# Integrating a structured metadata management system into the TALIA Semantic Framework

## 0. The TALIA project

The TALIA project – which stands for "Territorial Appropriation of Leading-edge Innovation Actions" – has been funded by European Commission in the context of an INTERREG-MED call. The broader goals of this call were to promote sustainable growth supported by technology in the mediterranean area, and, more specifically, to increase the capacity of communication and cooperation among the main actors in the most important social and economical sectors in EU.

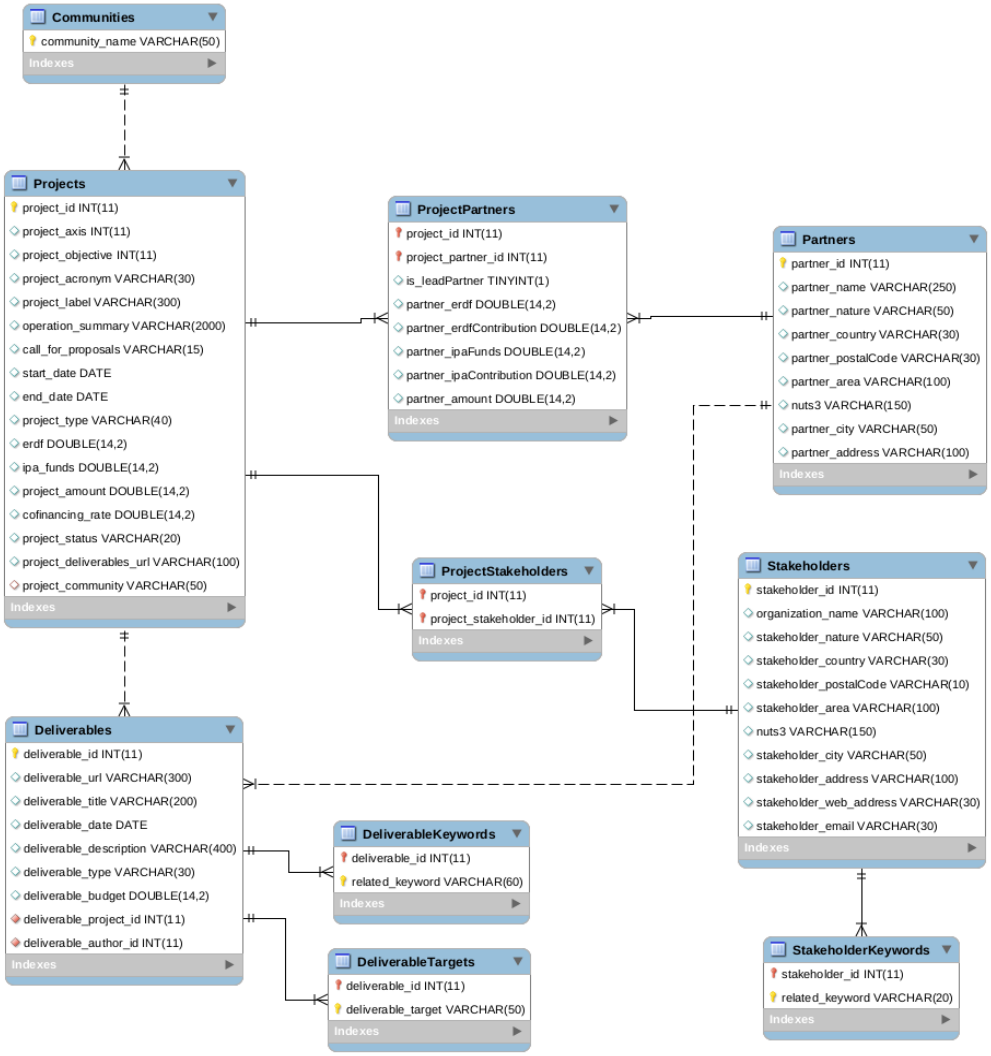
To accomplish these goals, TALIA has in turn, as its main goal, the development and release of a decision support system based on intelligent analysis of the textual content of documents written in each INTERREG-MED funded project. The decision support system, which makes available three different services to the end-user, is built upon the Semantic Framework, a platform for indexing and semantic analysis of documents developed by University of Bari "Aldo Moro", L.A.C.A.M. – S.W.A.P. Research Group as a partner of the TALIA project.

## 1. Why metadata management?

TALIA, involving a great number of projects and partners, outlines a project domain of modest complexity. While non-structured information from the project domain is exploited using services of the Semantic Framework, structured data, which describe subjects and entities involved in the project, are more suitably stored and organized in a relational database, thus being available for later integration in semantic services. Structured data describing the project domain include: data regarding all the vertical projects whose documents are analyzed in the TALIA project; data describing public deliverables of each vertical project and each partner and stakeholder involved.

Semantic services, retrieving information from vertical project's deliverables, can make their results more informative displaying some of the structured data available. Examples are: data regarding each deliverable itself (eg. a brief description of the deliverable, the budget assigned to it, its date and target audience, etc.), the project it is related to (eg. financial information of the project), the partners involved in the project and those who authored the deliverable (eg. the country each partner comes from, geographical and financial information about each partner).

## 2. Database Structure




*Structure of the relational database*

The relational database was created using MySql technology. The main concepts represented as entities in the database structure are: Deliverables, Projects, Partners, Stakeholders and Communities. According to the semantics in the TALIA project's domain, each real-world relationship between concepts has been modeled as a one-to-many relationship between entities, except those involving projects and their partners, and projects and their stakeholders, which have been modeled as many-to-many relationships (i.e. each Project can have several Partners as participants and one single Partner can participate to several different Projects). Additional entities have been created to represent Keywords and Targets associated with each Deliverable and Keywords associated with each Stakeholder.

## 3. Loading Metadata


*.json file structure regarding Blue Growth Community*

| Axis | Objective | Acronym | Project label | Operation summary | Lead Partner | Country | Postcode | Call for proposals | Start date | End date | Type of project | ERDF | IPA Funds | Amount of the project (ERDF +IPA + national counterpart) | Co financing rate | Community |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | + RESILIENT | Mediterranean Open REsouRcEs for Social Innovation of SociaLly ResponsIve ENTerprises | +RESILIENT puts together a 4-helix partnership of 8 MED countries to tackle the need for innovation conducive to increased socially-responsive competitiveness of SMEs & stimulate new jobs, especially for companies operating in the social economy. It aims to kickstart a process of policy change at regional | Veneto Region – Operational Unit for EU and State Relations | ITALY | 30123 | 2nd call | 01/02/2018 | 31/01/2022 | Integrated project | 2.667.650,20 | 119.100,13 | 3.278.529,80 | 85% | Social and Creative |

*.xlsx Projects file structure*

Metadata stored in the database were mainly obtained from the web-crawling and web-scraping processes carried out on the INTERREG-MED official website. These processes gave as results three kinds of data files in two different formats: one .json file for each INTERREG-MED Programme Community, and two .xlsx files including data regarding the entire INTERREG-MED Programme.

Every .json file is Deliverable-centered; it includes, for each Deliverable of every Project belonging to the considered Community, data about the Deliverable itself, data regarding the Project it belongs to, the Partners involved in that Project, and more (take a look at .json file structure above).

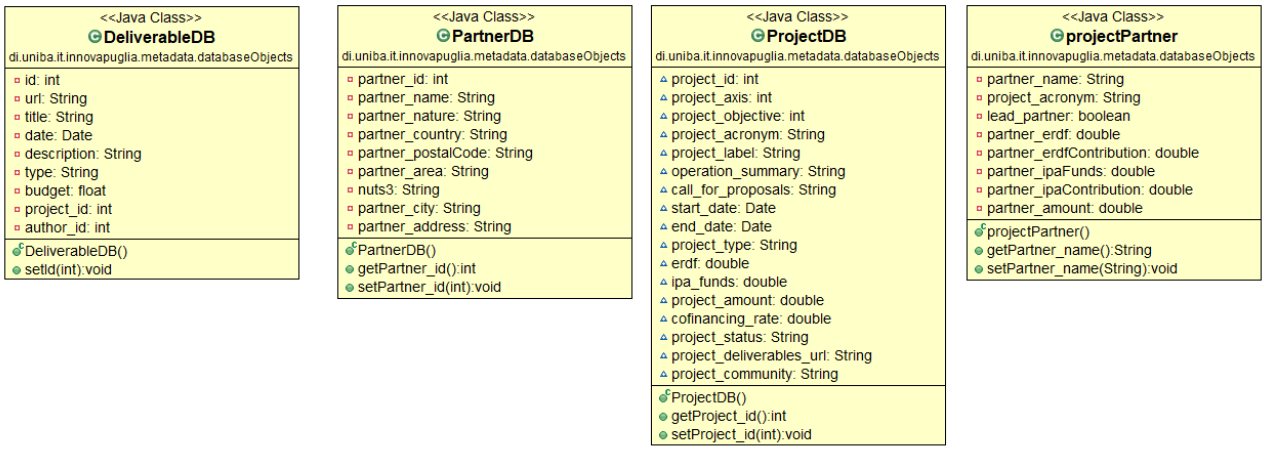
The two .xlsx files include information about all the vertical Projects funded by INTERREG-MED Programme calls, and information about every Partner involved in each vertical Project, respectively.

Both .json and .xlsx files have been generated by the web-scraping process; however they do not contain exactly the same data. Each .json file contains, in fact, data which are also stored in .xlsx files, because it

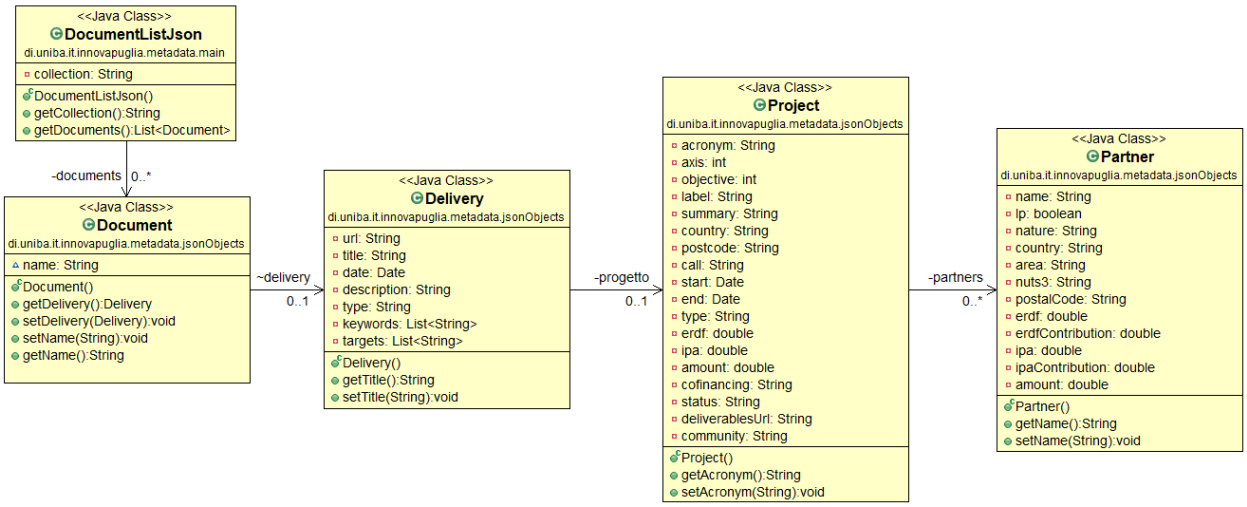
connects Deliverable data with Projects and Partners data; .json files are thus sometimes redundant, both with respect to the file itself and with respect to .xlsx files. However, .xlsx files store more complete data about Projects and Partners, as they have more attributes in their file structure; and, at the same time, .json files include some attributes which .xlsx files do not comprise. This abundance and variety of information has been carefully analyzed and used to design a complete database structure.

To be uploaded to the database, .xlsx files have been converted into the simpler .csv file format.

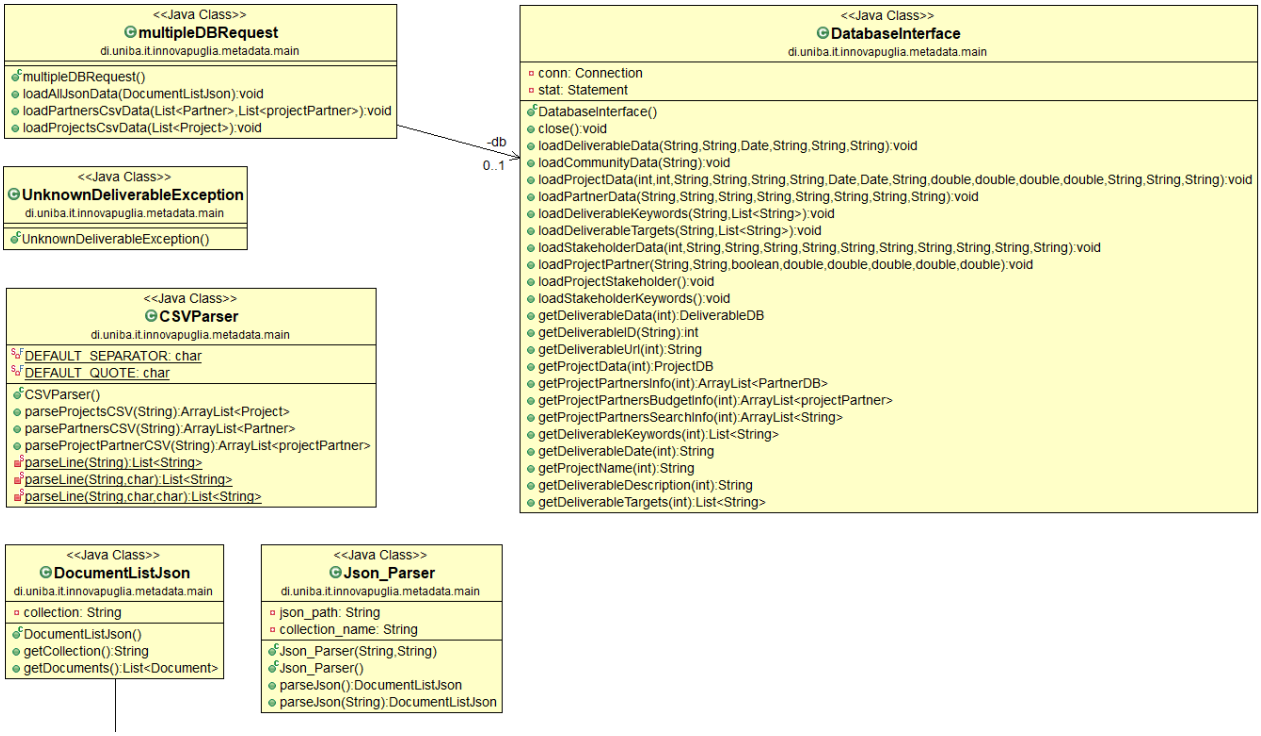
## 4. Java Package for metadata management




*UML class diagram of classes mirroring database structure (most getters and setters have been omitted)*



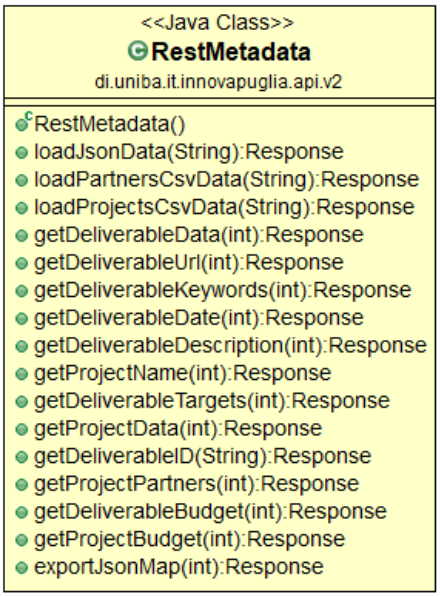

*UML class diagram of classes mirroring .json file structure (most getters and setters have been omitted)*




*UML diagram of classes managing db connection and underlying REST methods*

"Metadata" package in Innovapuglia project (alias the Semantic Framework) was created to manage both the connection to the database and the process of acquiring data from .csv and .json files. Three subpackages exist in the metadata package: databaseObjects package, jsonObjects package and main package. The databaseObjects package and the jsonObjects package include classes which mirror the structure of the database and of the .json files, respectively. The main package includes: the class which is concerned which represents the interface to the database; the classes which carry out the parsing process on .json and .csv files; two auxiliary classes: DocumentListJson class, used in the mapping process of .json files and multipleDBRequest class, used to easily load all data stored in a single .json or .csv file at once via REST APIs.

## 5. REST APIs for metadata management




*Java class whose methods have been exposed as REST methods*

Data loading and fetching operations implemented in metadata package are accessible to the end-user through RESTful APIs. RESTful APIs include two different kinds of methods: those whose goal is data loading from data files into the database (starting with the word “load”); and those whose goal is fetching relevant data from the database (starting with the word “get”). Additionally, the *exportJsonMap* method was designed to fetch the data which a data visualization tool needs to create a map of the results obtained using semantic services.

A more formal and detailed documentation of each REST method follows.

## Load Json Data to DB

| PUT loadJsonData | | |
|---|---|---|
| **Description** | Loads the data stored in the json file to the database. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/loadJsonData | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | json_string (`Required`) | |
| | `Description:` | The entire json file as plain text. |
| | `Example value:` | `{"id": 1, "first_name": "Jeanette", "last_name": "Penddreth", "email": "jpenddreth0@census.gov}` |
| **Example of request** | PUT<br>https://localhost:9001/ivp/v2/loadJsonData | |
| **Example of response** | `Success:` | No answer |
| | `Failure:` | SQLException \| NumberFormatException \| UnsupportedEncodingException \| ParseException \| UnknownDeliverableException |

## Load Partners' CSV Data to DB

| PUT loadPartnersCsvData | | |
|---|---|---|
| **Description** | Loads the data stored in the csv (xlsx) file relative to Partners from all Projects to the database. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/loadPartnersCsvData | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | csv_string (`Required`) | |
| | `Description:` | The entire csv file as plain text. |
| | `Example value:` | `1, Jeanette, Penddreth, jpenddreth0@census.gov` |
| **Example of request** | PUT<br>https://localhost:9001/ivp/v2/loadPartnersCsvData | |
| **Example of response** | `Success:` | No answer |
| | `Failure:` | SQLException \| NumberFormatException |

## Load Projects' CSV Data to DB

| PUT loadProjectsCsvData | | |
|---|---|---|
| **Description** | Loads the data stored in the csv (xlsx) file relative to all the Projects to the database. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/loadProjectsCsvData | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | csv_string (`Required`) | |
| | `Description:` | The entire csv file as plain text. |
| | `Example value:` | `1, Jeanette, Penddreth, jpenddreth0@census.gov` |
| **Example of request** | PUT<br>https://localhost:9001/ivp/v2/loadProjectsCsvData | |
| **Example of response** | `Success:` | No answer |
| | `Failure:` | SQLException \| NumberFormatException |

## Fetch Deliverable data from DB

| GET getDeliverableData | | |
|---|---|---|
| **Description** | The method returns the record in Deliverable database table corresponding to the given do-cid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableData | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getDeliverableData/1 | |
| **Example of response** | `Success:` | `{"id":1,"url":"https://confish.interreg-med.eu/what-we-achieve/deliverable-library/detail/?tx_elibrary_pi1%5Blivrable%5D=2353&tx_elibrary_pi1%5Baction%5D=show&tx_elibrary_pi1%5Bcontroller%5D=Frontend%5CLivrable&cHash=980eaf37be4c041e92505e271d349a39","title":"confish_poster_20102017_is","date":1508371200000,"description":"confish_poster_20102017_is","type":"Document","budget":0.0,"project_id":1}` |
| | `Failure:` | SQLException \| JsonProcessingException |

## Fetch Deliverable URL from DB

| GET getDeliverableUrl | | |
|---|---|---|
| **Description** | The method returns the value of deliverable_url field from the record in Deliverable database table corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableUrl | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getDeliverableUrl/1 | |
| **Example of response** | `Success:` | `https://confish.interreg-med.eu/what-we-achieve/deliverable-library/detail/?tx_elibrary_pi1%5Blivrable%5D=2353&tx_elibrary_pi1%5Baction%5D=show&tx_elibrary_pi1%5Bcontroller%5D=Frontend%5CLivrable&cHash=980eaf37be4c041e92505e271d349a39` |
| | `Failure:` | SQLException |

## Fetch Deliverable Keywords from DB

| GET getDeliverableKeywords | | |
|---|---|---|
| **Description** | The method returns the values in DeliverableKeywords table related to the Deliverable corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableKeywords | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getDeliverableKeywords/1 | |
| **Example of response** | `Success:` | `[coastal management, maritime issues]` |
| | `Failure:` | SQLException |

## Fetch Deliverable Date from DB

| GET getDeliverableDate | | |
|---|---|---|
| **Description** | The method returns the value of deliverable_date field from the record in Deliverable database table corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableDate | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET https://localhost:9001/ivp/v2/getDeliverableDate/1 | |
| **Example of response** | `Success:` | `2017-10-19` |
| | `Failure:` | SQLException |

## Fetch Deliverable Description from DB

| GET getDeliverableDescription | | |
|---|---|---|
| **Description** | The method returns the value of deliverable_description field from the record in Deliverable database table corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableDescription | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `13` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getDeliverableDescription/13 | |
| **Example of response** | `Success:` | `The Quality Assurance Plan (QAP) defines the methodology to be followed and the general working mechanisms for implementation of the Eco-SUSTAIN project. It will be used as an implementation guide for all EcoSUSTAIN project members.` |
| | `Failure:` | SQLException |

## Fetch Deliverable's Project Name from DB

| GET getProjectName | | |
|---|---|---|
| **Description** | The method returns the value of project_acronym field from the record in Project database table related to the Deliverable corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getProjectName | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getProjectName/1 | |
| **Example of response** | `Success:` | `ConFish` |
| | `Failure:` | SQLException |

## Fetch Deliverable Targets from DB

| GET getDeliverableTargets | | |
|---|---|---|
| **Description** | The method returns the values in DeliverableTargets table related to the Deliverable corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableTargets | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `13` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getDeliverableTargets/13 | |
| **Example of response** | `Success:` | `[General public, Interest groups including NGOs, National public authority]` |
| | `Failure:` | SQLException |

## Fetch Project data from DB

| GET getProjectData | | |
|---|---|---|
| **Description** | The method returns the record in Project database table corresponding to the Deliverable identified by the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getProjectData | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getProjectData/1 | |
| **Example of response** | `Success:` | `{"project_id":1,"project_axis":3,"project_objective":2,"project_acronym":"ConFish","project_label":"Connectivity among Mediterranean fishery stakeholders and scientists resolves connectivity of fishery populations","operation_summary":"In this era of fast global change, defining connectivity and adaptive potential of exploited marine stocks while estimating social impacts of change to the livelihood of those who directly exploit the resource are challenging but key requirements towards sustainability of fisheries. [...]","call_for_proposals":"1st call","start_date":1477872000000,"end_date":1532908800000,"project_type":"Studying","erdf":477357.0,"ipa_funds":0.0,"project_amount":561597.0,"cofinancing_rate":1.0,"project_status":"On going","project_deliverables_url":"http://confish.interreg-med.eu/what-we-achieve/deliverable-library/","project_community":"Biodiversity Protection"}` |
| | `Failure:` | SQLException \| JsonProcessingException |

## Fetch Deliverable ID from DB

| GET getDeliverableID | | |
|---|---|---|
| **Description** | The method returns the value of deliverable_id field in Deliverables database table from the record in which deliverable_name is equal to the given docName. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableID | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docName (`Required`) | |
| | `Description:` | The title of the deliverable whose id is required. |
| | `Example value:` | `confish_poster_20102017_is` |
| **Example of request** | GET https://localhost:9001/ivp/v2/getDeliverableID/confish_poster_20102017_is | |
| **Example of response** | `Success:` | `1` |
| | `Failure:` | SQLException |

## Fetch Project's Partners from DB

| GET getProjectPartners | | |
|---|---|---|
| **Description** | The method returns a string which lists all Partners involved in the Project which the deliverable identified by the given docid belongs to. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getProjectPartners | |
| **Information on the Resource** | Format of the Response:<br>Authentication required:<br>Requests limit: | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (Required) | |
| | Description: | The number used to identify each deliverable in Deliverables database table. |
| | Example value: | 1 |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getProjectPartners/1 | |
| **Example of response** | Success: | Faculty of Science, University of Zagreb\|true\|Higher education and research\|168472.0\|---Association for Nature, Environment and Sustainable Development Sunce\|false\|Interest groups including NGOs\|56100.0\|---Instituto Superior TÃ©cnico\|false\|Higher education and research\|114014.0\|---Italian National Institute for Environmental Protection and Research\|false\|National Public authority\|110000.0\|---CSIC - Institute of Marine Sciences\|false\|Higher education and research\|113011.0\| |
| | Failure: | SQLException |

## Fetch Deliverable Budget from DB

| GET getDeliverableBudget | | |
|---|---|---|
| **Description** | The method returns the value of deliverable_budget field from the record in Deliverable database table corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getDeliverableBudget | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getDeliverableBudget/1 | |
| **Example of response** | `Success:` | `54030.00` |
| | `Failure:` | SQLException |

## Fetch Project Budget from DB

| GET getProjectBudget | | |
|---|---|---|
| **Description** | The method returns the value of project_amount field from the record in Project database table related to the Deliverable corresponding to the given docid. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/getProjectBudget | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/getProjectBudget/1 | |
| **Example of response** | `Success:` | `561597.0` |
| | `Failure:` | SQLException |

## Fetch Data for Map Creation Tool

| GET exportMapData | | |
|---|---|---|
| **Description** | The method returns every avaliable information about the Deliverable corresponding to given docid, the Project it belongs to and the Partners involved in it. | |
| **URL of the resource** | https://[server address:port]/ivp/v2/exportMapData | |
| **Information on the Resource** | `Format of the Response:`<br>`Authentication required:`<br>`Requests limit:` | TEXT HTML<br>Yes<br>No |
| **Parameters** | docid (`Required`) | |
| | `Description:` | The number used to identify each deliverable in Deliverables database table. |
| | `Example value:` | `1` |
| **Example of request** | GET<br>https://localhost:9001/ivp/v2/exportMapData/1 | |
| **Example of response** | `Success:` | |
| | `Failure:` | SQLException \| JsonProcessingException |