

Bacatá: Notebooks for DSLs, Almost for Free

Mauricio Verano Merino, Jurgen Vinju, and Tijs van der Storm

Objective

Define and generate a notebook for a Domain-Specific Language (DSL) at the language abstraction level and not at the tool implementation level.

- Open up the interactive notebook metaphor for DSLs.
- Extend the current set of generated IDE services of language workbenches.
- Generate notebooks for DSLs with minimum effort.

Bacatá

Bacatá is a language-parametric interface between Jupyter and LWBs. It provides a mechanism to generate language kernels by reusing existing language components such as grammars, parsers, type checkers, code generators, and interpreters. Bacatá also supports the generation of language-specific components (e.g., syntax highlighters and completors).

Architecture

Bacatá has three components, Bacatá-Core, Bacatá-Rascal, and the ILanguageProtocol. Bacatá-Core enables the communication between Jupyter and a language workbench, and Bacatá-Rascal connects Rascal-based languages to Bacatá-Core. The ILanguageProtocol is a generic language protocol interface (comparable to the Language Server Protocol).

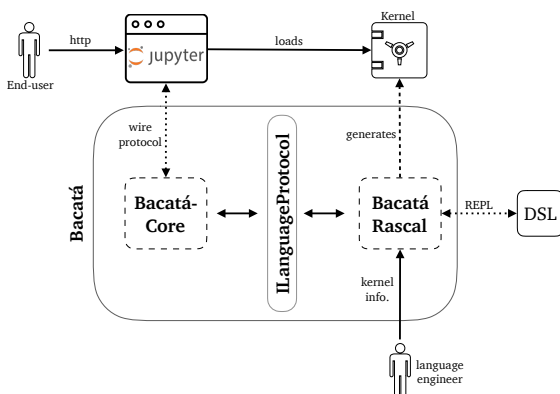


Figure 1: Bacatá's architecture overview.

How to create a notebook for a DSL?

To create a notebook for a DSL using Bacatá, language engineers need to follow the next three steps:

- Define a Read-Eval-Print-Loop (REPL).
- Create a Kernel data type.
- Call the Bacatá function to install the language kernel, and to start the Jupyter server.

Case studies

We have used Bacatá to generate notebook interfaces for different languages, namely, *Halide**, *SweeterJS*, and *QL*.

```

In [1]: 1 form myForm = taxOfficeExample {
2       "Did you buy a house in 2010?"
3       hasBoughtHouse: boolean
4
5       "Did you enter a loan?"
6       hasMaintLoan: boolean
7
8       "Did you sell a house in 2010?"
9       hasSoldHouse: boolean
10
11      if (hasSoldHouse) {
12          "What was the selling price?"
13          sellingPrice: money
14          "Private debts for the sold house:"
15          privateDebt: money
16          "Value residue:"
17          valueResidue: money = sellingPrice
18              - privateDebt
19      }
20 }

Out[1]: ok

In [2]: 1 html(myForm)

Out[2]: Form: taxOfficeExample
Did you buy a house in 2010? true/false
Did you enter a loan? true/false
Did you sell a house in 2010? true/false
What was the selling price? 243,000.00
Private debts for the sold house: 39,250.00
Value residue: 203,750.00
Submit

In [3]: 1 visualize(myForm)

Out[3]: Visualization
privateDebt --> hasSoldHouse
sellingPrice --> hasSoldHouse
valueResidue --> hasSoldHouse

```

Figure 2: QL notebook with a tax questionnaire.