Inter-Parameter Dependencies in Real-World Web APIs: The IDEA Dataset

Alberto Martin-Lopez, Sergio Segura, and Antonio Ruiz-Cortés

Universidad de Sevilla, Spain

{amarlop,sergiosegura,aruiz}@us.es

Abstract. **Context**: Web services often impose constraints that restrict the way in which two or more input parameters can be combined to form valid calls to the service, so-called *inter-parameter dependencies*. Current API design languages like the OpenAPI Specification (OAS) provide no support for their formal description, making it hardly possible to automatically discover and interact with services without human intervention. Researchers and practitioners are openly requesting support for modelling and validating inter-parameter dependencies in web APIs, but this is not possible unless we share a deep understanding of how these dependencies emerge in practice. **Objective**: We aim to provide evidence on how inter-parameter dependencies are used in real-world web APIs. This evidence will hopefully serve as a basis for future proposals for modelling and analysing inter-parameter dependencies and will open a new range of research possibilities in areas related to service-oriented computing. **Method**: The documentation of 2,557 operations from 40 real-world web APIs was reviewed and carefully analysed, and 633 inter-parameter dependencies were found and classified into seven different types. **Results**: A machine-readable dataset was generated. This dataset helps understand the dimension and recurrence of inter-parameter dependencies in web APIs, as well as their taxonomy.

Keywords: input constraint, dependency constraint, RESTful API.

Background & Summary

Web services are heavily used nowadays for integrating heterogeneous software systems. In this context, many contributions have been made to automate several processes such as service design, discovery, composition and testing. However, such approaches still suffer to a large extent from one fundamental problem: the management of inter-parameter dependencies. An inter-parameter dependency imposes a constraint on how two or more input parameters can be combined to form valid calls to the service. For example, in the YouTube API, when searching for videos in high definition (parameter videoDefinition), the parameter type must be set to 'video', otherwise an HTTP 400 status code ("bad request") is returned in the response. Unfortunately, current API specification languages such as the OpenAPI Specification (OAS) [3] or the RESTful API Modeling Language (RAML) [5] provide no support for the formal description of such dependencies. The interest of industry in having support for these is reflected in an open feature request in OAS entitled "Support interdependencies between query parameters" [2], created in January 2015 with the message shown below. At the time of this writing, the request has received over 280 votes and 55 comments from 33 participants:

"It would be great to be able to specify interdependencies between query parameters. In my app, some query parameters become 'required' only when some other query parameter is present. And when conditionally required parameters are missing when the conditions are met, the API fails. Of course I can have the API reply back that some required parameter is missing, but it would be great to have that built into Swagger."

This feature request has fostered an interesting discussion where the participants have proposed different ways of extending OAS to support dependencies among input parameters. However, each approach aims to address a particular type of dependency and thus show a very limited scope. Addressing the problem of modelling and validating inter-parameter dependencies in web APIs should necessarily start by understanding how dependencies emerge in practice. To address this problem, in a previous paper [10], we performed a thorough study on the presence of inter-parameter dependencies in industrial web APIs. Our study is based on an exhaustive review of 2,557 operations from 40 APIs selected among 10 different application domains. As a result, we managed to generate a dataset containing all inter-parameter dependencies found (633 in total) classified into seven general types.

This paper presents the IDEA dataset, generated as a part of our previous survey on the presence of inter-parameter dependencies in web APIs [10]. Among other results, we found that inter-parameter dependencies are extremely common and pervasive: they appear in 4 out of every 5 APIs across all application domains and types of operations. Researchers and practitioners can leverage the IDEA dataset for multiple purposes. For instance, proposals for modelling and validating dependencies may benefit from the results obtained in our survey. Furthermore, new approaches for web services testing and discovery may involve the management of inter-parameter dependencies to achieve a higher degree of automation.

Dataset Specification

Subject	Web Application Programming Interfaces (APIs).
Specific subject area	Specification of web APIs.
Type of data	Spreadsheets.
How data were acquired	Survey comprising 2,557 operations from 40 real-world web APIs. The online documentation of each API was carefully read and analysed by at least two people to reduce misunderstanding. *Instruments used*: Standard PC with Internet access for reviewing the documentation of the APIs.
Data format	Analysed.

Parameters for data collection	Only APIs complying with level 1 of the Richardson Maturity Model (i.e., "REST-like APIs") [6,12] were considered. A total of 2,557 operations (including read, write, update and delete operations) from 40 APIs were selected from the ProgrammableWeb API repository [4], including the 10 most popular APIs and the 3 most popular APIs from the 10 most popular categories.
Description of data collection	Inter-parameter dependencies were identified in two steps. First, the online documentation of each API was carefully read and analysed, recording all dependencies found. This was done by at least two authors per API, in order to reduce mistakes. In a second step, the shape of all dependencies was studied, and they were classified into seven general patterns.
Data source location	Institution: Applied Software Engineering (ISA) research group. University of Seville. City: Seville. Country: Spain.
Data accessibility	Repository name: IDEA Dataset: Inter-parameter DEpendencies in web Apis. Direct URL to data: https://bit.ly/35aqBBS
Related research article	A. Martin-Lopez, S. Segura, A. Ruiz-Cortés. A Catalogue of Inter-Parameter Dependencies in RESTful Web APIs. International Conference on Service-Oriented Computing (2019). https://doi.org/10.1007/978-3-030-33702-5_31

Value of the Data

- The IDEA dataset is key for the correct understanding of how inter-parameter dependencies are used in practice. It contains samples of all possible types of dependencies for which practitioners are asking support in OAS (and more). It is also machine-readable.

- Both researchers and practitioners can benefit from these data. Researchers on service-oriented architectures may leverage the dataset to automate the analysis of dependencies in web APIs. Practitioners may be interested to integrate the specification and analysis of dependencies in current standards such as OAS [2,3] and RAML [5].

- The IDEA dataset offers ample research opportunities in the field of service-oriented computing. Proposals for the specification and automated analysis of dependencies can be tackled. The fact that it is machine-readable can help automate processes such as web service discovery and composition.

- The IDEA dataset has already proved useful in some contexts. For example, we found that the misspecification of dependencies is a common problem in web APIs [8,9]. Furthermore, the coverage of valid and invalid combinations of input parameters could be considered a new test coverage criterion for web APIs [11].

Data Description

The dataset is available online as supplementary material of our survey [1]. It consists of a Google spreadsheet containing 67 tabs, described next:

- The first tab ("Home") includes information regarding the most relevant tabs in the spreadsheet.

- The second tab ("Subjects APIs full analysis") contains information about the 40 APIs analysed, including the name, category, URL to the documentation and number and percentage of operations with and without dependencies classified by type (i.e., their CRUD semantic), among others.

- The third tab ("Selection process") provides a summary of the selection process followed for filtering the APIs from ProgrammableWeb [4].

- Tabs 4 to 12 include statistics and charts regarding the size (number of operations), category and number of dependencies of the subject APIs.

- Tabs 13 to 27 include statistics and charts regarding the type and shape of all the dependencies found, such as the number and type of parameters involved (e.g., query and header parameters).

- Tabs 28 to 67 include details about all inter-parameter dependencies found in every API under study (40 tabs in total, one per API).

Experimental Design, Materials, and Methods

The subject APIs were selected from the popular API repository ProgrammableWeb [4]. More specifically, we selected the top 10 most popular APIs and the 3 top-ranked APIs from the 10 most popular categories. We selected APIs reaching level 1 or higher in the Richardson Maturity Model [6,12], which ensured a minimal adherence to the REST architectural style, the current de facto standard for web API design. Overall, we selected 40 APIs distributed among 10 different categories such as communication, social and mapping, including web APIs with millions of

users worldwide such as Google Maps and PayPal. Regarding the size, the majority of reviewed APIs (75%) provide between 1 and 50 operations, with the largest APIs having up to 305 (DocuSign eSignature) and 492 (GitHub) operations.

Dependencies were identified in two steps. First, we recorded all the inter-parameter dependencies found in the documentation of the subject APIs. It is worth mentioning that every dependency can be represented in multiple ways, e.g., in conjunctive normal form. At this point, we strove to represent them as they were described in the documentation of the API. This allowed us, for example, to record the *arity* of each dependency, i.e., number of parameters involved in each constraint. In a second step, we studied the shape of all the dependencies and managed to group them into seven general dependency types. The documentation collected from each API was reviewed by at least two different authors to reduce misunderstanding or missing information.

Acknowledgments

This work has been partially supported by the European Commission (FEDER) and the Spanish Government under projects HORATIO (RTI2018-101204-B-C21), APOLO (US-1264651) and EKIPMENT-PLUS (P18-FR-2895), and the FPU scholarship program, granted by the Spanish Ministry of Education and Vocational Training (FPU17/04077).

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships which have, or could be perceived to have, influenced the work reported in this article.

References

[1] [dataset] A. Martin-Lopez, S. Segura, A. Ruiz-Cortés, IDEA Dataset: Inter-parameter DEpendencies in web Apis, v1, 2020. https://bit.ly/35aqBBS

[2] Feature request in OAS: “Support interdependencies between query parameters”. https://github.com/OAI/OpenAPI-Specification/issues/256, 2015 (accessed April 2020).

[3] OpenAPI Specification. https://www.openapis.org/, 2020 (accessed April 2020).

[4] ProgrammableWeb API repository. https://www.programmableweb.com/, 2020 (accessed April 2020).

[5] RAML: RESTful API Modeling Language. https://raml.org/ (accesed April 2020).

[6] Richardson Maturity Model. https://martinfowler.com/articles/richardsonMaturityModel.html, 2010 (accessed April 2020).

[7] Swagger. https://swagger.io/, 2020 (accessed April 2020).

[8] A. Martin-Lopez, AI-Driven Web API Testing, International Conference on Software Engineering, Doctoral Symposium (2020). To appear.

[9] A. Martin-Lopez, Automated Analysis of Inter-Parameter Dependencies in Web APIs, International Conference on Software Engineering, ACM Student Research Competition (2020). To appear.

[10] A. Martin-Lopez, S. Segura, A. Ruiz-Cortés, A Catalogue of Inter-Parameter Dependencies in RESTful Web APIs, International Conference on Service-Oriented Computing (2019), 399—414. https://doi.org/10.1007/978-3-030-33702-5_31

[11] A. Martin-Lopez, S. Segura, A. Ruiz-Cortés, Test Coverage Criteria for RESTful Web APIs, International Workshop on Automating TEST Case Design, Selection, and Evaluation (2019), 15—21. https://doi.org/10.1145/3340433.3342822

[12] L. Richardson, S. Ruby, M. Amundsen, RESTful Web APIs, first ed., O'Reilly Media, Inc., 2013.