Programmatic access to PDG data

J. Beringer(*)

Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Summary. — The data published by the Particle Data Group (PDG) in the *Review of Particle Physics* has traditionally been made available to the HEP community and beyond as a biennial publication in a scientific journal, in print as the *PDG Book* and the *Particle Physics Booklet*, and more recently primarily via the PDG website and the interactive pdgLive web application. Except for a number of data files downloadable from the PDG website, these formats are aimed at human reading and do not support programmatic access. In order to make all PDG data easily accessible in machine-readable format for different use cases, PDG is developing a set of new tools, namely a REST API, a downloadable database file containing the PDG data, and an associated Python package. This article presents these tools and discusses the status of their implementation at the time of the HADRON 2023 conference.

1. – Introduction

The Particle Data Group (PDG) provides a comprehensive summary of particle physics and related areas from cosmology and astrophysics in a single publication, the *Review of Particle Physics* (RPP) [1], which is updated online every year and published in a journal every two years. In 2,270 pages, the 2022 journal publication summarizes 47,000 measurements from 13,000 publications in the "Particle Listings", provides corresponding world averages and best limits in the "Summary Tables", and includes 120 individual review articles. These review articles cover a wide range of topics, including the Standard Model, astrophysics and cosmology, experimental methods and colliders, mathematical tools, kinematics, cross-section formulae, hypothetical particles and concepts, as well as selected particles and particle properties.

Originating in 1957, the contents of the RPP have traditionally been made available in print as the *PDG Book* and *Particle Physics Booklet*, and more recently primarily via the PDG website [2], the interactive pdgLive [3] web application for browsing particle

(*) On behalf of the PDG Collaboration

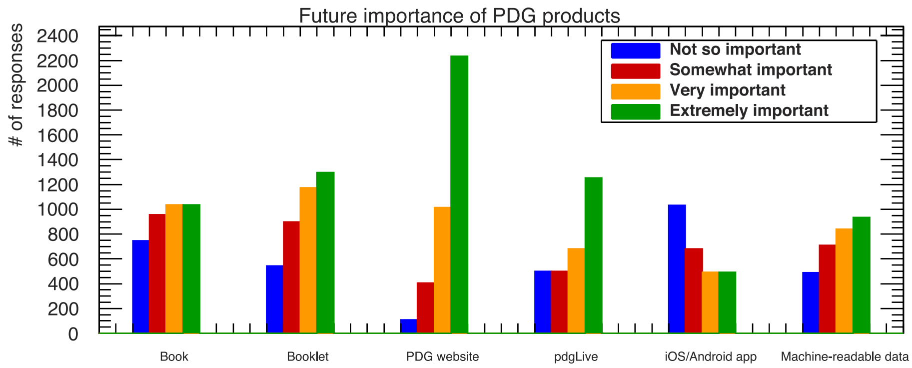

Fig. 1. – Summary of responses received in the 2022 PDG user survey to the question on how important the continuation or development of different PDG products will be for the respondents in the future.

properties data, an Android/web app for the *Particle Physics Booklet*, and a number of fixed-format data files.

In 2022, PDG carried out a comprehensive and widely-circulated survey to assess how important these different ways of making the RPP available still are today, and how useful possible new means of access would be, including in particular a full-featured smartphone app for both iOS and Android or providing all PDG data in machine-readable format. The results of this survey are summarized in fig. **1**. Perhaps not too surprisingly, the importance of programmatic access to PDG data is evident from the survey.

In order to make all PDG data easily accessible in a modern, machine-readable format, PDG is developing new tools as described below. These tools aim to address a wide range of different use cases, including for example automated update of PDG data in HEP software for simulation, data analysis or reconstruction; making calculations or plots based on PDG data; downloading of PDG data for custom averages or fits; enabling machine-learning applications using PDG data and metadata; meta-analysis of particle physics data; and the development of software applications that require an offline copy of PDG data.

2. – New tools to access PDG data in machine-readable format

The new tools are a REST API, a Python API, and SQLite database files that contain part of or the whole PDG dataset. These tools are briefly described below. Much more detailed documentation is available at [4].

Programmatic access to PDG data requires a concise and unambiguous way to reference specific data items. Because a textual or LaTeX-based description of many quantities used in particle physics would be cumbersome and difficult to use in programs, PDG defines a digital object identifier for each quantity. These are termed PDG Identifiers and consist of unique, case-insensitive, short strings. They are used throughout all PDG software. With PDG Identifiers, rather than trying to specify for example the branching fraction of the decay $B^0 \to J/\Psi(1S)K^*(892)^0\pi^+\pi^-$ in an ASCII representation, one can simply refer to its PDG Identifier `S042.214`. The PDG Identifier of each quantity can be seen in pdgLive, and a complete list is available on the PDG website.

2˙1. *REST API*. – The REST API allows the downloading of data in JSON format ([1]) from pdgLive via the new "JSON buttons" or by using the corresponding URLs directly. For example, one can get the current PDG world average of the branching fraction $B^0 \to J/\Psi(1S)K^*(892)^0\pi^+\pi^-$ in machine-readable format from

`https://pdgapi.lbl.gov/summaries/S042.214`

where `https://pdgapi.lbl.gov` is the URL of the PDG REST API, `/summaries` requests data from the PDG Summary Tables, and `S042.214` is the PDG Identifier of the quantity to be retrieved. This might return the following JSON document:


```
{
  "status_code": 200,
  "status_message": "OK",
  "request_timestamp": "2023-09-27 14:51:34 PST",
  "request_url": "https://pdgapi.lbl.gov/summaries/S042.214",
  "edition": "2023",
  "about": "For further information see https://pdg.lbl.gov/api",
  "pdgid": "S042.214",
  "description": "B0 --> J/psi(1S) K^*(892)0 pi+ pi-",
  "mode_number": 233,
  "section": "Charmonium modes",
  "pdg_values": [
    {
      "value": 6.6E-4,
      "error_positive": 2.19544984001001E-4,
      "error_negative": 2.19544984001001E-4,
      "value_text": "(6.6+-2.2)E-4",
      "type": "OUR AVERAGE"
    }
  ]
}
```


The REST API can be used both interactively and in scripts or programs. It is intended for incidental use and access is rate-limited.

2˙2. *Python API*. – The Python API provides high-level programmatic access to PDG data in one of the most widely used programming languages. For most users this will be the most convenient and versatile way to access PDG data from their programs. The Python API is implemented in package `pdg` [6] and can be installed like any other Python package.

As a simple example of how to use the Python API, the following shows how to print a list of exclusive B^0 meson branching fractions listed in the PDG Summary Tables in an interactive Python session:

```
>>> import pdg
>>> api = pdg.connect()
>>>
>>> b0 = api.get_particle_by_mcid(511)
>>>
>>> for bf in b0.exclusive_branching_fractions():
...     if not bf.is_limit:
...         print('%-40s %s' % (bf.description, bf.value))
...
B0 --> D- pi+                          0.002506090281692006
B0 --> D- rho+                         0.007574637658365 27
B0 --> D- K^*(892)+                    0.000445777202072539
```

([1]) JSON (JavaScript Object Notation) [5] is a widely used, lightweight, data-interchange format supported by all modern programming languages.

```
B0 ---> D- omega pi+                    0.0028
B0 ---> D- K+                           0.0002051413814562247
B0 ---> D- K+ pi+ pi-                   0.0003530128447741608
B0 ---> D- K+ Kbar^*(892)0              0.00088
# many more lines are not shown here
```

In this example, after importing the `pdg` package, an instance of the API that is connected to a PDG database is instantiated. By default, `connect()` uses a database of PDG Summary Table values that was current at the time the installed version of the PDG API was released. Next, the data of the desired particle is retrieved using its Monte Carlo particle number (one might also use an ASCII representation of the particle name, a PDG Identifier, or iterate over all particles), and a for loop is used to iterate over all exclusive branching fractions. Finally, a description and the central value from the PDG Summary Tables (without rounding or errors) is printed for each branching fraction, excluding any limits.

The Python API aims to handle many subtleties in PDG data in a user-friendly way. For example, when asking for the "best value of the mass of the top quark", most users would probably like to retrieve the value obtained from direct measurements, which corresponds to the parameter used in Monte-Carlo generators. But the Summary Tables also list values for the $\overline{\mathrm{MS}}$ or pole mass as obtained from cross-section measurements, so a choice has to be made. By default, in case of ambiguities, the API makes an assumption about what to return when the user asks for a single best value rather than a list of all values of a given quantity. If such assumptions are not desired, one can instantiate the API in so-called pedantic mode. In this case, assumptions are avoided and the user is warned with a Python exception in case of any ambiguities.

2·3. *Database files*. – PDG database files allow downloading part or all of the PDG data as a single file in SQLite ([2]) format. PDG database files provide low-level access to PDG data and are primarily intended for software developers who wish to have a local source of PDG data for an application, where the Python API is not suitable. In contrast to the Python API, the use of PDG database files requires not only the technical expertise for using SQL ([3]) but also a good understanding of how PDG data is organized, how different internal flags are used, and how special cases in PDG data have to be treated.

PDG database files for different editions and subsets of PDG data (e.g. without or with historical data, or with data from the Particle Listings) will be downloadable from the PDG website, and some are already available at the time of this writing. The Python API includes such a database file as its default local data source but can access other PDG database files the user may have downloaded.

3. – Status and challenges

At the time of the HADRON 2023 conference, only a preliminary version of the new tools described above was available for testing. It is expected that when these proceedings are published, a first version of the new API will be available for general use. This first version will provide access to the PDG Summary Tables but will not yet include data from the Particle Listings, which will be added in a later version.

([2]) SQLite [7] provides a relational database in the form of a single file. SQLite is widely used and available for many programming languages.
([3]) SQL (Structured Query Language) is the standard way to query relational databases.

One of the challenges in developing tools for programmatic access to PDG data comes from the fact that PDG started at a time where the primary means of scholarly communication was via printed materials. Thus for much of its existence, PDG has produced summaries to be read by humans. As a result of this, the tables in the RPP contain a lot of information (such as for example the meaning of ℓ or indenting) that is intuitively understood by human readers, well known, or clarified by footnotes. For programmatic access, such implicit information needs to be modeled and added in the form of metadata that can be easily interpreted by programs.

4. – Conclusions

The newly developed PDG API provides three complementary tools for programmatic access to PDG data, addressing a wide range of use cases. A test version was available at the time of the HADRON 2023 conference, and a first version of the API providing full access to the data in the PDG Summary Tables is expected to become available by the time these proceedings are published. Thus, for the first time, the PDG branching fraction data will be available in machine-readable format.

Development of the PDG API continues in order to also provide access to the data included in the Particle Listings, and it is planned to add data from tables in selected review articles, too, at some time in the future.

Suggestions and feedback about these new tools would be welcome and should be addressed to `api@pdg.lbl.gov`.

* * *

This work was primarily supported by the Office of Science, Office of High Energy Physics of the U.S. Department of Energy under Contract No. DE–AC02–05CH11231, with important contributions from Japan (MEXT: Ministry of Education, Culture, Sports, Science and Technology) under the U.S.-Japan Science and Technology Cooperation Program in High Energy Physics, the Italian National Institute of Nuclear Physics (INFN), the Physical Society of Japan (JPS), and the European Laboratory for Particle Physics (CERN). The PDG was designated as and is supported by the Office of Science as a Public Reusable Research (PuRe) Data Resource.

REFERENCES

[1] Workman, R.L. *et al.* (Particle Data Group), *Prog. Theor. Exp. Phys.*, **2022** (2022) 083C01, doi:10.1093/ptep/ptac097.
[2] `https://pdg.lbl.gov`
[3] `https://pdglive.lbl.gov`
[4] `https://pdgapi.lbl.gov/doc`
[5] `https://www.json.org`
[6] `https://pypi.org/project/pdg`
[7] `https://sqlite.org`