EASTIN Web services specs

In EASTIN Web services there are two groups of Web methods: the **batch methods** and the **live methods**. **Batch methods** are invoked by automatic processes which run in EASTIN central server and are used to update some almost-static information inside the EASTIN Portal (for example the ISO tree, the keyword lists, etc.). These methods are called with different frequencies (from once a day to once a month), depending on how often the retrieved information content is supposed to change inside each EASTIN partner's local system. For example, the method which returns the keywords related to the ISO Classification is called once a month, because the ISO classification and the related keywords are supposed to be almost constant. Instead, the method which returns the number of products for a given ISO class is invoked once a day because new products could be often added inside the EASTIN partners' local systems or perhaps their description could have been modified.

On the other side **live methods** are invoked directly by the end users through the EASTIN Portal Web pages and they return the results of searches inside EASTIN partners' databases about products, actors (also called "companies" in the EASTIN Web pages), and associated information.

Batch methods

IsoClassLocalizationDto[] GetIsoClasses()

Input parameters:

none.

Returns:

 IsoClassLocalizationDto[]: an array of IsoClassLocalizationDto objects containing information about the ISO classes. If no ISO class is found returns a not null IsoClassLocalizationDto[] array with zero elements.

Frequency:

- Once a day; the unique call to the Web method is executed at 04.00: AM (UTC).

This is a batch method which returns an array of objects belonging to the class IsoClassLocalizationDto, which represents an element of the ISO classification. The method searches into the local database for information about all the ISO classes. The information items retrieved by the method and stored in the IsoClassLocalizationDto object are:

- the ISO code;
- the title of the ISO class in the local database language;
- the scope notes of the ISO class in the local database language (if it exists);
- the number of assistive products present in the local database related to this ISO class.

The retrieves information about ISO classes are updated in the EASTIN Portal centralized ISO tree. For a complete description of the IsoClassLocalizationDto object see below.

KeywordDto[] GetKeywords()

Input parameters:

none.

Returns:

KeywordDto[]: an array of KeywordDto objects containing information about keywords. If no keyword is found returns a not null KeywordDto[] array with zero elements.

Frequency:

Once a month; the unique call to the Web method is executed on the third day of every month at 05.00 AM (UTC).

This is a batch method which returns an array of objects belonging to the class KeywordDto. The method searches into the EASTIN partners' local databases for the dictionary of (keywords -> ISO classes) which will be used in the keyword research of the EASTIN portal. This method requires no parameter. Each KeywordDto object contains the following information:

- the keyword id in the partner's local database;
- the keyword text;
- an array of ISO codes which are related to this keyword.

If no keyword is found the method returns a not null array with zero elements. The method is designed to work in batch mode. Once a month the method is called and the returned information are updated in the EASTIN portal keyword lists. For a complete description of the KeywordDto object see below.

Live search methods

1. Product searches

SmallProductDto[] FindSmallProducts(string[] isoCodes, FeatureDto[] features, string commercialName, string manufacturer, dateTime insertDateMin, dateTime insertDateMax)

Input parameters:

- string[] isoCodes: an array of strings representing ISO classes (for example ["12.22", "09.03.03"]);
- FeatureDto[] features: an array of FeatureDto objects (for a complete description of the FeatureDto object see "EASTIN custom data type" below);
- string commercialName: the whole or a part of the commercial name of the products to be searched;
- string **manufacturer**: the whole or a part of the manufacturer name of the products to be searched;
- dateTime **insertDateMin**: the lower bound for the insert date of the products to be searched;
- dateTime insertDateMax: the upper bound for the insert date of the products to be searched.

Returns:

 SmallProductDto[]: array of SmallProductDto objects containing each a light set of information about a product (for a complete description of the SmallProductDto object see below). If no product is found returns a not null SmallProductDto[] array with zero elements.

This method returns an array of objects belonging to the class SmallProductDto. The method implements five different kinds of searches:

- 1. If the **isoCodes** array is not void the method searches for all products belonging to the ISO classes passed, using an *OR* statement. For example if ["12.22", "09.03.03"] is the isoCodes array, all products belonging to the 12.22 ISO class *OR* to the 09.03.03 class are returned.
- 2. If the features array is not void the method searches for all products that possess the indicated FeatureDto objects and whose measures for the respective features are compatible with the measure boundaries specified in the FeatureDto objects. For example if features contains the FeatureDtos [{"Width (cm)", 30, 50}, {"Height (cm)", 80, 100}] the method will search for all products having some widths in the range [30, 50] AND having some heights in the range [80, 100]. Note that if a product declares for example to have a fixable width between 20 and 40 it should be included in the search results since for some of its configurations it satisfies the boundaries. The product is included in the search results only if the compatibility between its measures and the given boundaries present in in the

FeatureDto objects are satisfied for all FeatureDto objects. For a complete list of Features see the paragraph "EASTIN feature vocabulary" below.

- 3. If **commercialName** is not void the method searches a matching between the words contained in the commercialName parameter and the respective data in the EASTIN partner's local database. Since into the commercialName parameter there could be one or more words, the method must split the words and search inside its database for products whose commercial name contains *all* these words (even if present as substrings inside of biggest strings). For example if commercialName = "quickie xenon" the method must search for all products whose commercial name contains both words "quickie" *AND* "xenon".
- 4. If **manufacturer** is not void the method executes the search using the same criteria specified in 2 but applied to products' manufacturer name.
- 5. If **insertDateMin** and **insertDateMax** are both not null all products whose insert date is included within the interval [insertDateMin, insertDateMax], endpoints included, are returned. These two parameters must be both not null or both null.

If more than one parameter is not void at the same time, the results coming from the matches for each parameter are merged together with an *AND* logic: only results satisfying the conditions specified for each parameter are returned. If no product is found the method returns a not null SmallProductDto[] array with zero elements.

ProductDto GetProduct(string productCode)

Input parameters:

- string **productCode**: the id of the product in the EASTIN partner's system.

Returns:

 ProductDto: an object containing detailed information about a single product. If no product is found than returns the null object.

This method returns an object belonging to the class ProductDto (for a complete description of the ProductDto object see below). The method searches into EASTIN partner's local databases for the product which has the id matching with the method parameter productCode. If no product is found the method returns the null object.

2. Actor searches

SmallActorDto[] FindSmallActors(string actorType, string[] isoCodes, string[] icfCodes, string actorName, dateTime insertDateMin, dateTime insertDateMax)

Input parameters:

- string actorType: the type of the actor;
- string[] isoCodes: an array of strings representing ISO classes (for example ["12.22", "09.03.03"]);
- string[] icfCodes: an array of strings representing the EASTIN ICF classes (for example ["b1", "d2"]) which are a subset of the official ICF classification;
- string actorName: the whole or a part of the name of the searched actor;
- dateTime insertDateMin: the lower bound for the insert date of the actors to be searched;
- dateTime insertDateMax: the upper bound for the insert date of the actors to be searched.

Returns:

 SmallActorDto[]: an array of SmallActorDto objects containing each a light set of information about an actor (for a complete description of the SmallActorDto object see below). If no actor is found returns a not null SmallActorDto[] array with zero elements. This method returns an array of objects belonging to the class SmallActorDto. The method implements five different kinds of searches:

- 1. If the **type** parameter is not void the method searches for all actors belonging to the specified type; the possible values for type are: "companies", "projects" and "serviceproviders".
- 2. If the **isoCodes** array is not void the method searches for all actors belonging to the ISO classes passed, using an *OR* statement. For example if ["12.22", "09.03.03"] is the isoCodes array, all actors belonging to the 12.22 ISO class *OR* to the 09.03.03 class are returned.
- 3. If the **icfCodes** array is not void the method searches for all actors belonging to the ICF classes passed, using an *OR* statement. For example if ["b1", "d2"] is the icfCodes array, all actors belonging to the b1 ICF class *OR* to the d2 class are returned.
- 4. If **actorName** is not void the method searches a matching between the words contained in the actorName parameter and the respective data in the EASTIN partner's local database. Since into the actorName parameter there could be one or more words, the method must split the words and search inside its database for actors whose name contains *all* these words (even if present as substrings inside of biggest strings). For example if actorName = "metlex Itd" the method must search for all actors whose name contains both words "metlex" *AND* "Itd".
- 5. If **insertDateMin** and **insertDateMax** are both not null all actors whose insert date is included within the interval [insertDateMin, insertDateMax], endpoints included, are returned. These two parameters must be both not null or both null.

If more than one parameter is not void at the same time, the results coming from the matches for each parameter are merged together with an *AND* logic: only results satisfying the conditions specified for each parameter are returned. If no actor is found returns a not null SmallActorDto[] array with zero elements.

ActorDto **GetActor**(string actorType, string actorCode)

Input parameters:

- string actorType: the type of the actor;
- string actorCode: the id identifying a single actor inside the EASTIN partner's local system.

Returns:

 ActorDto: an object containing detailed information about a single actor (for a complete description of the ActorDto object see below). If no actor is found than returns the null object.

This method returns an object belonging to the class ActorDto. The method searches into EASTIN partner's local database for the actor of the type specified in the actorType parameter which has the id matching with the method parameter actorCode. If no actor is found the method returns the null object.

3. Associated information searches

SmallAssociatedInfoDto[] FindSmallAssociatedInfos(string infoType, string[] isoCodes, string[] icfCodes, string title, string author, dateTime insertDateMin, dateTime insertDateMax)

Input parameters:

- string infoType: the type of the associated information document;
- string[]isoCodes: an array of strings representing ISO classes (for example ["12.22", "09.03.03"]);
- string[]icfCodes: an array of strings representing EASTIN ICF classes (for example ["b1", "d2"]);
- string title: the whole or a part of the title (in the original language or in English) of the searched associated information document;
- string author: the whole or a part of the author names of the searched associated information document;

- dateTime insertDateMin: the lower bound for the insert date of the associated information documents to be searched;
- dateTime insertDateMax: the upper bound for the insert date of the associated information documents to be searched.

Returns:

 SmallAssociatedInfoDto[]: an array of SmallAssociatedInfoDto objects containing each a light set of information about an associated information document (for a complete description of the SmallAssociatedInfoDto object see below). If no associated information document is found returns a not null SmallAssociatedInfoDto[] array with zero elements.

This method returns an array of objects belonging to the class SmallAssociatedInfoDto. The method implements six different kinds of searches:

- 1. If the **type** parameter is not void the method searches for all associated information documents belonging to the specified type; the possible values are: "articles", "casedescriptions", "ideas", "faqs", "forums", "news" and "regulations".
- 2. If the **isoCodes** array is not void the method searches for all associated information documents belonging to the ISO classes passed, using an *OR* statement. For example if ["12.22", "09.03.03"] is the isoCodes array, all associated information documents belonging to the 12.22 ISO class *OR* to the 09.03.03 class are returned.
- 3. If the **icfCodes** array is not void the method searches for all associated information documents belonging to the ICF classes passed, using an *OR* statement. For example if ["b1", "d2"] is the icfCodes array, all associated information documents belonging to the b1 ICF class *OR* to the d2 class are returned.
- 4. If **title** is not void the method searches a matching between the words contained in the title parameter and the respective data in the EASTIN partner's local database. Since into the title parameter there could be one or more words, the method must split the words and search inside its database for associated information documents whose title (in original language *OR* in English if present) contains *all* these words (even if present as substrings inside of biggest strings). For example if title = "a guide to wheeled walking frames" the method must search for all associated information documents whose original title or whose English title contain all words "a", "guide", "to", "wheeled", "walking" and "frames".
- 5. If the **author** parameter is not void the method executes the search using the same criteria specified in 4 but applied to the name of the authors of the associated information document (in this case no distinction is needed between original language and English).
- 6. If **insertDateMin** and **insertDateMax** are both not null all associated information documents whose insert date is included within the interval [insertDateMin, insertDateMax], endpoints included, are returned. These two parameters must be both not null or both null.

If more than one parameter is not void at the same time, the results coming from the matches for each parameter are merged together with an *AND* logic: only results satisfying the conditions specified for each parameter are returned. If no associated information document is found the method returns a not null SmallAssociatedInfoDoc[] array with zero elements.

AssociatedInfoDto GetAssociatedInfo(string infoType, string associatedInfoCode)

Input parameters:

- string **infoType**: the type of the associated information document;
- string associatedInfoCode: the id identifying a single associated information document inside the EASTIN partner's local systems.

 AssociatedInfoDto: an object containing detailed information about a single associated information document (for a complete description of the AssociatedInfoDto object see below). If no associated information document is found than returns the null object.

The method searches into the EASTIN partner's local database for the associated information document of the type specified in the infoType parameter which has the id matching with the method parameter associatedInfoCode. If no associated information document is found the method returns the null object.

EASTIN custom data types

As we have seen EASTIN Web services return basic SOAP types, such as String, Int and DateTime, but also custom defined types. A complete description of EASTIN custom defined types follows below. All mandatory fields are marked with a "*" (all the other fields can be considered as nullable). For the array fields in case they are empty do not assign a null value to them but a not null array of zero elements.

IsoClassLocalizationDto

- string IsoCode*: the code of the ISO class;
- string Title*: the name of the ISO class;
- string ScopeNote: the ISO class description.
- Int ProductCount*: the number of assistive products related to this ISO class (zero if no product is related to the ISO class)

KeywordDto

- string KeywordId*: the id of the keyword in the partner's local database;
- string Text*: the keyword text;
- string[] IsoCodes*: the array of all ISO classification codes related to the keyword (for example ["12.22", "09.03.03"]).

FeatureDto

- integer FeatureId*: the ID of the EASTIN feature. For the complete list of EASTIN features and corresponding IDs see the paragraph "EASTIN feature vocabulary" below
- decimal ValueMin: the lower bound value of the measure specified for this feature;
- decimal ValueMax: the upper bound value of the measure specified for this feature.

SmallProductDto

- string ProductCode*: the id of the product in the partner's local database;
- string IsoCodePrimary*: the primary ISO Code of the product (for example "09.03.03");
- string[] IsoCodesOptional: the array of all secondary ISO classification codes of the product (for example ["12.22", "09.03.03"]);
- string CommercialName*: the commercial name of the product;
- string ManufacturerCode*: the id of the product's manufacturer in the partner's local database;
- string ManufacturerOriginalFullName*: the full name in the original language of the product's manufacturer;
- dateTime InsertDate*: the insert date of the product;
- dateTime LastUpdateDate*: the last update date of the product;

string ThumbnailImageUrl: the URL of the small format picture of the product (used when displaying list of products in EASTIN Portal). The URL must be accessible on the Web by the end user's browser. Picture dimensions should be: width 90 px, height 90 px.

ProductDto

- string ProductCode*: the id of the product in the partner's local database;
- string IsoCodePrimary*: the primary ISO Code of the product (for example "09.03.03");
- string[] IsoCodesOptional: the array of all secondary ISO classification codes of the product (for example ["12.22", "09.03.03"]);
- string CommercialName*: the commercial name of the product;
- string ManufacturerCode*: the id of the product's manufacturer in the partner's local database;
- string ManufacturerOriginalFullName*: the full name in the original language of the product's manufacturer;
- dateTime InsertDate*: the insert date of the product;
- dateTime LastUpdateDate*: the last update date of the product;
- string ThumbnailImageUrl: the URL of the small format image of the product (used when displaying list
 of products in the EASTIN portal). The URL must be accessible on the Web by the end user's browser.
 Picture dimensions should be: width 90 px, height 90 px.
- bool IsReviewAllowed*: if true the end user is authorized to review this product;
- string ManufacturerAddress: the address of the product's manufacturer;
- string ManufacturerPostalCode: the postal code of the product's manufacturer;
- string ManufacturerTown: the town of the product's manufacturer;
- string ManufacturerCountry*: the country code of the product's manufacturer in ISO 3166-1-alpha-2 code (for example "IT", "US", etc.);
- string ManufacturerPhone: the phone of the product's manufacturer;
- string ManufacturerFax: the fax of the product's manufacturer;
- string ManufacturerEmail: the email of the product's manufacturer;
- string ManufacturerSkype: the Skype account name of the product's manufacturer;
- string ManufacturerWebSiteUrl: the Web site URL of the product's manufacturer;
- string[] ManufacturerSocialNetworkUrls: an array of URLs linking to the product's manufacturer page inside the main social networks (for example Facebook, Twitter, LinkedIn, etc.);
- string ImageUrl: the URL of the big format image of the product (used when displaying the detail view of the product in the EASTIN portal). The URL must be accessible on the Web by the end user's browser. Picture dimensions should be: width 450 px, height 450 px.
- string OriginalDescription: the description of the product in the original language;
- string EnglishDescription: the description of the product in English;
- string OriginalUrl: the URL of the Web page in the original language on the original EASTIN partner's Web site in which the product is presented. The URL must be accessible on the Web by the end user's browser;
- string EnglishUrl: the URL of the Web page in English on the original EASTIN partner's Web site in which the product is presented. The URL must be accessible on the Web by the end user's browser;
- string OriginalDownloadUrl: the URL of the download Web page in the original language on the original EASTIN partner's Web site in which the product is presented. The URL must be accessible on the Web by the end user's browser;
- string EnglishDownloadUrl: the URL of the download Web page in English on the original EASTIN partner's Web site in which the product is presented. The URL must be accessible on the Web by the end user's browser;
- string[] UserManualUrls: an array containing the URLs of product's user manuals;

- string[] VideoUrls: an array containing the URLs of product's demo videos;
- string[] BrochureUrls: an array containing the URLs of product's brochures;
- string[] FurtherInfoUrls: an array containing the URLs of other information available on the Web
 related to the product;
- FeatureDto[] Features: an array of FeatureDto objects containing all the EASTIN Taxonomy features (with measure values if needed) for this product.

SmallActorDto

- string ActorCode*: the id of the actor in the EASTIN partner's local database;
- string OriginalFullName*: the full name of the actor in the original language;
- string Country*: the country code of the actor in ISO 3166-1-alpha-2 code (for example "IT", "US", etc.);
- dateTime InsertDate*: the insert date of the actor in the EASTIN partner's local database;
- dateTime LastUpdateDate*: the insert date of the actor in the EASTIN partner's local database.

ActorDto

- string ActorCode*: the id of the actor in the EASTIN partner's local database;
- string OriginalFullName*: the full name of the actor in the original language;
- string Country*: the country code of the actor in ISO 3166-1-alpha-2 code (for example "IT", "US", etc.);
- dateTime InsertDate*: the insert date of the actor in the EASTIN partner's local database;
- dateTime LastUpdateDate*: the insert date of the actor in the EASTIN partner's local database;
- string ShortName*: the short name of the actor;
- string EnglishFullName*: the full name of the actor in English;
- string OriginalDescription: the description of the Actor in the original language;
- string EnglishDescription: the description of the Actor in English;
- dateTime StartDate*: the start date of the actor
- dateTime EndDate: the end date of the actor
- string ContactBody: the reference organization of the actor;
- string Address: the address of the actor;
- string PostalCode: the postal code of the actor;
- string Town: the town of the actor;
- string Phone: the phone of the actor;
- string Fax: the fax of the actor;
- string Email: the email of the actor;
- string Skype: the Skype account name of the actor;
- string WebSiteUrl: the Web site URL of the actor. The URL should be accessible on the Web by the end user's browser;
- string ContactPersonFullName: the complete name of the contact person for the actor;
- string OriginalUrI: the URL of the Web page in the original language on the original EASTIN partner's Web site in which the actor is presented. The URL must be accessible on the Web by the end user's browser;
- string EnglishUrl: the URL of the Web page in English on the original EASTIN partner's Web site in which the actor is presented. The URL must be accessible on the Web by the end user's browser
- string[] SocialNetworkUrls: an array of URLs linking to the actor page inside the main social networks (for example Facebook, Twitter, LinkedIn, etc.);
- string[] IcfCodes*: the array of all EASTIN ICF classification codes of the actor (for example ["b1", "d2"]);

string[] IsoCodes*: the array of all ISO classification codes of the actor (for example ["12.22", "09.03.03"]);

SmallAssociatedInfoDto

- string AssociatedInfoCode*: the ID of the associated information document in the EASTIN partner's local database;
- string Authors*: a string containing the names (or the initials) of the authors of the associated information document (this is not an array but a single string);
- string OriginalTitle*: the original title in the native language of the associated information document
- string EnglishTitle*: the English translation of the original title of the associated information document
- string OriginalLanguage*: the ISO 639-1 code of the native language of the associated information document (for example: "en", "it", "de");
- dateTime InsertDate*: the insert date of the associated information document in the EASTIN partner's local database;
- dateTime LastUpdateDate*: the last update date of the associated information document in EASTIN partner's local database.

AssociatedInfoDto

- string AssociatedInfoCode*: the ID of the associated information document in the EASTIN partner's local database;
- string Authors*: a string containing the names (or the initials) of the authors of the associated information document (this is not an array but a single string);
- string OriginalTitle*: the original title in the native language of the associated information document
- string EnglishTitle*: the English translation of the original title of the associated information document
- string OriginalLanguage*: the ISO 639-1 code of the native language of the associated information document (for example: "en", "it", "de");
- dateTime InsertDate*: the insert date of the associated information document in the EASTIN partner's local database;
- dateTime LastUpdateDate*: the last update date of the associated information document in EASTIN partner's local database
- integer PublicationYear*: the publication year of the associated information document;
- string PublishingDetails: the publishing details (for example the publishing house) of the associated information document;
- string OriginalAbstract: the abstract of the associated information document in the original language;
- string EnglishAbstract: the abstract of the associated information document in the original language;
- string OriginalUrl: the URL of the Web page in the original language on the original EASTIN partner's Web site in which the associated information document is presented. The URL must be accessible on the Web by the end user's browser;
- string EnglishUrI: the URL of the Web page in English on the original EASTIN partner's web site in which the associated information document is presented. The URL must be accessible on the Web by the end user's browser;
- string OriginalDownloadUrl: the URL for the download of the associated information document in the original language;
- string EnglishDownloadUrl: the URL for the download of the associated information document in English;
- string ImageUrI: the URL of the picture related to the associated information document (used when displaying the detail view of the associated information document in EASTIN Portal). The URL must be accessible on the Web by the end user's browser;

- string[] FurtherInfoUrls: an array containing the URLs of other information present on the Web related to the associated information document;
- string[] IcfCodes*: the array of all EASTIN ICF classification codes of the associated information document (for example ["b1", "d2"]);
- string[] IsoCodes*: the array of all ISO classification codes of the associated information document (for example ["12.22", "09.03.03"]);

ANNEX - EASTIN Web service REST API implementation - Version 2.0

Introduction

In the following specifications the most widely accepted JSON formatting best practices have been adopted:

- All empty objects/properties in the JSON objects may be valorised to null or omitted.
- Numbers representing decimal values use the "." separator for the fractional part (ex: "value":3.56)
- The date properties must always be expressed in ISO 8601 format and use UTC time. The specific ISO format used is "yyyy-MM-ddTHH:mm:ss.fffz" (ex.: "insertDate":"2014-01-03T14:05:59.423Z") where:
 - yyyy: four digit year (ex.: 2016)
 - MM: two digit month (ex: 01 is January; the eventual leading zero must be specified)
 - dd: two digit day (ex: 03 is the third day of the month; the eventual leading zero must be specified)
 - HH: two digit hours in 24 hours format (ex.: 14 is two PM, 01 is one AM; the eventual leading zero must be specified)
 - mm: two digit minutes (ex.: 05; the eventual leading zero must be specified)
 - ss: two digit seconds (ex.: 59; the eventual leading zero must be specified)
 - fff: three digit milliseconds (ex.: 423, 120 (eventual ending zeros must be specified), 003 (eventual leading zeros must be specified)
 - z: a character indicating the time zone. It must always be equal to Z (indicating UTC time)
- For every method the results are embedded inside a wrapper JSON object with this format (many different implementations of this schema will be found in the examples below):

{

```
"apiVersion":"<API version>",
```

"data": { <JSON object containing results> },

```
"error": { <JSON object containing eventual errors> }
```

}

where **apiVersion** is always valorized (to "2.0" for the current version of the specs) while the valorizations of **data** and **error** are mutually exclusive. If present the error object has this format: "error":

{

"message":"<a string representing the error occurred>"

}

(even if it contains just a single property "error" has been designed as a complex JSON object because in future API versions other properties may be added)

• Each response should contain the proper HTTP status code, for example 200 for successful execution, 404 for resource not found, 500 for internal server error and so on.

Here below the detailed description of the implementation of the Web Service specs via a REST based API can be found.

Batch methods

Matching method in specifications: GetIsoClasses()

URL:

```
http://<partner_server>/<partner_defined_subpath>/v2.0/isoclasses
(ex.: http://portale.siva.it/eastinwebapi/v2.0/isoclasses)
```

HTTP Verb:

GET

URL parameters: None

Request content parameters: None.

```
A JSON object. Ex.:
{
"apiVersion":"2.0",
"data":
{
"items":
[
{
"isoCode":"04",
"title":"Ausili per l'assistenza di funzioni corporee",
"scopeNote":"Prodotti che servono per ...",
"productCount":233
},
{
"isoCode":"04.03",
"title":"Ausili per la respirazione",
"scopeNote": "Apparecchiature per assistere..."
,"productCount":0
},
{
"isoCode":"04.03.03",
"title":"Dispositivi per il pretrattamento dell'aria per inalazione",
"scopeNote":"Dispositivi che trattano...",
"productCount":24
},
]
},
```

"error":null } (this property may be omitted)

Notes

The **data.items** property in the returned JSON object is an array of JSON-serialized IsoClassLocalizationDto objects.

Matching method in specifications: GetKeywords()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/keywords (ex.: http://portale.siva.it/eastinwebapi/v2.0/keywords)

HTTP Verb:

GET

URL parameters:

None

Request content parameters: None.

```
A JSON object: Ex.:
{
"apiVersion":"2.0",
"data":
{
"items":
[
{
"keywordId":"E314R",
"text": "wheelchairs",
"isoCodes":["12.22.03", "12.22.06"]
},
{
"keywordId":"E31234",
"text":"poles",
"isoCodes":["12.22.03", "12.22.06"]
},
...
{
"keywordId":"DF234",
"text": "hoisters",
"isoCodes":["12.22.03", "12.22.06"]
}
]
```

```
},
"error":null
}
```

(this property may be omitted)

Notes

The **data.items** property in the returned JSON object is an array of JSON-serialized KeywordDto objects.

Live search methods

1. Product searches

Matching method in specifications: FindSmallProducts()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/products (ex.: http://portale.siva.it/eastinwebapi/v2.0/products)

HTTP Verb:

POST

URL parameters: None

Request content parameters:

```
A JSON object. Ex.:
{
"apiVersion":"2.0",
"params":
{
"isoCodes":["12.22.03", "12.22.06"],
"feaures":
ſ
{ "featureId":122, "valueMin":0.0 "valueMax":0.0 },
{ "featureId":2, "valueMin":8.0, "valueMax":100.5 }
],
"commercialName":"ministar",
"manufacturer": "offcarr",
"insertDateMin":"2014-03-31T13:22:05.245Z",
"insertDateMax":"2016-12-01T17:22:56.941Z"
}
}
```

```
A JSON object. Ex.: {

"apiVersion":"2.0",
```

```
"data":
{
"items":
ſ
{
"productCode":"47056",
"isoCodePrimary":"12.22.03",
"isoCodesOptional":["12.22.06", "12.22.09"],
"commercialName":"Cleo ultralet kørestol til børn",
"manufacturerCode":"1A2",
"manufacturerOriginalFullName":"Sunrise Medical B.V.",
"insertDate":"2015-10-30T09:37:50.130Z",
"lastUpdateDate":"2016-11-30T10:33:38.204Z",
"thumbnailImageUrl": "http://portale.siva.it/files/images/product/thumbs/18459_s.jpg"
},
{
"productCode":"E23091",
"isoCodePrimary":"12.22.03",
"isoCodesOptional":["12.22.06"],
"commercialName":"Quickie Neon<sup>2</sup> Swing Away",
"manufacturerCode":"1A2",
"manufacturerOriginalFullName":"Sunrise Medical B.V.",
"insertDate":"2006-11-13T09:37:50.031Z",
"lastUpdateDate":"2016-11-28T09:37:50.123Z",
"thumbnailImageUrl": "http://portale.siva.it/files/images/product/thumbs/28659_s.jpg"
},
...
{
"productCode":"E6501",
"isoCodePrimary":"12.22.03",
"isoCodesOptional":["12.22", "12.22.06", "12.22.09"],
"commercialName":"Küschall Compact / Compact Junior 2009",
"manufacturerCode":"R13",
"manufacturerOriginalFullName":"Offcarr",
"insertDate":"1995-10-31T00:00:00.000Z",
"lastUpdateDate":"2016-11-28T00:00:00.000Z",
"thumbnailImageUrl": "http://portale.siva.it/files/images/product/thumbs/1839 s.jpg"
}
]
},
"error":null
                                       (this property may be omitted)
}
```

Notes

Any of the fields in the **params** JSON object (representing the method's parameters) may be empty; if no field is valorised, such as in this example:

{

```
"apiVersion":"2.0",
"params":null
}
```

the method should return all Products available in the db. The **isoCodes** array contains the ISO codes (specified as string). The **features** array contains the JSON serialization of FeatureDto objects (see the general specs above for the complete description). If a feature is of type **Attribute** its values from and to will be both 0.0 and won't be considered in the query building process; if a feature is of type **Measure** than see the specs for **FindSmallProducts()** above for the detailed query building criteria. Feature of type **Cluster** will never be used as parameter. For a complete reference about Features see the **ANNEX - EASTIN feature vocabulary** above.

The **data.items** property in the returned JSON object is an array of JSON serialized **SmallProductDto** objects.

Matching method in specifications: GetProduct()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/products/<productCode> (ex.: http://portale.siva.it/eastinwebapi/v2.0/products/EF4234)

HTTP Verb:

GET

URL parameters:productCode (ex.: EF4234)

Request content parameters: None

```
Returns:
A JSON object: Ex.:
{
"apiVersion":"2.0",
"data":
{
"productCode":"EF4234",
"isoCodePrimary":"12.22.03",
"isoCodesOptional":["12.22.06", "12.22.09"],
"commercialName":"Cleo ultralet kørestol til børn",
"manufacturerCode":"12",
"manufacturerOriginalFullName":"Sunrise Medical B.V.",
"insertDate":"2015-10-30T09:37:50.130Z",
"lastUpdateDate":"2016-11-30T10:33:38.204Z",
"thumbnailImageUrl": "http://portale.siva.it/files/images/product/thumbs/18659_s.jpg",
"isReviewAllowed":true,
"manufacturerAddress":"Via Trasimeno 3",
"manufacturerPostalCode": "20100",
"manufacturerTown": "Milano",
```

Version 5.0

```
"manufacturerCountry":"IT",
"manufacturerPhone":"+39 02 419 2249",
"manufacturerFax":"+39 02 419 2224",
"manufacturerEmail": "info@sunrise.com",
"manufacturerSkype": "sunriseSkype",
"manufacturerWebSiteUrl": "http://www.sunrise.com",
"manufacturerSocialNetworkUrls":["http://www.facebook.com/meyra/", "http://www.linkedin.com/meyra"],
"imageUrl": "http://portale.siva.it/files/images/product/thumbs/18659 b.jpg",
"originalDescription":"Questo prodotto è composto da...",
"englishDescription": "This product is made of ... ",
"originalUrl": "http://portale.siva.it/it-IT/databases/products/detail/id-18564",
"englishUrl": "http://portale.siva.it/en-GB/databases/products/detail/id-18564",
"originalDownloadUrl": "http://portale.siva.it/it-IT/databases/products/download/id-18564",
"englishDownloadUrl": "http://portale.siva.it/en-GB/databases/products/download/id-18564",
"userManualUrls":["http://www.someurl1.com", "http://www.someurl2.com"],
"videoUrls": ["http://www.someurl3.com", "http://www.someurl4.com"],
"brochureUrls": ["http://www.someurl5.com", "http://www.someurl6.com"],
"furtherInfoUrls": ["http://www.someurl7.com", "http://www.someurl8.com"],
"features":
[
{ "featureId":122, "valueMin":0.0, "valueMax":0.0 },
{ "featureId":2, "valueMin":50.3", "valueMax":200.15 },
{ "featureId":7, "valueMin":1.0", "valueMax":10.0 }
1
},
"error":null
}
```

Notes

The **data** property in the returned JSON object is a JSON serialized **ProductDto** object.

2. Actor searches

Matching method in specifications: FindSmallActors()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/actors (ex.: http://portale.siva.it/eastinwebapi/v2.0/actors)

HTTP Verb: POST

URL parameters: None

Request content parameters:

Returns:

```
A JSON object: Ex.:
{
"apiVersion":"2.0",
"data":
{
"items":
[
{
"actorCode ":"F356",
"originalFullName ":"Meyra Inc.",
"country":"DE",
"insertDate":"2014-10-30T09:37:50.130Z",
"lastUpdateDate":"2015-11-30T10:33:38.204Z"
},
{
"actorCode ":"OJ2343456",
"originalFullName ":"Merac ltd.",
"country":"UK",
"insertDate":"2013-10-30T09:37:50.130Z",
"lastUpdateDate":"2014-11-30T10:33:38.204Z"
},
...
{
"actorCode ":"23456",
"originalFullName ":"Melt Spa.",
"country":"IT",
"insertDate":"2011-10-30T09:37:50.130Z",
"lastUpdateDate":"2016-11-30T10:33:38.204Z"
}
]
},
"error":null
```

(this property may be omitted)

}

Notes

Any of the fields in the **params** JSON object (representing the method's parameters) except **actorType** may be empty; if no other field is valorised, such as in this example:

```
{
"apiVersion":"2.0",
"params":
{
"actorType": "companies",
"isoCodes":null,
                                       (this property may be omitted)
"icfCodes":null,
                                       (this property may be omitted)
"actorName":null,
                                       (this property may be omitted)
"insertDateMin":null,
                                       (this property may be omitted)
"insertDateMax":null
                                       (this property may be omitted)
}
}
```

the method should return all Actors of the specified type available in the db. For the parameters the same considerations stand as for **FindSmallActors()** specs (see above).

The data.items property in the returned JSON object is an array of JSON serialized SmallActorDto objects.

Matching method in specifications: GetActor()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/actors/<actorType>/<actorCode> (ex.: http://portale.siva.it/eastinwebapi>/v2.0/actors/companies/1R233)

HTTP Verb:

GET

URL parameters:

- actorType (ex.: companies)
- actorCode (ex.: 1R233)

```
Request content parameters: None
```

```
Returns:

A JSON object: Ex.:

{

"apiVersion":"2.0",

"data":

{

"actorCode":"1R233",

"originalFullName":"Meyra Srl",
```

"country":"IT", "insertDate":"2015-10-30T09:37:50.130Z", "lastUpdateDate":"2016-11-30T10:33:38.204Z", "shortName":"Meyra", "englishFullName": "Meyra Gmbh", "originalDescription":"Questa azienda è...", "englishDescription": "This company is...", "startDate":"2015-10-30T09:37:50.130Z", "endDate":"2016-11-30T10:33:38.204Z", "contactBody": "Meyra International", "address":"Via Trasimeno 3", "postalCode":"20100", "town":"Milano", "phone":"+39 02 419 2249", "fax":"+39 02 419 2224", "email":"info@sunrise.com", "skype": "meyraSkype", "webSiteUrl": "http://www.meyra.com", "contactPersonFullName": "Mr. John Smith", "originalUrl": "http://www.meyra.com/it", "englishUrl": "http:// www.meyra.com/en", "socialNetworkUrls":["http://www.facebook.com/meyra/", "http://www.linkedin.com/meyra"], "icfCodes": ["b1", "d2"], "isoCodes":["12.22.03", "12.22.06"] }, "error":null (this property may be omitted) }

Notes

The data property in the returned JSON object is a JSON serialized ActorDto object.

3. Associated information searches

Matching method in specifications: FindSmallAssociatedInfos()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/associatedinfo (ex.: http://portale.siva.it/eastinwebapi/v2.0/associatedinfo)

HTTP Verb: POST

URL parameters: None

Request content parameters: A JSON object. Ex.:

```
{
"apiVersion":"2.0",
"params":
{
"infoType":"articles",
"isoCodes":["12.22.03", "12.22.06"],
"icfCodes": ["b1", "d2"],
"title": "Disabilità e lavoro",
"author": "Andrich",
"insertDateMin":"2014-03-31T13:22:05.245Z",
"insertDateMax":"2016-12-01T17:22:56.941Z"
}
}
Returns:
A JSON object: Ex.:
{
"apiVersion":"2.0",
"data":
{
"items":
[
ł
"associatedInfoCode ":"A34R324",
"authors":"R. Andrich, V. Gower",
"originalTitle": "Disabilità e lavoro",
"englishTitle":"Disability and work",
"originalLanguage":"it",
"insertDate":"2014-10-30T09:37:50.130Z",
"lastUpdateDate":"2015-11-30T10:33:38.204Z"
},
{
"associatedInfoCode ":"235ERWT5",
"authors":"R. Andrich, V. Gower",
"originalTitle": "Disabilità e lavoro",
"englishTitle":"Disability and work",
"originalLanguage":"it",
"insertDate":"2014-10-30T09:37:50.130Z",
"lastUpdateDate":"2015-11-30T10:33:38.204Z"
},
...
{
"associatedInfoCode ":"3455DFGSG",
"authors":"R. Andrich, V. Gower",
"originalTitle":"Disabilità e lavoro",
"englishTitle":"Disability and work",
"originalLanguage":"it",
```

```
"insertDate":"2014-10-30T09:37:50.130Z",
"lastUpdateDate":"2015-11-30T10:33:38.204Z"
}
]
},
"error":null (this property may be omitted)
}
```

Notes

Any of the fields in the **params** JSON object (representing the method's parameters) except **infoType** may be empty; if no other field is valorised, such as in this example:

```
{
"apiVersion":"2.0",
"params":
{
"infoType":"articles",
"isoCodes":null,
                                        (this property may be omitted)
"icfCodes":null,
                                        (this property may be omitted)
"title":null,
                                        (this property may be omitted)
"author":null,
                                        (this property may be omitted)
"insertDateMin":null,
                                        (this property may be omitted)
"insertDateMax":null
                                        (this property may be omitted)
}
}
```

the method should return all AssociatedInfo of the specified type available in the db. For the parameters the same considerations stand as for **FindSmallAssociatedInfos()** specs (see above).

The **data.items** property in the returned JSON object is an array of JSON serialized SmallAssociatedInfoDto objects.

Matching method in specifications: GetAssociatedInfo()

URL:

http://<partner_server>/<partner_defined_subpath>/v2.0/associatedinfo/<infoType>/<associatedInfoCode> (ex.: http://portale.siva.it/eastinwebapi/v2.0/associatedinfo/articles/1ERT244S)

HTTP Verb:

GET

URL parameters:

- infoType (ex.: articles)
- associatedInfoCode (ex.: 1ERT244S)

Request content parameters:

None

```
Returns:
A JSON object: Ex.:
{
"apiVersion":"2.0",
"data":
{
"associatedInfoCode":"1ERT244S",
"authors":"R. Andrich, V. Gower",
"originalTitle": "Disabilità e lavoro"
"englishTitle":"Disability and work",
"originalLanguage":"it",
"insertDate":"2014-10-30T09:37:50.130Z",
"lastUpdateDate":"2015-11-30T10:33:38.204Z",
"publicationYear":2011,
"publishingDetails": "Associated Press",
"originalAbstract": "Questo articolo parla di...",
"englishAbstract": "This article talks about...",
"originalUrl": "http://portale.siva.it/it/associatedinfo/articles/1ERT244S",
"englishUrl":" http://portale.siva.it/en/associatedinfo/articles/1ERT244S",
"originalDownloadUrl": "http://portale.siva.it/it/associatedinfo/articles/1ERT244S/download",
"englishDownloadUrl":" http://portale.siva.it/en/associatedinfo/articles/1ERT244S/download",
"imageUrl": "http://portale.siva.it/en/associatedinfo/articles/1ERT244S b.jpg",
"furtherInfoUrls": ["http://www.someurl1.com", "http://www.someurl2.com"],
"icfCodes": ["b1", "d2"],
"isoCodes":["12.22.03", "12.22.06"]
},
"error":null
                                        (this property may be omitted)
}
```

Notes

The **data** property in the returned JSON object is a JSON serialized AssociatedInfoDto object.