# ProXI

***Proteomics eXpression interface specification***

# Table of Contents

# Preface

*Status of This Document*

This document presents the final specification of the ProXI (The Proteomics expression Interface) an API specification developed by members of the Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI) and ProteomeXchange partners. Distribution is unlimited.

*Version of This Document*

The current version of this document is: version 1.0.0-draft, Apr 2018.

The latest draft version of this document may be found at https://github.com/HUPI-PSI/proxi-schemas.

# Abstract

The Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI) and the ProteomeXchange Consortium (ProteomeXchange) define community standards for data representation in proteomics to facilitate data comparison, exchange and verification.

In this context, the two organizations are working together on a shared standard for downstream results, following mass spectrometry (MS) analysis. This document defines a API specification to report proteomics results.

# Chapter 1. Introduction

## 1.1. Background

Protein expression information (identification and quantification) is fundamental for understanding the function of the living organisms. Consequently, more and more research groups are working in this domain. An increasing amount of the mass spectrometry (MS)-based proteomics data generated is made available in public proteomics repositories such as PRIDE (www.ebi.ac.uk/pride/) and PeptideAtlas (wwww.peptideatlas.org). In addition, protein expression information can be accessed in other resources like the antibody-based Human Protein Atlas and protein knowledge bases such as UniProt or neXtProt. At present, scientists need to access all the different proteomics resources when they want to get all information about a given protein. The alternative is to access knowledge bases such as UniProt or neXtProt, but the amount of data coming from MS proteomics repositories in these resources is still limited.

The main goal behind the development of ProXI (Proteomics eXpression Interface) is to support researchers by making possible a distributed search of protein expression data through heterogeneous resources like the ones outlined above, among others.

The primary aim of project is to enable a biology-centric distributed search for protein expression data in bioinformatics resources taking into account metadata such as location (e.g. tissue, cell type) and level of expression, detection in disease states, etc. ProXI will provide a common entry point for showing information coming from different data resources.

PROXI will be developed with the following general tasks in mind (more specific use cases are provided in Chapter 2):

T1. Integrate protein and peptide expression information from ProteomeXchange partners and potentially others.

T2. Act as an output format of (web-) *services* that report MS-based results and thus can produce standardized result pages.

T5. Define a common query language (PXQL) as a way to allow more powerful and flexible queries, by using a specific syntax.

T6. Implement of a standard REST web service and service registry containing a defined list of well-documented methods.

**Note**: This document presents a specification, not a tutorial. As such, the presentation of technical details is deliberately direct.

## 1.2. Document Structure

The remainder of this document is structured as follows.

Chapter 2 lists use cases ProXI is designed to support.

Chapter 3 describes the terminology used in this document following the OpenAPI specifications

Chapter 4

[resolved-design-and-scope-issues] discusses the reasoning behind several design decisions taken.

Chapter 6 contains the documentation of the file.

[non-supported-use-cases] lists use cases that are currently not supported.

Conclusions are presented in [conclusions].

# Chapter 2. Use Cases for ProXI

The following cases of usage have driven the development of the ProXI data model, and are used to define the scope of the format in version 1.0.

1. ProXI API should be simple enough to make proteomics results accessible to people outside the respective fields. This should facilitate the sharing of data beyond the borders of the fields and make it accessible to non-experts.
2. ProXI should contain sufficient information to provide biological insights of all findings in a proteomics study.
3. ProXI should enable reporting at different levels of detail: ranging from a simple dataset summary, experimental design to final quantitative information.
4. It should develop and provide a Query Language (PXQL) to query mass spectrometry data server by different repositories.
5. It should be possible to query any ProXI implementation and retrieve which type of information is provided by the resource and which methods are implemented. The API should be self discoverable.
6. ProXI API should be possible to retrieve information about independent experiment as long as aggregation analysis of multiple experiments.

# Chapter 3. Notational Conventions

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” are to be interpreted as described in RFC-2119 (Bradner 1997).

# Chapter 4. API standard (OpenAPI)

The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.

The OpenAPI specification can be read here https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md

# Chapter 5. API conventions

## 5.1. Naming conventions

1. A **resource** SHOULD be represented as a collection. For example, “datasets” is a collection of datasets where we can identify a single “dataset” using the resource URI “/datasets/{accession}”.
2. MUST NOT use trailing forward slash (/) in URIs:

```
  http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/
  http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets    // This is much better
version
```

1. MUST NOT use underscores ( _ ), instead use ( - ):

It’s possible to use an underscore in place of hyphen to be used as separator – But depending on the application’s font, it’s possible that the underscore (_) character can either get partially obscured or completely hidden in some browsers or screens.

```
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/PXD000001/proteins/QPR001/peptide
s-scores   //More readable
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/PXD000001/proteins/QPR001/peptide
s_scores  //More error prone
```

1. SHOULD NOT Use uppercase letters in URIs:

When convenient, lowercase letters should be consistently preferred in URI paths. **RFC 3986** defines URIs as case-sensitive except for the scheme and host components. e.g.

```
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/PXD000001        // Better
representation
http://www.ebi.ac.uk/pride/archive/proxi/v1/Datasets/PXD000001
```

1. MUST NOT use file extensions

File extensions look bad and do not add any advantage. Removing them decrease the length of URIs as well. No reason to keep them.

```
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/PXD000001.json
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/PXD000001        // This is
correct URI
```

1. MUST NOT use CRUD function names in URIs

URIs MUST NOT be used to indicate that a CRUD function is performed. URIs should be used to

uniquely identify resources and not any action upon them. HTTP request methods (headers) should be used to indicate which CRUD function is performed.

```
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets                          //Get all
datasets
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets/{accession}              //Get
datasets for given Accession
```

# 5.2. Filtering collections

1. We RECOMMENDED to use query component to filter URI collection

Many times, you will come across requirements where you will need a collection of resources sorted, filtered or limited based on some certain resource attribute.

For this, do not create new APIs – rather enable sorting, filtering and pagination capabilities in resource collection API and pass the input parameters as query parameters:

```
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets?species=Human
```

## 5.2.1. Compact vs Full Object versions

1. We RECOMMENDED the use of **Compact** version of an object as default behavior for collections end-points.

Each object in the API MUST provide two flavours (Compact or Full). The **Compact** version of an object will be used when retrieve collections of the Object, for example:

```
http://www.ebi.ac.uk/pride/archive/proxi/v1/datasets
```

The Compact version will contains all the mandatory attributes for an Object. When designing each Data Object, mandatory fields should be the Minimum Information requeried to identified the object.

# Chapter 6. Format specification

## 6.1. Representing Ontology Terms

Each object in **ProXI** API will be heavy represented using controlled vocabulary (CV) terms (https://www.ebi.ac.uk/ols/index). An ontology term will be represent by a cvLabel (name of the Ontology database), accession (accession of the Term in the Ontology database), name (name of the Term in the Ontology database), value (value of the a parametrized Term).

```
{
   cvLabel: "PSI-MS",
   accession: "MS:1000449",
   name: "LTQ Orbitrap"
}
```

The **cvLabel** and **name** MUST be provided, the **cvLabel** is RECOMMENDED and **value** is OPTIONAL.

## 6.2. Retrieving Dataset Information

### 6.2.1. Dataset Object schema

A ProteomeXchange dataset contains the minimum metadata to describe a Proteomics experiments (http://www.proteomexchange.org/docs/guidelines_px.pdf). The ProXI dataset endpoint will retrive all the datasets from ProteomeXchange members including datasets after ProteomeXchange creation (PXD0000001) or internal datasets (MSVnnn - MassIVE or PRD000001 - PRIDE or PAe00002 - PeptideAtlas).

The philosophy behind the design of **Dataset** schema is to keep it as flexible as possible with an overall structure based on the heavy use of controlled vocabulary (CV) terms Section 6.1.

This is the list of elements in the schema:

- *Dataset*: This is the root element with mandatory attributes.
- *accession* (**required**): The unique accession of the dataset in the resource (e.g. PXD008339).
- *title* (**required**): The title of the dataset (e.g. Characterisation of proteome of a novel Escherichia coli strain exhibiting mucoviscous phenotype.)
- *summary*: Description of the dataset (e.g. Protein expression by E. coli 26561 during the late-exponential phase of cultures under anaerobic conditions was examined. E. coli 26561 is a multidrug resistant (MDR) and shows an unusual hyper-mucoviscous phenotype. Resistance includes ESBL (CTX-M-14) and proteome was determined with and without exposure to sub-MIC concentrations of the 3rd generation cephalosporin ceftazidime.)
- *species* (**required**): Contains information about the species included in the dataset as Ontology Terms.

```
[
  {
    cvLabel:"MS",
    accession:"MS:1001469",
    name:"taxonomy: scientific name",
    value:"Escherichia coli"
  },
  {
    cvLabel:"MS",
    accession:"MS:1001467",
    name:"taxonomy: NCBI TaxID",
    value:"562"
  }
]
```

- *instruments* (**required**): Element holding the overall information about the instrumentation used in the generation of the data as Ontology Terms.

```
[
  {
    cvLabel:"MS",
    accession:"MS:1002732",
    name:"Orbitrap Fusion Lumos"
  }
]
```

**Note**: The previous example contains an Ontology Term whihout **value**. Please check the Section 6.1 for a full description of Ontology Terms Objects.

- *contacts* (**required**): Information about the researchers involved in the generation and submission of the dataset. Each Contact will be a list of Ontology terms including name of the contact, email, affiliation or role (lab head).

```
[
 {
   [
    {
      cvLabel: "MS",
      accession: "MS:1000586",
      name: "contact name",
      value: "Yasset Perez-Riverol"
    },
    {
      cvLabel: "MS",
      accession: "MS:1000589",
      name: "contact email",
      value: "yperez@ebi.ac.uk"
    },
    {
      cvLabel: "MS",
      accession:"MS:1002037"
      name:"dataset submitter"
     }
   ]
 },
 {
 [
     {
       cvLabel: "MS",
       accession: "MS:1000586",
       name: "contact name",
       value: "Eric Deutch"
     },
     {
       cvLabel: "MS",
       accession: "MS:1000589",
       name: "contact email",
       value: "edeutch@systembiology.org"
     },
     {
       cvLabel: "MS",
       accession:"MS:1002037"
       name:"Head Lab"
      }
    ]
 }
]
```

- *publications* (**required**): The list of publications that the dataset has generated.

```
[
 {
  [
   {
    cvLabel: "MS",
    accession: "MS:1000879",
    name: "PubMed identifier",
    value: "29315472"
   },
   {
     cvLabel: "PRIDE",
     accession: "PRIDE:0000400",
     name: "Reference",
     value: "Mokart D, Saillard C, Zemmour C, Bisbal M, Sannini A, Chow-Chine L, Brun
JP, Faucher M, Boher JM, Toiron Y, Chabannon C, Borg JP, Gonçalves A, Camoin L. Early
prognostic factors in septic shock cancer patients: a prospective study with a
proteomic approach. Acta Anaesthesiol Scand. 2018 62(4):493-503"
   }
  ]
 }
]
```

- *modifications*: All protein modifications (natural and artificial) are listed in this record (specified as Ontology terms).

```
[
  {
   cvLabel:"MOD",
   accession:"MOD:00696",
   name:"phosphorylated residue"
  }
]
```

**Note**: If a dataset does not contain any modifications, it is also explicitly announced here with a specific CV term.

- *keywords*: One or more CV terms that define a list of keywords that may be attributed to the dataset.

```
[
  {
   cvLabel:"MS",
   accession:"MS:1001925",
   name:"submitter keyword",
   value: "Escherichia coli, mucoviscous, anaerobic, antibiotic, MIC, ceftazidime,
TMT, shotgun, quantification"
  },
  {
   cvLabel:"MS",
   accession:"MS:1001926",
   name="curator keyword",
   value:"Biological"
  }
]
```

- *datasetLink*: List of links that will allow access to the data. Different links may be used for different ways of accessing the data (for example FTP download or repository web link) or for different repositories hosting the same data.

```
[
       {
         cvLabel:"PRIDE",
         accession:"PRIDE:0000411",
         name: "Dataset FTP location",
         value: "ftp://ftp.pride.ebi.ac.uk/pride/data/archive/2018/08/PXD008339"
        }
        {
         cvLabel:"MS",
         accession:"MS:1001930",
         name:"PRIDE project URI",
         value:"http://www.ebi.ac.uk/pride/archive/projects/PXD008339"
        }
]
```

- *dataFiles*: Optional element to provide individual links to all the submitted files (mass spectrometer output files, search engine output files, etc) belonging to the dataset.

```
[
  {
   cvLabel:"PRIDE",
   accession:"PRIDE:0000404",
   name:"Associated raw file URI",

value:"ftp://ftp.pride.ebi.ac.uk/pride/data/archive/2018/08/PXD008339/2781_Phos_1.raw"
  },
  {
   cvLabel:"PRIDE",
   accession:"PRIDE:0000404",
   name:"Associated raw file URI",

value:"ftp://ftp.pride.ebi.ac.uk/pride/data/archive/2018/08/PXD008339/2781_Phos_2.raw"
  }
]
```

### 6.2.2. Dataset API entry point

Retrieving datasets will be performed using the **datasets** entry point:

```
http://www.ebi.ac.uk/pride/archive/proxy/v1/datasets
```

This entry point will retrieve a collection of datasets from the specific resources. Each collection in ProXI can be filter to refine the collection objects using the datasets properties (see Section 5.2).