HEPData Converter Library and Tools
Project id: 15511

Michał Szostak
szostak.m.f@gmail.com
Graduate Computer Science Student
at Cracow University of Technology, Cracow, Poland
Summer Student at GS, CERN

Supervised by:
Javier Martin Montull javier.martin.montull@cern.ch
Eamonn Maguire eamonn.maguire@cern.ch

Geneva, Switzerland
September 2015
Table of Contents

1 Abstract 2
2 About HEPData 2
3 HEPData Converter 3
   3.1 Architecture ................................................. 3
   3.2 Other Components ............................................ 3
      3.2.1 HEPData Converter Web Services ................. 4
      3.2.2 HEPData Converter Web Services Client ....... 4
4 Contribution to the HEPData and Inspire Projects 4
5 Special Thanks 5
6 References 5
   6.1 Materials .................................................. 5
   6.2 Code ......................................................... 6

1 Abstract

This report summarizes my contributions to the HEPData project, which I worked on as a Summer Student Participant between July and September 2015. My work was focused on creating format conversion services for new HEPData system that will store data from high energy physics publications. Conversion between different file formats is a crucial element of the HEPData system, as it allows for: 1) the conversion of existing formats for submission to the new HEPData system; and 2) provision of data in different formats to suit the numerous data analysis pipelines operated by theoretical and experimental physicists alike.

2 About HEPData

The Durham High Energy Physics Database (HEPData) has been built up over the past four decades as a unique open-access repository for scattering data from experimental particle physics \(^1\). It is comprised of data points from plots and tables related to over eight thousand publications, some of which are from the Large Hadron Collider (LHC).

HEPData is being rewritten from the ground up in Python programming language and is based on the Invenio framework \(^2\). The software is open source with the current site available at \(^3\) with: a more streamlined submission system; advanced reviewing functionalities; interactive plotting library; and an attractive user interface.

\(^1\)http://HEPData.cedar.ac.uk
\(^2\)http://invenio-software.org/
\(^3\)http://hepdata.net
3 HEPData Converter

The HEPData converter is a python library providing the ability to convert between different data formats used by High Energy Physics Experiments. Formats supported as an input at this time are:

- OldHEPData format (sample file available at http://HEPData.cedar.ac.uk/resource/sample.input);
- YAML format used by new HEPData service https://github.com/HEPData/HEPData-submission; and
- YODA format used by Rivet toolkit 4.

Formats supported as an output at this time are:

- YAML format used by new HEPData service (description of the format is available at 5);
- YODA format used by Rivet toolkit (YODA website: 6);
- CSV; and
- ROOT 7.

The library uses many dependencies mostly related to reading and writing input and output formats (ROOT python bindings, YODA python bindings, rootpy, and yaml, etc.).

3.1 Architecture

The simplified architecture is presented in the figure below. In short conversion between any input and any output format is possible thanks to introduction of intermediary data format. The graph is presented in Fig.1.

3.2 Other Components

After developing a working version of HEPData Converter it was decided to extend the project in order to provide conversion functionality as a web service. The web service has been written using the Flask Python micro framework 8 which is quick to learn, allows for fast development, low maintenance, and has a low CPU and memory footprint. In addition to the HEPData Converter Web Service, a client to connect to the service has also been written. Its main purpose is to ease use of the web services without the need to manually create HTTP requests.

---

4https://yoda.hepforge.org/
5https://github.com/HEPData/HEPData-submission
6https://yoda.hepforge.org/
7https://root.cern.ch/
8http://flask.pocoo.org/
3.2.1 HEPData Converter Web Services

The web services provide a thin wrapper allowing for the HEPData-converter to run as a standalone service. By placing the converter in its own container, and on a different machine, we avoid installation of many libraries and dependencies on the web server, and also help distribute the load. The API and usage is explained in the package documentation available at https://github.com/HEPData/HEPData-converter-ws. In addition to being a thin wrapper around the HEPData-converter, the web service also provides a caching layer which speeds up requests for the same files. The caching element is optional, and requests must specifically state that it should be enabled.

3.2.2 HEPData Converter Web Services Client

Usage of HEPData WS requires the compression and encoding of input files with base64 encoding. In order to make this process as simple as possible, a client library was developed in order to ease development of components which have to communicate with HEPData Converter Web Services.

4 Contribution to the HEPData and Inspire Projects

During the time of my project I also proposed the introduction of Docker\(^9\) into the HEPData development workflow.

Docker is a software tool which allows user to package an application with all of its dependencies into a standardized unit for software development. Docker containers wrap up a piece of software in a complete filesystem that contains

\(^9\)https://www.docker.com/
everything it needs to run: code; system tools; and system libraries. This guarantees that it will always run the same, regardless of the environment it is running in.

The features provided through use of docker as a base for testing software was very enticing, as it allowed to bake all software dependencies (also the ones which required source code compilation) into a docker image, which was created only once (when the dependencies changed) and could be reused multiple times to run tests in an immutable environment. By doing so, test run time decreased on the Travis CI environment from 15 minutes to 1 minute.

This approach has been validated by its adoption by the Inspire project at CERN also.

5 Special Thanks

I would particularly like to thank my supervisors Javier Martin Montull, and Eamonn Maguire for their guidance, help and all the ideas which helped me make the most of my work here at CERN.

I would also like to thank Graeme Watt whose comments and insight helped to make this software better.

Additionally I would also like to thank CERN, for the unique opportunity to gain hands on experience as well as hopefully create something of use to others, and Summer Student Team for all their hard work done on our behalf.

Last but not least, I would like to express my thanks towards my fellow Summer Students and the rest of my friends and colleagues at CERN for great moments we shared and the precious discussions we had.

6 References

All relevant materials with links to them are presented below. Additionally links to all Github repositories are provided below.

6.1 Materials

Here are collected all links to materials connected to HEPData explaining its purpose, architecture, etc.

1. http://HEPData.cedar.ac.uk/ original (old) HEPData service

2. http://HEPData.cedar.ac.uk/resource/hepdata-aahep7.pdf An overview of the HEPData project which was presented in a talk at AAHEP7 on 2nd April 2014

3. http://HEPData.cedar.ac.uk/resource/hepdata-aahep8.pdf An overview of the HEPData project which was presented in a talk at AAHEP8 on 2nd September 2015
4. https://www.slideshare.net/secret/Li7QQ7HbWpfJf7 Presentation about HepData Converter and associated topics made at CERN on 21st September 2015

6.2 Code

All the code written as part of this project is inside of the following repositories:

1. https://github.com/HEPData/HEPData-converter Main repository for HEPData-converter
2. https://github.com/HEPData/HEPData-converter-docker Docker repository for HEPData-converter

All the new HEPData code is available from the HEPData organisation on GitHub: https://github.com/HEPData/