

Communication

ChEMBL Beaker: A Lightweight Web Framework Providing Robust and Extensible Cheminformatics Services

Michal Nowotka, Mark Davies, George Papadatos and John P. Overington *

European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton CB10 1SD, UK; E-Mails: mnowotka@ebi.ac.uk (M.N.); mdavies@ebi.ac.uk (M.D.); georgep@ebi.ac.uk (G.P.)

* Author to whom correspondence should be addressed; E-Mail: jpo@ebi.ac.uk; Tel.: +44-(0)-1223-492-666; Fax: +44-(0)-1223-494-468.

External Editor: Luc Patiny

Received: 18 August 2014; in revised form: 29 October 2014 / Accepted: 5 November 2014 /

Published: 17 November 2014

Abstract: ChEMBL Beaker is an open source web framework, exposing a versatile chemistry-focused API (Application Programming Interface) to support the development of new cheminformatics applications. This paper describes the current functionality offered by Beaker and outlines the future technology roadmap.

Keywords: web services; framework; server; REST; API; open source

1. Introduction

In recent years, applications based on a client-server architecture have gained in popularity. In a typical workflow, a user interacts with a client-side interface *via* a web browser or native application running locally on the user's computer or handheld device. The client-side forwards the processing task to a remote server, which then carries out the potentially computationally heavy task and returns results to the client-side interface.

Due to the wide range of available server solutions, software developers are often faced with the challenge of identifying the most suitable software stack, to expose methods and services of the server to the client-side interface. This important design decision may require tradeoffs between efficiency, flexibility and the number of available features offered by the software library. The lack of widely accepted and general standards in this area means that developers have to decide on the software stack

each time they create a new application. Such a decision requires a broad knowledge of existing solutions and can take a lot of time, which may lead to a significant extension in the development time of the application. This issue is especially true in the field of computational chemistry, where chemical software libraries are often dependent on a complex hierarchy of low-level, operating system libraries and third-party software packages [1–3]. The technical challenge of installing chemical software libraries and their dependencies can prevent installation in local and production environments [4].

To remediate this, Beaker provides access to such libraries utilising the previously described client-server software model. The immediate benefit to both users and administrators is that Beaker is installed once and is then available to all applications, which have network access to the Beaker host. Beaker provides access to the most common functionality required by chemistry-oriented applications. The growing set of methods currently includes several molecular formats inter-conversion, fingerprint and descriptor calculation, 2D and 3D compound depictions, InChI [5] and InChI Key generation, as well as optical structure recognition. By design, all the core methods exposed by Beaker manipulate data provided by the user, *i.e.*, there is no interaction with any chemical database or any other persistence layer apart from an optional caching mechanism.

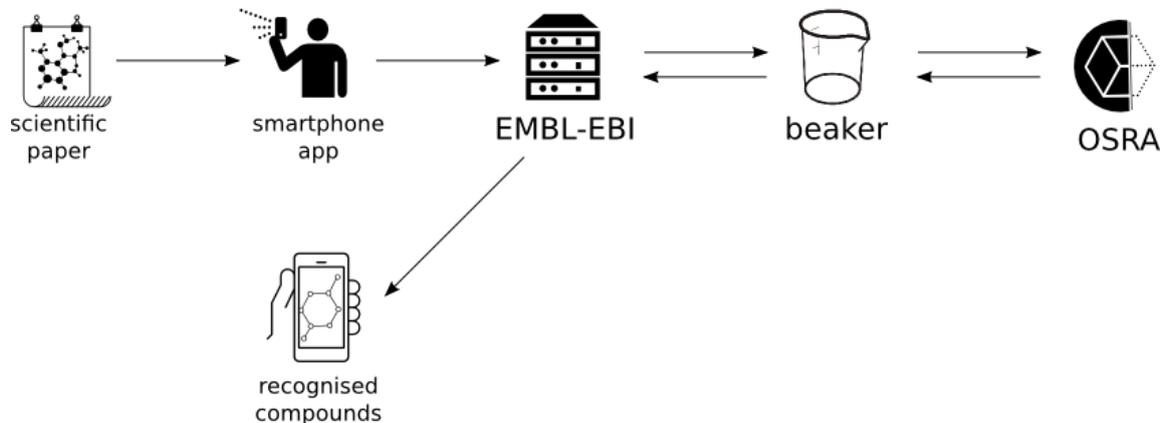
Beaker is written in the Python programming language. As an interpreted, dynamically typed scripting language, Python quickly became one of the most widely used scripting languages and a foundation of many successful small- and large-scale web frameworks. Moreover, Python has become a very popular language in the fields of scientific computing, data analysis and cheminformatics. The reason for Python's popularity in such fields is primarily due to the many new and powerful libraries it has at its disposal, *e.g.*, scikit-learn, pandas, matplotlib and RDKit [6]. The choice of RDKit, as the primary cheminformatics library used by Beaker, was due to the authors' familiarity and the excellent support provided by the RDKit community.

Beaker depends on a small number of software libraries, each of which is a *de facto* standard in the web server field. This makes it extremely easy to install and deploy Beaker, even in the most demanding of environments, which may include small embedded devices, such as Raspberry Pi [7] as well as large multi-core workstations. The list of Beaker software dependencies has been carefully chosen, in order to make the installation process as simple and robust as possible. All Beaker dependencies are open source and the core chemical components are RDKit and OSRA [8]. The ChEMBL group [9] also host a secure public instance of Beaker [10], which means the service does not even require a local installation.

Moreover, the extension of functionality offered by Beaker is also possible due its modular design. This allows developers to easily create and expose new methods, without depending on complicated software installations.

Beaker exposes all of its services *via* a RESTful API [11]. The simplicity and popularity of REST allows Beaker to be used as a backend for web-based widgets, mobile applications (Figure 1) and lightweight desktop tools. For example, Beaker can be used as a replacement web service backend for the ChemAxon MarvinJS chemical structure editor [12]. Finally, Beaker provides live, online documentation, generated automatically from the Python codebase. The documentation follows the SPORE specification [13] and provides interactive examples for each method exposed by Beaker.

Figure 1. An example of Beaker-based mobile application. A user takes a photo of small molecule using a mobile device application. The mobile application sends a chemical image to Beaker RESTful API, which then converts the image to a standard chemical structure format.



2. Results and Discussion

Beaker offers the following benefits both to the end user and the cheminformatics developer:

- **Ease of Use**—Access RDKit and OSRA *via* a documented RESTful API.
- **Application Development**—Beaker can speed up the process of creating new chemistry oriented applications.
- **Simple Installation**—All software dependencies are reduced to a minimum, making the Beaker installation trivially simple in most cases.
- **Speed**—Beaker uses the Tornado asynchronous networking library [14], which results in very efficient network access.
- **Security**—Beaker provides control over cross-origin policy by supporting CORS [15] to guard against cross-site request forgery attacks. It also provides throttling module to prevent DDOS attacks and offers IP black/white listing and a configurable limit on maximum uploaded file size. GET methods use URL-safe base64 parameter encoding, which prevents the injection of malicious content in URLs processed by Beaker.
- **Flexibility**—Beaker functionality can be easily extended by writing custom modules.
- **Configurability**—Fully configurable web server deployment.
- **Portability**—Beaker can be installed on any platform with an available Python interpreter.
- **No Costs**—Beaker is an open source software project, which uses free and open source tools and libraries.

The current release of Beaker (ver. 0.5.34), provides users with 36 distinct methods and each method can be accessed by a GET or POST request. When compared to similar services provided by other groups, such as ChemSpider [16] and National Cancer Institute [17], Beaker provides a greater number of methods and range of functionality. Another advantage Beaker offers over many equivalent services is that it is an open source project and its source code is available online, which means it does not “blackbox” any of its functionality from end users. The source code for Beaker is also registered in

the Python Package Index (PyPI), making it possible to easily install a local instance of Beaker. This means that Beaker can be run securely behind a company's firewall, which is often a very important requirement due to a companies' proprietary nature of chemical structures.

3. Conclusions

In conclusion, Beaker can help streamline the development of chemistry-focused applications, by centralising the deployment of chemical software libraries and providing access *via* a language agnostic RESTful interface. Future versions of Beaker will introduce the concept of a "chemistry-backend". This will allow a developer to explicitly choose the chemistry toolkit (e.g., RDKit, Indigo [18] or OpenBabel [19]) for a particular operation, such as structure image rendering. Furthermore, scalability will be addressed in future Beaker releases. This will be achieved by taking advantage of the Python multiprocessing [20] package, which will allow the contents of the file to be processed in parallel, significantly speeding up the SD [21] file processing time. Finally, it is hoped that the community of cheminformatics' developers and users will benefit from this new resource and contribute their own enhancements and use cases.

Acknowledgments

We acknowledge the following people, projects and communities, without whom Beaker would not have been possible:

1. Greg Landrum and the RDKit community (<http://www.rdkit.org/>)
2. Igor Filippov and OSRA software (<http://cactus.nci.nih.gov/osra/>)
3. Tornado web server and networking library (<http://www.tornadoweb.org/en/stable/>)
4. Bottle web framework (<http://bottlepy.org/docs/dev/index.html>)
5. Icons designed by Mister Pixel, Luis Prado, Eddy Ymeri, Zoe Austin, Jane Terekhov Vector from the thenounproject.com

Funding: Strategic Award for Chemogenomics from the Wellcome Trust [WT086151/Z/08/Z]; Member States of European Molecular Biology Laboratory.

Author Contributions

Michał Nowotka designed and implemented the Beaker service. Mark Davies and George Papadatos have built tools, which use the Beaker service and have provided valuable feedback to Michał Nowotka. Mark Davies, George Papadatos and John P. Overington have all contributed to discussions regarding the development plans of Beaker.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. List of OSRA Software Dependencies. Available online: <http://cactus.nci.nih.gov/osra/#2> (accessed on 29 October 2014).
2. List of RDKit Software Dependencies. Available online: <http://www.rdkit.org/docs/Install.html#installing-prerequisites-as-packages> (accessed on 29 October 2014).
3. List of OpenBabel Software Dependencies. Available online: <http://open-babel.readthedocs.org/en/latest/Installation/install.html#requirements> (accessed on 18 August 2014).
4. Installation Related Questions Asked on Rdkit-Discuss Mailing List. Available online: <http://sourceforge.net/p/rdkit/mailman/search/?q=install> (accessed on 29 October 2014).
5. The IUPAC International Chemical Identifier (InChI). Available online: <http://www.iupac.org/home/publications/e-resources/inchi.html> (accessed on 29 October 2014).
6. RDKit: Cheminformatics and Machine Learning Software. Available online: Available online: <http://www.rdkit.org/> (accessed on 29 October 2014).
7. Blog Post Describing Installation of ChEMBL Web Services on Raspberry Pi. Available online: <http://chembl.blogspot.jp/2013/10/tastypie-on-chempi.html> (accessed on 29 October 2014).
8. OSRA: Optical Structure Recognition Application. Available online: <http://cactus.nci.nih.gov/osra/> (accessed on 29 October 2014).
9. Bento, A.P.; Gaulton, A.; Hersey, A.; Bellis, L.J.; Chambers, J.; Davies, M.; Krueger, F.A.; Light, Y.; Mak, L.; McGlinchey, S.; *et al.* The ChEMBL bioactivity database: An update. *Nucl. Acids Res. Database Issue*. 2014, 42, D1083–D1090.
10. Public Live Instance of Beaker Software Provided by EBI. Available online: <https://www.ebi.ac.uk/chembl/api/utils/docs> (accessed on 29 October 2014).
11. Fielding, R.T.; Taylor, R.N. Principled design of the modern web architecture. In *ACM Transactions on Internet Technology (TOIT)*; Association for Computing Machinery: New York, NY, USA, 2002; pp. 115–150.
12. Marvin 4 JS Web Services Specification. Available online: <https://marvinjs-demo.chemaxon.com/latest/docs/dev/webservices.html> (accessed on 29 October 2014).
13. SPORE—Specification to a POrtable Rest Environment. Available online: <https://github.com/SPORE/specifications> (accessed on 29 October 2014).
14. Tornado: Facebook's Real-Time Web Framework for Python. Available online: <https://developers.facebook.com/blog/post/301> (accessed on 29 October 2014).
15. Cross-Origin Resource Sharing, W3C Recommendation 16 January 2014. Available online: <http://www.w3.org/TR/access-control/> (accessed on 29 October 2014).
16. ChemSpider Web Services Main Web Page. Available online: <http://www.chemspider.com/AboutServices.aspx> (accessed on 29 October 2014).
17. National Cancer Institute, Chemical Identifier Resolver. Available online: <http://cactus.nci.nih.gov/chemical/structure> (accessed on 29 October 2014).
18. Indigo: Universal Organic Chemistry Toolkit. Available online: <http://www.ggasoftware.com/opensource/indigo> (accessed on 29 October 2014).
19. Open Babel: The Open Source Chemistry Toolbox. Available online: http://openbabel.org/wiki/Main_Page (accessed on 29 October 2014).

20. Multiprocessing—Process-Based “Threading” Interface. Available online: <https://docs.python.org/2/library/multiprocessing.html> (accessed on 29 October 2014).
21. Chemical Files Format Specifications. Available online: <http://download.accelrys.com/freeware/ctfile-formats/ctfile-formats.zip> (accessed on 29 October 2014).

Availability

Beaker source code can be accessed online using the following link: https://github.com/mnowotka/chembl_beaker. It can be installed directly *via* python setup tools by invoking the following command: “pip install chembl_beaker”. ChEMBL group provides public instance of Beaker software available directly using following URL: <https://wwwdev.ebi.ac.uk/chembl/api/utis/docs> or using python client library called “chembl_webresource_client” available in PyPI and GitHub: https://github.com/chembl/chembl_webresource_client.

© 2014 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).