LALA: A Light Application-Level API

Seth Carbon*

17th Bioinformatics Open Source Conference (BOSC) 2017, Prague, Czech Republic

Website: https://github.com/kltm/lala
Repository: https://github.com/kltm/lala
License: CC-BY 4.0

Much thought and paper has been put into sharing information between resources, from the use of common identifiers to use of common data stores. We would like to explore an area that has not, to the authors' knowledge, been as well explored: the sharing of simple information directly between web-based applications, without the use of a server and central API as intermediary. For example, a curator may wish to import IDs from one tool to another, curate the same paper using different tools, or a researcher may wish to analyze the same piece of genome in different analysis tools.

More concretely, the use case that we wish to look at is how to obtain small packets of specific information from an external resource that has its own associated web application; we look to do this by round-tripping a packet of information (acting as a "black box" or "piggy bank") through the external application using basic HTTP methods, allowing an easy high-level "federation". In an environment where much effort is being spent on the creation of rich web applications tailored to the habits of specific groups of scientists and curators, users and engineers would have much to gain by being able to reuse functionality from external applications in their workflows and stacks, as well as having the added benefit of driving traffic to external web applications that implement such a common specification.

In creating this protocol, we want to captures the following qualities:

- Easy to implement
  Complexity can be a barrier to adoption and implementation
- Basic HTTP tooling
  Any system should have easy access to the tools necessary
- "Stateless"
  Simplifying debugging and implementation
- Minimal need for initial coordination
  Beyond what data is to be returned, the external application does not need to understand the transiting packet
- No need for calling application to coordinate changes to own API after initial coordination
  As the calling application is responsible for decoding the information it initially sent and the location it is sent to, major API changes can occur without the need to coordinate with any other resource
- Have the ability to perform operations "remotely" while still logged-in to the calling application, without the need to coordinate cross-site logins
  This can be done by placing an authorization token in the transiting packet

Taking inspiration from the methods used by Galaxy [1] to pull data in from external applications, we'd like to propose (and get feedback) about a general pattern for passing relatively simple data directly between applications. Variations of this proposal have been implemented or explored in applications such as: Noctua, Textpresso Central, AmiGO, and PubAnnotation.

*Berkeley Bioinformatics Open-source Projects, Lawrence Berkeley National Lab, Berkeley, CA, USA.

References

[1] The Galaxy platform for accessible, reproducible and collaborative biomedical analyses, 2016 update *Nucleic Acids Research* 44(W1): W3-W10, doi:10.1093/nar/gkw343