryImages | The response of type “CountryImages” is defined by the schema CountryImages.xsd.  Inside the CountryImages message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return. | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error. | |

### getCountryFlagImageLocation

This method is used to return the location (URL) of the flag image with a specific size (small, medium or large) of a specific Country.

* **Request Parameters:**

This web method receives two input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| alpha2Code | Alpha2Code | This parameter identifies uniquely the Country |
| imageType | ImageType | This parameter is used to return the flag image location having in consideration the size requested.  Three values are valid for this parameter:   * **S** – Returns the small flag image. * **M** – Returns the medium flag image. * **L** – Returns the large flag image. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryFlagImageURL | The response of type “CountryFlagImageURL” is defined by the schema CountryImageURL.xsd.  Inside the CountryFlagImageURL message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 100 | NOK – Internal Error | |

**Note:** In case there is no flag to return, the returned URL will point to a blank flag image.

### getCountryFlagImageLocationContext

This method is used to return the location context (URL) of the country flag images location.

* **Request Parameters:**

This web method does not accept input parameters.

* **Response**

This web method returns the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryFlagImageURL | The response of type “CountryFlagImageURL” is defined by the schema CountryImageURL.xsd.  Inside the CountryFlagImageURL message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 100 | NOK – Internal Error | |

### getCountryHistory

This method is used to return the history country information in a specific time or a period of time. Start and end dates are included in the boundaries when present.

* **Request Parameter(s):**

This web method accepts three input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| searchMsg | CountryHistorySearchCriteria | The input parameter of type “CountryHistorySearchCriteria” is defined by the schema CBR\_CountryHistorySearchCriteria\_1\_0.xsd.  All parameters must be provided in order to perform a search.   |  |  | | --- | --- | | **Parameter** | **Meaning** | | Alpha2Code | Returns the country identified by a given ‘alpha2Code’. | | StartDate | This parameter is mandatory and represents the search criteria start date. | | EndDate | This parameter is mandatory and represents the search criteria end date. | |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received parameter does not exist in the Country Base Registry database. | | 30 | NOK – Received input parameters are not valid. | | 100 | NOK – Internal Error | |

### getSubscribedCountries

This method is used to return the list of individually subscribed countries and respective details for a specific external application. It does not include countries added because they are part of a Regional Agreement or Country Type.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | AppAcronym | This parameter identifies the external application requesting the service and may contain one of the values presented in Table 1 . |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 50 | NOK – The received appAcronym has no subscription service | | 100 | NOK – Internal Error | |

### getSubscribedCountryTypes

This method is used to return the list of subscribed country types.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | AppAcronym | This parameter identifies the external application requesting the service and may contain one of the values presented in Table 1 . |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryTypes | The response of type “CountryTypes” is defined by the schema CBR\_CountryTypes\_1\_0.xsd.  Inside the CountryTypes message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 50 | NOK – The received appAcronym has no subscription service | | 100 | NOK – Internal Error | |

### getCountryCategories

This method is used to return the list of country categories

* **Request Parameter(s):**

This web method will not receive input parameters.

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryCategories | The response of type “CountryCategories” is defined by the schema CBR\_CountryCategories\_1\_0.xsd.  Inside the CountryCategories message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 100 | NOK – Internal Error | |

### getCountryTypes

This method is used to return the list of country types

* **Request Parameter(s):**

This web method will not receive input parameters.

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryTypes | The response of type “CountryTypes” is defined by the schema CBR\_ CountryTypes\_1\_0.xsd.  Inside the CountryTypes message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 100 | NOK – Internal Error | |

### getSubscribedRegionalAgreements

This method is used to return the list of subscribed country types.

* **Request Parameter(s):**

This web method will receive one input parameter:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| appAcronym | AppAcronym | This parameter identifies the external application requesting the service and may contain one of the values presented in Table 1 . |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| CountryDetails | The response of type “CountryDetails” is defined by the schema CBR\_CountryDetails\_1\_0.xsd.  Inside the CountryDetails message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 10 | OK – No data to return | | 20 | NOK – The received appAcronym does not exist in the reference list. | | 50 | NOK – The received appAcronym has no subscription service | | 100 | NOK – Internal Error | |

## Announcement Web Service

This web service is used to send announcements to external systems.

### notify

This method is used by the country base registry to deliver a change announcement to an external system.

* **Request Parameter(s):**

This web method will receive one input parameters:

|  |  |  |
| --- | --- | --- |
| Parameter Name | Parameter Type | Parameter Description |
| announcement | SubscriberAnnouncement | This parameter is defined by the schema CBR\_SubscriberAnnouncement\_1\_0.xsd. |

* **Response**

This web method will return the following response:

|  |  |
| --- | --- |
| Response Type | Parameter Description |
| Receipt | The response of type “Receipt” is defined by the schema CBR\_Receipt\_1\_0.xsd.  Inside the receipt message, the ResponseType parameter may assume one of the following values:   |  |  | | --- | --- | | **Response** | **Meaning** | | 0 | OK – The request was successfully processed. | | 100 | NOK – Internal Error | |

## User Groups and Access Rights

WS-Security (Web Services Security) an extension to SOAP is used to apply security to web services. In the CCBR it is used to pass credentials to authenticate the user to the OSB, no encryption is used.

The OSB will use the services of the EMSA Open LDAP service in order to validate the user’s authentication and authorization.

When LifeRay connects to the OSB an enduser name has to be passed in the SOAP header. The name of the field passing the end user is CCBR\_ENDUSER. Other applications can also pass this field.

The CCBR\_ENDUSER field is only used for logging purposes, not for authentication or authorization.

Example WS-Security header:

<soapenv:Header>  
<wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<wsse:UsernameToken wsu:Id="UsernameToken824">

<wsse:Username>CCBR\_SYS\_ADMIN</wsse:Username>

<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">CCBR\_SYS\_ADMIN</wsse:Password>

<wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">ytPuvtE8PYR/icS7fH23Wg==</wsse:Nonce><wsu:Created>2013-09-13T15:24:57.088Z</wsu:Created>

</wsse:UsernameToken></wsse:Security>

<ccbr:CCBR\_ENDUSER>Test Enduser</ccbr:CCBR\_ENDUSER>

</soapenv:Header>

Username: System user connecting to the OSB  
Password: Password of the user connecting to the OSB  
CCBR\_ENDUSER: End user logged into the system represented by Username. This field is obligatory for the LifeRay system user and optional for all others; however the field will not be enforced, only logged.

Roles

The CCBR system has five roles; CCBRSysAdmin, CCBRSysMember, CCBRAdmin, CCBRMember, CCBRReader.

The table below shows the access rights for the different roles for all operations:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Web Service** | **Operation** | **CBBRSysAdmin** **CCBRAdmin** | **CCBRSysMember CCBRMember** | **CCBRReader** |
| **Subscription** |  |  |  |  |
|  | signSubscription | **V** | **X** | **X** |
|  | updateSubscription | **V** | **X** | **X** |
|  | revokeSubscription | **V** | **X** | **X** |
|  | subscribe | **V** | **V** | **X** |
|  | unsubscribe | **V** | **V** | **X** |
|  | getServiceSubscribers | **V** | **V** | **X** |
|  | getSubscriptionOperations | **V** | **V** | **X** |
|  | getServiceSubscribersList | **V** | **V** | **X** |
|  | updateService announcementStatus | **V** | **V** | **X** |
| **Country Information** |  |  |  |  |
|  | getCountryList | **V** | **V** | **V** |
|  | getCountryFullList | **V** | **V** | **V** |
|  | getSubscribedCountryList | **V** | **V** | **V** |
|  | getCountry | **V** | **V** | **V** |
|  | getParentCountry | **V** | **V** | **V** |
|  | getChildCountryList | **V** | **V** | **V** |
|  | getCountryMIDList | **V** | **V** | **V** |
|  | getCountryCallsignList | **V** | **V** | **V** |
|  | getCountryFlagImageList | **V** | **V** | **V** |
|  | getCountryFlagImages | **V** | **V** | **V** |
|  | getCountryFlagImage | **V** | **V** | **V** |
|  | getCountryFlagImageLocation | **V** | **V** | **V** |
|  | getCountryFlagImageLocationContext | **V** | **V** | **V** |
|  | getCountryHistory | **V** | **V** | **V** |
|  | getCountryCategories | **V** | **V** | **V** |
|  | getCountryTypes | **V** | **V** | **V** |
|  | getSubscribedCountries | **V** | **V** | **V** |
|  | getSubscribedCountryTypes | **V** | **V** | **V** |
|  | getSubscribedRegionalAgreements | **V** | **V** | **V** |
| **Manage Countries** |  | **V** | **V** | **V** |
|  | insertCountry | **V** | **X** | **X** |
|  | updateCountry | **V** | **X** | **X** |
|  | deleteCountry | **V** | **X** | **X** |
|  | insertMID | **V** | **X** | **X** |
|  | updateMID | **V** | **X** | **X** |
|  | deleteMID | **V** | **X** | **X** |
|  | insertCallsign | **V** | **X** | **X** |
|  | updateCallsign | **V** | **X** | **X** |
|  | deleteCallsign | **V** | **X** | **X** |

# Annex I

This annex contains the wsdl for the four mentioned web services:

## SubscriptionService

*http://<host>:<port/CountryBaseRegistryWS/CountryBaseRegistrySubscription?wsdl*

***CountryBaseRegistrySubscription?wsdl:***

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://ccbr.emsa.europa.eu/" name="CountryBaseRegistrySubscriptionService">  <types>  <xsd:schema>  <xsd:import namespace="http://ccbr.emsa.europa.eu/" schemaLocation="http://172.16.120.206:7033/ShipDB-CountryBaseRegistryWS/CountryBaseRegistrySubscriptionService?xsd=1"/>  </xsd:schema>  </types>  <message name="signSubscription">  <part name="parameters" element="tns:signSubscription"/>  </message>  <message name="signSubscriptionResponse">  <part name="parameters" element="tns:signSubscriptionResponse"/>  </message>  <message name="updateSubscription">  <part name="parameters" element="tns:updateSubscription"/>  </message>  <message name="updateSubscriptionResponse">  <part name="parameters" element="tns:updateSubscriptionResponse"/>  </message>  <message name="revokeSubscription">  <part name="parameters" element="tns:revokeSubscription"/>  </message>  <message name="revokeSubscriptionResponse">  <part name="parameters" element="tns:revokeSubscriptionResponse"/>  </message>  <message name="subscribe">  <part name="parameters" element="tns:subscribe"/>  </message>  <message name="subscribeResponse">  <part name="parameters" element="tns:subscribeResponse"/>  </message>  <message name="unsubscribe">  <part name="parameters" element="tns:unsubscribe"/>  </message>  <message name="unsubscribeResponse">  <part name="parameters" element="tns:unsubscribeResponse"/>  </message>  <message name="getServiceSubscribers">  <part name="parameters" element="tns:getServiceSubscribers"/>  </message>  <message name="getServiceSubscribersResponse">  <part name="parameters" element="tns:getServiceSubscribersResponse"/>  </message>  <message name="getSubscriptionOperations">  <part name="parameters" element="tns:getSubscriptionOperations"/>  </message>  <message name="getSubscriptionOperationsResponse">  <part name="parameters" element="tns:getSubscriptionOperationsResponse"/>  </message>  <message name="getServiceSubscribersList">  <part name="parameters" element="tns:getServiceSubscribersList"/>  </message>  <message name="getServiceSubscribersListResponse">  <part name="parameters" element="tns:getServiceSubscribersListResponse"/>  </message>  <message name="updateServiceAnnouncementStatus">  <part name="parameters" element="tns:updateServiceAnnouncementStatus"/>  </message>  <message name="updateServiceAnnouncementStatusResponse">  <part name="parameters" element="tns:updateServiceAnnouncementStatusResponse"/>  </message>  <portType name="CountryBaseRegistrySubscription">  <operation name="signSubscription">  <input message="tns:signSubscription"/>  <output message="tns:signSubscriptionResponse"/>  </operation>  <operation name="updateSubscription">  <input message="tns:updateSubscription"/>  <output message="tns:updateSubscriptionResponse"/>  </operation>  <operation name="revokeSubscription">  <input message="tns:revokeSubscription"/>  <output message="tns:revokeSubscriptionResponse"/>  </operation>  <operation name="subscribe">  <input message="tns:subscribe"/>  <output message="tns:subscribeResponse"/>  </operation>  <operation name="unsubscribe">  <input message="tns:unsubscribe"/>  <output message="tns:unsubscribeResponse"/>  </operation>  <operation name="getServiceSubscribers">  <input message="tns:getServiceSubscribers"/>  <output message="tns:getServiceSubscribersResponse"/>  </operation>  <operation name="getSubscriptionOperations">  <input message="tns:getSubscriptionOperations"/>  <output message="tns:getSubscriptionOperationsResponse"/>  </operation>  <operation name="getServiceSubscribersList">  <input message="tns:getServiceSubscribersList"/>  <output message="tns:getServiceSubscribersListResponse"/>  </operation>  <operation name="updateServiceAnnouncementStatus">  <input message="tns:updateServiceAnnouncementStatus"/>  <output message="tns:updateServiceAnnouncementStatusResponse"/>  </operation>  </portType>  <binding name="CountryBaseRegistrySubscriptionPortBinding" type="tns:CountryBaseRegistrySubscription">  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>  <operation name="signSubscription">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="updateSubscription">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="revokeSubscription">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="subscribe">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="unsubscribe">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getServiceSubscribers">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getSubscriptionOperations">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getServiceSubscribersList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="updateServiceAnnouncementStatus">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  </binding>  <service name="CountryBaseRegistrySubscriptionService">  <port name="CountryBaseRegistrySubscriptionPort" binding="tns:CountryBaseRegistrySubscriptionPortBinding">  <soap:address location="http://172.16.120.206:7033/ShipDB-CountryBaseRegistryWS/CountryBaseRegistrySubscriptionService"/>  </port>  </service>  </definitions> |

**CountryBaseRegistrySubscriptionService?xsd=1:**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <xs:schema xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ccbr.emsa.europa.eu/" elementFormDefault="qualified" version="1.0">  <xs:element name="Alpha2Code">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="AnnouncementEndpoint">  <xs:complexType>  <xs:sequence>  <xs:element name="AnnouncementEndpoint" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="AppAcronym">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="Receipt">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="ServiceAnnouncementStatus">  <xs:complexType>  <xs:sequence>  <xs:element name="SubscribersAnnouncementList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="SubscriberAnnouncement" type="tns:SubscriberAnnouncementType" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="ServiceSubscribers">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int" form="unqualified"/>  <xs:element name="SubscribersList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="Subscriber" type="tns:SubscriberShortType" form="unqualified" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="SubsCriteria">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="RegionalAgreementsList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" type="xs:string" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="TypesList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="xs:string" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="SubscriberDetails">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int" form="unqualified"/>  <xs:element name="SubscriberList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="Subscriber" type="tns:SubscriberType" form="unqualified" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="SubscriptionOperations">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int" form="unqualified"/>  <xs:element name="Subscriber" type="xs:string" form="unqualified"/>  <xs:element name="SubscriptionOperationsList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="Operation" type="tns:OperationDetailsType" form="unqualified" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="getServiceSubscribers" type="tns:getServiceSubscribers"/>  <xs:element name="getServiceSubscribersList" type="tns:getServiceSubscribersList"/>  <xs:element name="getServiceSubscribersListResponse" type="tns:getServiceSubscribersListResponse"/>  <xs:element name="getServiceSubscribersResponse" type="tns:getServiceSubscribersResponse"/>  <xs:element name="getSubscriptionOperations" type="tns:getSubscriptionOperations"/>  <xs:element name="getSubscriptionOperationsResponse" type="tns:getSubscriptionOperationsResponse"/>  <xs:element name="revokeSubscription" type="tns:revokeSubscription"/>  <xs:element name="revokeSubscriptionResponse" type="tns:revokeSubscriptionResponse"/>  <xs:element name="signSubscription" type="tns:signSubscription"/>  <xs:element name="signSubscriptionResponse" type="tns:signSubscriptionResponse"/>  <xs:element name="subscribe" type="tns:subscribe"/>  <xs:element name="subscribeResponse" type="tns:subscribeResponse"/>  <xs:element name="unsubscribe" type="tns:unsubscribe"/>  <xs:element name="unsubscribeResponse" type="tns:unsubscribeResponse"/>  <xs:element name="updateServiceAnnouncementStatus" type="tns:updateServiceAnnouncementStatus"/>  <xs:element name="updateServiceAnnouncementStatusResponse" type="tns:updateServiceAnnouncementStatusResponse"/>  <xs:element name="updateSubscription" type="tns:updateSubscription"/>  <xs:element name="updateSubscriptionResponse" type="tns:updateSubscriptionResponse"/>  <xs:complexType name="updateSubscription">  <xs:sequence>  <xs:element name="appAcronym" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="announcementEndpoint" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AnnouncementEndpoint" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateSubscriptionResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="signSubscription">  <xs:sequence>  <xs:element name="appAcronym" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="announcementEndpoint" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AnnouncementEndpoint" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="signSubscriptionResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="unsubscribe">  <xs:sequence>  <xs:element name="appAcronym" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="unsubscribeCriteriaMsg" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="RegionalAgreementsList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" type="xs:string" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="TypesList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="xs:string" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="unsubscribeResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="revokeSubscription">  <xs:sequence>  <xs:element name="appAcronym" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="revokeSubscriptionResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getServiceSubscribers">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getServiceSubscribersResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int" form="unqualified"/>  <xs:element name="SubscriberList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="Subscriber" type="tns:SubscriberType" form="unqualified" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="SubscriberType">  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  <xs:element name="AnnouncementEndpoint" type="xs:string" form="unqualified"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified"/>  <xs:element name="UpdateTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getServiceSubscribersList">  <xs:sequence>  <xs:element name="alpha2code" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getServiceSubscribersListResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int" form="unqualified"/>  <xs:element name="SubscribersList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="Subscriber" type="tns:SubscriberShortType" form="unqualified" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="SubscriberShortType">  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  <xs:element name="AnnouncementEndpoint" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateServiceAnnouncementStatus">  <xs:sequence>  <xs:element name="announcementStatus" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="SubscribersAnnouncementList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="SubscriberAnnouncement" type="tns:SubscriberAnnouncementType" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="SubscriberAnnouncementType">  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="AnnouncementType" type="xs:string" form="unqualified"/>  <xs:element name="AnnouncementStatus" type="xs:int" form="unqualified"/>  <xs:element name="AnnouncementTimestamp" type="xs:anySimpleType" form="unqualified"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateServiceAnnouncementStatusResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscriptionOperations">  <xs:sequence>  <xs:element name="appAcronym" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="startDate" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="endDate" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscriptionOperationsResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int" form="unqualified"/>  <xs:element name="Subscriber" type="xs:string" form="unqualified"/>  <xs:element name="SubscriptionOperationsList" form="unqualified">  <xs:complexType>  <xs:sequence>  <xs:element name="Operation" type="tns:OperationDetailsType" form="unqualified" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="OperationDetailsType">  <xs:sequence>  <xs:element name="OperationTimestamp" type="xs:anySimpleType" form="unqualified"/>  <xs:element name="OperationDescription" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="subscribe">  <xs:sequence>  <xs:element name="appAcronym" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="subscribeCriteriaMsg" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="RegionalAgreementsList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" type="xs:string" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="TypesList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="xs:string" form="unqualified" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="subscribeResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:schema> |

## ManageCountriesService

*http://<host>:<port/CountryBaseRegistryWS/CountryBaseRegistryManageCountriesService?wsdl*

***CountryBaseRegistryManageCountriesService?wsdl:***

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://ccbr.emsa.europa.eu/" name="CountryBaseRegistryManageCountriesService">  <types>  <xsd:schema>  <xsd:import namespace="http://ccbr.emsa.europa.eu/" schemaLocation="http://172.16.120.206:7033/ShipDB-CountryBaseRegistryWS/CountryBaseRegistryManageCountriesService?xsd=1"/>  </xsd:schema>  </types>  <message name="insertCountry">  <part name="parameters" element="tns:insertCountry"/>  </message>  <message name="insertCountryResponse">  <part name="parameters" element="tns:insertCountryResponse"/>  </message>  <message name="updateCountry">  <part name="parameters" element="tns:updateCountry"/>  </message>  <message name="updateCountryResponse">  <part name="parameters" element="tns:updateCountryResponse"/>  </message>  <message name="deleteCountry">  <part name="parameters" element="tns:deleteCountry"/>  </message>  <message name="deleteCountryResponse">  <part name="parameters" element="tns:deleteCountryResponse"/>  </message>  <message name="insertMID">  <part name="parameters" element="tns:insertMID"/>  </message>  <message name="insertMIDResponse">  <part name="parameters" element="tns:insertMIDResponse"/>  </message>  <message name="updateMID">  <part name="parameters" element="tns:updateMID"/>  </message>  <message name="updateMIDResponse">  <part name="parameters" element="tns:updateMIDResponse"/>  </message>  <message name="deleteMID">  <part name="parameters" element="tns:deleteMID"/>  </message>  <message name="deleteMIDResponse">  <part name="parameters" element="tns:deleteMIDResponse"/>  </message>  <message name="insertCallsign">  <part name="parameters" element="tns:insertCallsign"/>  </message>  <message name="insertCallsignResponse">  <part name="parameters" element="tns:insertCallsignResponse"/>  </message>  <message name="updateCallsign">  <part name="parameters" element="tns:updateCallsign"/>  </message>  <message name="updateCallsignResponse">  <part name="parameters" element="tns:updateCallsignResponse"/>  </message>  <message name="deleteCallsign">  <part name="parameters" element="tns:deleteCallsign"/>  </message>  <message name="deleteCallsignResponse">  <part name="parameters" element="tns:deleteCallsignResponse"/>  </message>  <portType name="CountryBaseRegistryManageCountries">  <operation name="insertCountry">  <input message="tns:insertCountry"/>  <output message="tns:insertCountryResponse"/>  </operation>  <operation name="updateCountry">  <input message="tns:updateCountry"/>  <output message="tns:updateCountryResponse"/>  </operation>  <operation name="deleteCountry">  <input message="tns:deleteCountry"/>  <output message="tns:deleteCountryResponse"/>  </operation>  <operation name="insertMID">  <input message="tns:insertMID"/>  <output message="tns:insertMIDResponse"/>  </operation>  <operation name="updateMID">  <input message="tns:updateMID"/>  <output message="tns:updateMIDResponse"/>  </operation>  <operation name="deleteMID">  <input message="tns:deleteMID"/>  <output message="tns:deleteMIDResponse"/>  </operation>  <operation name="insertCallsign">  <input message="tns:insertCallsign"/>  <output message="tns:insertCallsignResponse"/>  </operation>  <operation name="updateCallsign">  <input message="tns:updateCallsign"/>  <output message="tns:updateCallsignResponse"/>  </operation>  <operation name="deleteCallsign">  <input message="tns:deleteCallsign"/>  <output message="tns:deleteCallsignResponse"/>  </operation>  </portType>  <binding name="CountryBaseRegistryManageCountriesPortBinding" type="tns:CountryBaseRegistryManageCountries">  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>  <operation name="insertCountry">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="updateCountry">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="deleteCountry">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="insertMID">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="updateMID">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="deleteMID">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="insertCallsign">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="updateCallsign">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="deleteCallsign">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  </binding>  <service name="CountryBaseRegistryManageCountriesService">  <port name="CountryBaseRegistryManageCountriesPort" binding="tns:CountryBaseRegistryManageCountriesPortBinding">  <soap:address location="http://172.16.120.206:7033/ShipDB-CountryBaseRegistryWS/CountryBaseRegistryManageCountriesService"/>  </port>  </service>  </definitions> |

**CountryBaseRegistryManageCountriesService?xsd=1:**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <xs:schema xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ccbr.emsa.europa.eu/" elementFormDefault="qualified" version="1.0">  <xs:element name="Alpha2Code">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="Callsign">  <xs:complexType>  <xs:sequence>  <xs:element name="ID" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CallsignID">  <xs:complexType>  <xs:sequence>  <xs:element name="CallsignID" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="Country">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="Alpha3Code" type="xs:string" form="unqualified"/>  <xs:element name="NumericCode" type="xs:long" form="unqualified"/>  <xs:element name="CountryName" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2CodeAlias" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="ParentCountry" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="LRIT\_ID" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="CategoryId" type="xs:int" form="unqualified"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  <xs:element name="FlagImageSmall" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="FlagImageMedium" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="FlagImageLarge" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="CountryTypesList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" form="unqualified" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryTypeId" type="xs:int" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="RegionalAgreementsList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" form="unqualified" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreementId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="MID">  <xs:complexType>  <xs:sequence>  <xs:element name="ID" type="xs:int" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="MidID">  <xs:complexType>  <xs:sequence>  <xs:element name="MidID" type="xs:int" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="Receipt">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="UserID">  <xs:complexType>  <xs:sequence>  <xs:element name="UserID" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="deleteCallsign" type="tns:deleteCallsign"/>  <xs:element name="deleteCallsignResponse" type="tns:deleteCallsignResponse"/>  <xs:element name="deleteCountry" type="tns:deleteCountry"/>  <xs:element name="deleteCountryResponse" type="tns:deleteCountryResponse"/>  <xs:element name="deleteMID" type="tns:deleteMID"/>  <xs:element name="deleteMIDResponse" type="tns:deleteMIDResponse"/>  <xs:element name="insertCallsign" type="tns:insertCallsign"/>  <xs:element name="insertCallsignResponse" type="tns:insertCallsignResponse"/>  <xs:element name="insertCountry" type="tns:insertCountry"/>  <xs:element name="insertCountryResponse" type="tns:insertCountryResponse"/>  <xs:element name="insertMID" type="tns:insertMID"/>  <xs:element name="insertMIDResponse" type="tns:insertMIDResponse"/>  <xs:element name="updateCallsign" type="tns:updateCallsign"/>  <xs:element name="updateCallsignResponse" type="tns:updateCallsignResponse"/>  <xs:element name="updateCountry" type="tns:updateCountry"/>  <xs:element name="updateCountryResponse" type="tns:updateCountryResponse"/>  <xs:element name="updateMID" type="tns:updateMID"/>  <xs:element name="updateMIDResponse" type="tns:updateMIDResponse"/>  <xs:complexType name="deleteCountry">  <xs:sequence>  <xs:element name="alpha2Code" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="userID" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="UserID" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="deleteCountryResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="deleteCallsign">  <xs:sequence>  <xs:element name="callsignId" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CallsignID" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="userID" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="UserID" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="deleteCallsignResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateMID">  <xs:sequence>  <xs:element name="mid" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ID" type="xs:int" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateMIDResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="insertCountry">  <xs:sequence>  <xs:element name="country" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="Alpha3Code" type="xs:string" form="unqualified"/>  <xs:element name="NumericCode" type="xs:long" form="unqualified"/>  <xs:element name="CountryName" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2CodeAlias" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="ParentCountry" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="LRIT\_ID" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="CategoryId" type="xs:int" form="unqualified"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  <xs:element name="FlagImageSmall" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="FlagImageMedium" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="FlagImageLarge" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="CountryTypesList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" form="unqualified" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryTypeId" type="xs:int" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="RegionalAgreementsList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" form="unqualified" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreementId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="insertCountryResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="deleteMID">  <xs:sequence>  <xs:element name="midId" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="MidID" type="xs:int" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="userID" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="UserID" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="deleteMIDResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="insertCallsign">  <xs:sequence>  <xs:element name="callsign" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ID" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="insertCallsignResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateCallsign">  <xs:sequence>  <xs:element name="callsign" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ID" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateCallsignResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="insertMID">  <xs:sequence>  <xs:element name="mid" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ID" type="xs:int" form="unqualified"/>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" form="unqualified" minOccurs="0"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="insertMIDResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateCountry">  <xs:sequence>  <xs:element name="country" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string" form="unqualified"/>  <xs:element name="Alpha3Code" type="xs:string" form="unqualified"/>  <xs:element name="NumericCode" type="xs:long" form="unqualified"/>  <xs:element name="CountryName" type="xs:string" form="unqualified"/>  <xs:element name="Alpha2CodeAlias" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="ParentCountry" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="LRIT\_ID" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="CategoryId" type="xs:int" form="unqualified"/>  <xs:element name="UserId" type="xs:string" form="unqualified"/>  <xs:element name="FlagImageSmall" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="FlagImageMedium" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="FlagImageLarge" type="xs:string" form="unqualified" minOccurs="0"/>  <xs:element name="CountryTypesList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" form="unqualified" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryTypeId" type="xs:int" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="RegionalAgreementsList" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" form="unqualified" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreementId" type="xs:string" form="unqualified"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="updateCountryResponse">  <xs:sequence>  <xs:element name="return" form="unqualified" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="ResponseMsg" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:schema> |

## CountryInformationService

*http://<host>:<port/CountryBaseRegistryWS/CountryBaseRegistryInformationService?wsdl*

***CountryBaseRegistryInformationService?wsdl:***

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://ccbr.emsa.europa.eu/" name="CountryBaseRegistryInformationService">  <types>  <xsd:schema>  <xsd:import namespace="http://ccbr.emsa.europa.eu/" schemaLocation="http://172.16.120.206:7033/ShipDB-CountryBaseRegistryWS/CountryBaseRegistryInformationService?xsd=1"/>  </xsd:schema>  </types>  <message name="getCountryList">  <part name="parameters" element="tns:getCountryList"/>  </message>  <message name="getCountryListResponse">  <part name="parameters" element="tns:getCountryListResponse"/>  </message>  <message name="getCountryFullList">  <part name="parameters" element="tns:getCountryFullList"/>  </message>  <message name="getCountryFullListResponse">  <part name="parameters" element="tns:getCountryFullListResponse"/>  </message>  <message name="getSubscribedCountryList">  <part name="parameters" element="tns:getSubscribedCountryList"/>  </message>  <message name="getSubscribedCountryListResponse">  <part name="parameters" element="tns:getSubscribedCountryListResponse"/>  </message>  <message name="getSubscribedCountryTypes">  <part name="parameters" element="tns:getSubscribedCountryTypes"/>  </message>  <message name="getSubscribedCountryTypesResponse">  <part name="parameters" element="tns:getSubscribedCountryTypesResponse"/>  </message>  <message name="getSubscribedRegionalAgreements">  <part name="parameters" element="tns:getSubscribedRegionalAgreements"/>  </message>  <message name="getSubscribedRegionalAgreementsResponse">  <part name="parameters" element="tns:getSubscribedRegionalAgreementsResponse"/>  </message>  <message name="getSubscribedCountries">  <part name="parameters" element="tns:getSubscribedCountries"/>  </message>  <message name="getSubscribedCountriesResponse">  <part name="parameters" element="tns:getSubscribedCountriesResponse"/>  </message>  <message name="getCountry">  <part name="parameters" element="tns:getCountry"/>  </message>  <message name="getCountryResponse">  <part name="parameters" element="tns:getCountryResponse"/>  </message>  <message name="getParentCountry">  <part name="parameters" element="tns:getParentCountry"/>  </message>  <message name="getParentCountryResponse">  <part name="parameters" element="tns:getParentCountryResponse"/>  </message>  <message name="getChildCountryList">  <part name="parameters" element="tns:getChildCountryList"/>  </message>  <message name="getChildCountryListResponse">  <part name="parameters" element="tns:getChildCountryListResponse"/>  </message>  <message name="getCountryHistory">  <part name="parameters" element="tns:getCountryHistory"/>  </message>  <message name="getCountryHistoryResponse">  <part name="parameters" element="tns:getCountryHistoryResponse"/>  </message>  <message name="getCountryMIDList">  <part name="parameters" element="tns:getCountryMIDList"/>  </message>  <message name="getCountryMIDListResponse">  <part name="parameters" element="tns:getCountryMIDListResponse"/>  </message>  <message name="getCountryCallsignList">  <part name="parameters" element="tns:getCountryCallsignList"/>  </message>  <message name="getCountryCallsignListResponse">  <part name="parameters" element="tns:getCountryCallsignListResponse"/>  </message>  <message name="getCountryFlagImageList">  <part name="parameters" element="tns:getCountryFlagImageList"/>  </message>  <message name="getCountryFlagImageListResponse">  <part name="parameters" element="tns:getCountryFlagImageListResponse"/>  </message>  <message name="getCountryFlagImages">  <part name="parameters" element="tns:getCountryFlagImages"/>  </message>  <message name="getCountryFlagImagesResponse">  <part name="parameters" element="tns:getCountryFlagImagesResponse"/>  </message>  <message name="getCountryFlagImage">  <part name="parameters" element="tns:getCountryFlagImage"/>  </message>  <message name="getCountryFlagImageResponse">  <part name="parameters" element="tns:getCountryFlagImageResponse"/>  </message>  <message name="getCountryFlagImageLocation">  <part name="parameters" element="tns:getCountryFlagImageLocation"/>  </message>  <message name="getCountryFlagImageLocationResponse">  <part name="parameters" element="tns:getCountryFlagImageLocationResponse"/>  </message>  <message name="getCountryFlagImageLocationContext">  <part name="parameters" element="tns:getCountryFlagImageLocationContext"/>  </message>  <message name="getCountryFlagImageLocationContextResponse">  <part name="parameters" element="tns:getCountryFlagImageLocationContextResponse"/>  </message>  <message name="getCountryTypes">  <part name="parameters" element="tns:getCountryTypes"/>  </message>  <message name="getCountryTypesResponse">  <part name="parameters" element="tns:getCountryTypesResponse"/>  </message>  <message name="getCountryCategories">  <part name="parameters" element="tns:getCountryCategories"/>  </message>  <message name="getCountryCategoriesResponse">  <part name="parameters" element="tns:getCountryCategoriesResponse"/>  </message>  <portType name="CountryBaseRegistryInformation">  <operation name="getCountryList">  <input message="tns:getCountryList"/>  <output message="tns:getCountryListResponse"/>  </operation>  <operation name="getCountryFullList">  <input message="tns:getCountryFullList"/>  <output message="tns:getCountryFullListResponse"/>  </operation>  <operation name="getSubscribedCountryList">  <input message="tns:getSubscribedCountryList"/>  <output message="tns:getSubscribedCountryListResponse"/>  </operation>  <operation name="getSubscribedCountryTypes">  <input message="tns:getSubscribedCountryTypes"/>  <output message="tns:getSubscribedCountryTypesResponse"/>  </operation>  <operation name="getSubscribedRegionalAgreements">  <input message="tns:getSubscribedRegionalAgreements"/>  <output message="tns:getSubscribedRegionalAgreementsResponse"/>  </operation>  <operation name="getSubscribedCountries">  <input message="tns:getSubscribedCountries"/>  <output message="tns:getSubscribedCountriesResponse"/>  </operation>  <operation name="getCountry">  <input message="tns:getCountry"/>  <output message="tns:getCountryResponse"/>  </operation>  <operation name="getParentCountry">  <input message="tns:getParentCountry"/>  <output message="tns:getParentCountryResponse"/>  </operation>  <operation name="getChildCountryList">  <input message="tns:getChildCountryList"/>  <output message="tns:getChildCountryListResponse"/>  </operation>  <operation name="getCountryHistory">  <input message="tns:getCountryHistory"/>  <output message="tns:getCountryHistoryResponse"/>  </operation>  <operation name="getCountryMIDList">  <input message="tns:getCountryMIDList"/>  <output message="tns:getCountryMIDListResponse"/>  </operation>  <operation name="getCountryCallsignList">  <input message="tns:getCountryCallsignList"/>  <output message="tns:getCountryCallsignListResponse"/>  </operation>  <operation name="getCountryFlagImageList">  <input message="tns:getCountryFlagImageList"/>  <output message="tns:getCountryFlagImageListResponse"/>  </operation>  <operation name="getCountryFlagImages">  <input message="tns:getCountryFlagImages"/>  <output message="tns:getCountryFlagImagesResponse"/>  </operation>  <operation name="getCountryFlagImage">  <input message="tns:getCountryFlagImage"/>  <output message="tns:getCountryFlagImageResponse"/>  </operation>  <operation name="getCountryFlagImageLocation">  <input message="tns:getCountryFlagImageLocation"/>  <output message="tns:getCountryFlagImageLocationResponse"/>  </operation>  <operation name="getCountryFlagImageLocationContext">  <input message="tns:getCountryFlagImageLocationContext"/>  <output message="tns:getCountryFlagImageLocationContextResponse"/>  </operation>  <operation name="getCountryTypes">  <input message="tns:getCountryTypes"/>  <output message="tns:getCountryTypesResponse"/>  </operation>  <operation name="getCountryCategories">  <input message="tns:getCountryCategories"/>  <output message="tns:getCountryCategoriesResponse"/>  </operation>  </portType>  <binding name="CountryBaseRegistryInformationPortBinding" type="tns:CountryBaseRegistryInformation">  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>  <operation name="getCountryList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryFullList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getSubscribedCountryList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getSubscribedCountryTypes">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getSubscribedRegionalAgreements">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getSubscribedCountries">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountry">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getParentCountry">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getChildCountryList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryHistory">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryMIDList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryCallsignList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryFlagImageList">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryFlagImages">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryFlagImage">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryFlagImageLocation">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryFlagImageLocationContext">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryTypes">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  <operation name="getCountryCategories">  <soap:operation soapAction=""/>  <input>  <soap:body use="literal"/>  </input>  <output>  <soap:body use="literal"/>  </output>  </operation>  </binding>  <service name="CountryBaseRegistryInformationService">  <port name="CountryBaseRegistryInformationPort" binding="tns:CountryBaseRegistryInformationPortBinding">  <soap:address location="http://172.16.120.206:7033/ShipDB-CountryBaseRegistryWS/CountryBaseRegistryInformationService"/>  </port>  </service>  </definitions> |

**CountryBaseRegistryInformationService?xsd=1:**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <xs:schema xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ccbr.emsa.europa.eu/" version="1.0">  <xs:element name="Alpha2Code">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="AppAcronym">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryCallSigns">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CallSignList">  <xs:complexType>  <xs:sequence>  <xs:element name="CallSign" type="tns:CallSignDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryCategories">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryCategoriesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryCategory" type="tns:CountryCategoryType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryDetailedSearchCriteria">  <xs:complexType>  <xs:sequence>  <xs:element name="CategoryList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CategoryCode" type="xs:string" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="TypesList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="xs:string" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="RegionalAgreementsList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" type="xs:string" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryDetails">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryFlagImageURL">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="FlagImageURL" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryImages">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryImagesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryImage" type="tns:CountryImageDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryMIDs">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="MIDList">  <xs:complexType>  <xs:sequence>  <xs:element name="MID" type="tns:MIDDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountrySearchCriteria">  <xs:complexType>  <xs:sequence>  <xs:element name="NumericCode" type="xs:long" minOccurs="0"/>  <xs:element name="Alpha2Code" type="xs:string" minOccurs="0"/>  <xs:element name="Alpha3Code" type="xs:string" minOccurs="0"/>  <xs:element name="Callsign" type="xs:string" minOccurs="0"/>  <xs:element name="MID" type="xs:int" minOccurs="0"/>  <xs:element name="LRIT\_ID" type="xs:string" minOccurs="0"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryTypes">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryTypesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="tns:CountryTypesType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="HistoricSearchCriteria">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  <xs:element name="StartDate" type="xs:anySimpleType"/>  <xs:element name="EndDate" type="xs:anySimpleType"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="ImageType">  <xs:complexType>  <xs:sequence>  <xs:element name="ImageType" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="getChildCountryList" type="tns:getChildCountryList"/>  <xs:element name="getChildCountryListResponse" type="tns:getChildCountryListResponse"/>  <xs:element name="getCountry" type="tns:getCountry"/>  <xs:element name="getCountryCallsignList" type="tns:getCountryCallsignList"/>  <xs:element name="getCountryCallsignListResponse" type="tns:getCountryCallsignListResponse"/>  <xs:element name="getCountryCategories" type="tns:getCountryCategories"/>  <xs:element name="getCountryCategoriesResponse" type="tns:getCountryCategoriesResponse"/>  <xs:element name="getCountryFlagImage" type="tns:getCountryFlagImage"/>  <xs:element name="getCountryFlagImageList" type="tns:getCountryFlagImageList"/>  <xs:element name="getCountryFlagImageListResponse" type="tns:getCountryFlagImageListResponse"/>  <xs:element name="getCountryFlagImageLocation" type="tns:getCountryFlagImageLocation"/>  <xs:element name="getCountryFlagImageLocationContext" type="tns:getCountryFlagImageLocationContext"/>  <xs:element name="getCountryFlagImageLocationContextResponse" type="tns:getCountryFlagImageLocationContextResponse"/>  <xs:element name="getCountryFlagImageLocationResponse" type="tns:getCountryFlagImageLocationResponse"/>  <xs:element name="getCountryFlagImageResponse" type="tns:getCountryFlagImageResponse"/>  <xs:element name="getCountryFlagImages" type="tns:getCountryFlagImages"/>  <xs:element name="getCountryFlagImagesResponse" type="tns:getCountryFlagImagesResponse"/>  <xs:element name="getCountryFullList" type="tns:getCountryFullList"/>  <xs:element name="getCountryFullListResponse" type="tns:getCountryFullListResponse"/>  <xs:element name="getCountryHistory" type="tns:getCountryHistory"/>  <xs:element name="getCountryHistoryResponse" type="tns:getCountryHistoryResponse"/>  <xs:element name="getCountryList" type="tns:getCountryList"/>  <xs:element name="getCountryListResponse" type="tns:getCountryListResponse"/>  <xs:element name="getCountryMIDList" type="tns:getCountryMIDList"/>  <xs:element name="getCountryMIDListResponse" type="tns:getCountryMIDListResponse"/>  <xs:element name="getCountryResponse" type="tns:getCountryResponse"/>  <xs:element name="getCountryTypes" type="tns:getCountryTypes"/>  <xs:element name="getCountryTypesResponse" type="tns:getCountryTypesResponse"/>  <xs:element name="getParentCountry" type="tns:getParentCountry"/>  <xs:element name="getParentCountryResponse" type="tns:getParentCountryResponse"/>  <xs:element name="getSubscribedCountries" type="tns:getSubscribedCountries"/>  <xs:element name="getSubscribedCountriesResponse" type="tns:getSubscribedCountriesResponse"/>  <xs:element name="getSubscribedCountryList" type="tns:getSubscribedCountryList"/>  <xs:element name="getSubscribedCountryListResponse" type="tns:getSubscribedCountryListResponse"/>  <xs:element name="getSubscribedCountryTypes" type="tns:getSubscribedCountryTypes"/>  <xs:element name="getSubscribedCountryTypesResponse" type="tns:getSubscribedCountryTypesResponse"/>  <xs:element name="getSubscribedRegionalAgreements" type="tns:getSubscribedRegionalAgreements"/>  <xs:element name="getSubscribedRegionalAgreementsResponse" type="tns:getSubscribedRegionalAgreementsResponse"/>  <xs:complexType name="getSubscribedRegionalAgreements">  <xs:sequence>  <xs:element name="appAcronym" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedRegionalAgreementsResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CountryDetailsType">  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  <xs:element name="Alpha3Code" type="xs:string"/>  <xs:element name="NumericCode" type="xs:long"/>  <xs:element name="CountryName" type="xs:string"/>  <xs:element name="Alpha2CodeAlias" type="xs:string" minOccurs="0"/>  <xs:element name="ParentCountry" type="xs:string" minOccurs="0"/>  <xs:element name="LRIT\_ID" type="xs:string" minOccurs="0"/>  <xs:element name="CreationTimestamp" type="xs:anySimpleType" minOccurs="0"/>  <xs:element name="UpdateTimestamp" type="xs:anySimpleType" minOccurs="0"/>  <xs:element name="CategoryType">  <xs:complexType>  <xs:sequence>  <xs:element name="CategoryId" type="xs:int"/>  <xs:element name="CategoryCode" type="xs:string"/>  <xs:element name="CategoryDesc" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="CountryTypesList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryTypeId" type="xs:int"/>  <xs:element name="CountryTypeDesc" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="RegionalAgreementsList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" maxOccurs="unbounded">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreementCode" type="xs:string"/>  <xs:element name="RegionalAgreementDesc" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountry">  <xs:sequence>  <xs:element name="searchMsg" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="NumericCode" type="xs:long" minOccurs="0"/>  <xs:element name="Alpha2Code" type="xs:string" minOccurs="0"/>  <xs:element name="Alpha3Code" type="xs:string" minOccurs="0"/>  <xs:element name="Callsign" type="xs:string" minOccurs="0"/>  <xs:element name="MID" type="xs:int" minOccurs="0"/>  <xs:element name="LRIT\_ID" type="xs:string" minOccurs="0"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImage">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="imageType" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ImageType" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImageResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryImagesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryImage" type="tns:CountryImageDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CountryImageDetailsType">  <xs:sequence>  <xs:element name="Country" type="xs:string"/>  <xs:element name="Image" type="tns:ImageDetailsType"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ImageDetailsType">  <xs:sequence>  <xs:element name="Small" type="xs:string" minOccurs="0"/>  <xs:element name="Medium" type="xs:string" minOccurs="0"/>  <xs:element name="Large" type="xs:string" minOccurs="0"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryCallsignList">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryCallsignListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CallSignList">  <xs:complexType>  <xs:sequence>  <xs:element name="CallSign" type="tns:CallSignDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CallSignDetailsType">  <xs:sequence>  <xs:element name="ID" type="xs:string"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedCountryList">  <xs:sequence>  <xs:element name="appAcronym" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedCountryListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImageLocationContext">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getCountryFlagImageLocationContextResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="FlagImageURL" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getChildCountryList">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getChildCountryListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImageList">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getCountryFlagImageListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryImagesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryImage" type="tns:CountryImageDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryTypes">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getCountryTypesResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryTypesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="tns:CountryTypesType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CountryTypesType">  <xs:sequence>  <xs:element name="CountryTypeId" type="xs:int"/>  <xs:element name="CountryTypeCode" type="xs:string"/>  <xs:element name="CountryTypeDesc" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImages">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImagesResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryImagesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryImage" type="tns:CountryImageDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryHistory">  <xs:sequence>  <xs:element name="searchMsg" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  <xs:element name="StartDate" type="xs:anySimpleType"/>  <xs:element name="EndDate" type="xs:anySimpleType"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryHistoryResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedCountryTypes">  <xs:sequence>  <xs:element name="appAcronym" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedCountryTypesResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryTypesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="tns:CountryTypesType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedCountries">  <xs:sequence>  <xs:element name="appAcronym" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="AppAcronym" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSubscribedCountriesResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryMIDList">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryMIDListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="MIDList">  <xs:complexType>  <xs:sequence>  <xs:element name="MID" type="tns:MIDDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="MIDDetailsType">  <xs:sequence>  <xs:element name="ID" type="xs:int"/>  <xs:element name="RecordTimestamp" type="xs:anySimpleType"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFullList">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getCountryFullListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImageLocation">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="imageType" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ImageType" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryFlagImageLocationResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="FlagImageURL" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryCategories">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getCountryCategoriesResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryCategoriesList">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryCategory" type="tns:CountryCategoryType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CountryCategoryType">  <xs:sequence>  <xs:element name="CountryCategoryId" type="xs:int"/>  <xs:element name="CountryCategoryCode" type="xs:string"/>  <xs:element name="CountryCategoryDesc" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getParentCountry">  <xs:sequence>  <xs:element name="alpha2Code" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="Alpha2Code" type="xs:string"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getParentCountryResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryList">  <xs:sequence>  <xs:element name="searchMsg" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CategoryList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CategoryCode" type="xs:string" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="TypesList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="CountryType" type="xs:string" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  <xs:element name="RegionalAgreementsList" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="RegionalAgreement" type="xs:string" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCountryListResponse">  <xs:sequence>  <xs:element name="return" minOccurs="0">  <xs:complexType>  <xs:sequence>  <xs:element name="ResponseType" type="xs:int"/>  <xs:element name="CountryList">  <xs:complexType>  <xs:sequence>  <xs:element name="Country" type="tns:CountryDetailsType" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:sequence>  </xs:complexType>  </xs:schema> |

## NotificationService

*http://<host>:<port/CountryBaseRegistryWS/* *CountryBaseRegistryAnnouncementService?wsdl*

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Oracle JAX-WS 2.1.5. -->  -<definitions name="CountryBaseRegistryAnnouncementService" targetNamespace="http://ccbr.emsa.europa.eu/" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://ccbr.emsa.europa.eu/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">-<types>-<xsd:schema>  <xsd:import schemaLocation="http://172.16.120.90:7001/OSB-CountryBaseRegistryWS/CountryBaseRegistryAnnouncementService?xsd=1" namespace="http://ccbr.emsa.europa.eu/"/>  </xsd:schema>  </types>-<message name="notify">  <part name="parameters" element="tns:notify"/>  </message>-<message name="notifyResponse">  <part name="parameters" element="tns:notifyResponse"/>  </message>-<portType name="CountryBaseRegistryAnnouncement">-<operation name="notify">  <input message="tns:notify"/>  <output message="tns:notifyResponse"/>  </operation>  </portType>-<binding name="CountryBaseRegistryAnnouncementPortBinding" type="tns:CountryBaseRegistryAnnouncement">  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>-<operation name="notify">  <soap:operation soapAction=""/>-<input>  <soap:body use="literal"/>  </input>-<output>  <soap:body use="literal"/>  </output>  </operation>  </binding>-<service name="CountryBaseRegistryAnnouncementService">-<port name="CountryBaseRegistryAnnouncementPort" binding="tns:CountryBaseRegistryAnnouncementPortBinding">  <soap:address location="http://172.16.120.90:7001/OSB-CountryBaseRegistryWS/CountryBaseRegistryAnnouncementService"/>  </port>  </service>  </definitions> |

# Annex II

This decoding table provides the user of ISO 3166-1 with an easy access to the definition of all 676 code elements available in the alpha-2 code of ISO's country code standard.

The content of this document is taken from two sources: The official code list of ISO 3166-1 and the list of reserved ISO 3166-1 code elements which contains information on code allocations which are not officially part of ISO 3166-1 (see "Reserved code elements"). The reserved code elements have different statuses with regard to restrictions on use within ISO's country code system. Some may be used - others must not be used.

Source: <http://www.iso.org/iso/iso-3166-1_decoding_table>

