

IUPAC Protocol Services Demo

WorldFAIR Chemistry

2023-10-07

WorldFAIR Chemistry: Protocol Services

Interactive Demonstration

This notebook is intended as an interactive demonstration of the services being proposed by the IUPAC WorldFAIR Chemistry D3.3 project team. A complete description of the project is available at: <https://iupac.github.io/WFChemProtocols/intro.html>.

This notebook is an RMarkdown version of the original Jupyter Notebook, which is available at: <https://github.com/IUPAC/WFChemProtocols/blob/main/IUPACProtocolsDemo.ipynb>

Resolver Summary

While more detail is provided in the documentation linked above, in short what is described here is a web service called a “resolver” that performs two main functions:

1. Check for the presence of a chemical record in the hosting organization’s database.
2. Validate the machine-readable chemical structure according to the hosting organization’s rules.

Resolver Base URL

The service being proposed in this project is a regular HTTP web service, using standard CGI URL syntax, and a well-defined data model for the information returned. This demonstration uses a prototype service hosted by PubChem, using JSON as the response format (although in principle it could be XML or any other structured data format).

One key point of this proposal is that the base URL for the resolver CGI would vary from one institution to another, but the inputs (CGI arguments) and outputs (JSON data) would be standard, the same for any organization implementing the service. So simply by switching the base URL, one can run the same query on multiple different sites, without otherwise needing to change any code.

In R, this could look like this:

```
library(httr)
library(rjson)
resolver_base_url <- "https://pubchem.ncbi.nlm.nih.gov/resolver/resolver.cgi"
```

When called without any arguments, the resolver will return some information about what inputs and outputs it can handle.

```
url <- resolver_base_url
res <- GET(URLencode(url))
url_data <- httr::content(res, type="text", encoding="UTF-8")
# display the results
url
```

```
## [1] "https://pubchem.ncbi.nlm.nih.gov/resolver/resolver.cgi"
url_data

## [1] "{\n  \"Result\": {\n    \"ServiceDetails\": {\n      {\n        \"Resource\": \"PubChem\", \n\n# put into object
r <- fromJSON(url_data)
r

## $Result
## $Result$ServiceDetails
## $Result$ServiceDetails[[1]]
## $Result$ServiceDetails[[1]]$Resource
## [1] "PubChem"
##
## $Result$ServiceDetails[[1]]$ResourceURL
## [1] "https://pubchem.ncbi.nlm.nih.gov"
##
## $Result$ServiceDetails[[1]]$ResolverURL
## [1] "https://pubchem.ncbi.nlm.nih.gov/resolver/resolver.cgi"
##
## $Result$ServiceDetails[[1]]$AvailableInputs
## $Result$ServiceDetails[[1]]$AvailableInputs$SDF
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableInputs$SMILES
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableInputs$InChI
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableInputs$InChIKey
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableInputs$PNG
## [1] FALSE
##
## $Result$ServiceDetails[[1]]$AvailableInputs$Name
## [1] TRUE
##
##
## $Result$ServiceDetails[[1]]$AvailableOutputs
## $Result$ServiceDetails[[1]]$AvailableOutputs$IUPACName
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableOutputs$SMILES
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableOutputs$InChI
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableOutputs$InChIKey
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableOutputs$ResourceIdentifier
```

```
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableOutputs$RecordURL
## [1] TRUE
##
## $Result$ServiceDetails[[1]]$AvailableOutputs$ImageUrl
## [1] TRUE
```