# Goupile: A New Paradigm for the Development and Implementation of Clinical Report Forms.

Niels Martignene, Ali Amad, Julie Bellet, Julien Tabareau, Fabien D'Hondt, Thomas Fovet, Antoine Lamer

▶ **To cite this version:**

Niels Martignene, Ali Amad, Julie Bellet, Julien Tabareau, Fabien D'Hondt, et al.. Goupile: A New Paradigm for the Development and Implementation of Clinical Report Forms.. Studies in Health Technology and Informatics, 2022, Studies in Health Technology and Informatics, 294, pp.540-544. 10.3233/SHTI220517 . hal-04338285

**HAL Id: hal-04338285**
**https://hal.univ-lille.fr/hal-04338285**

Submitted on 15 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

*Challenges of Trustable AI and Added-Value on Health*
*B. Séroussi et al. (Eds.)*
*© 2022 European Federation for Medical Informatics (EFMI) and IOS Press.*
*This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).*
*doi:10.3233/SHTI220517*

# Goupile: A New Paradigm for the Development and Implementation of Clinical Report Forms

Niels MARTIGNENE[a,b,1], Ali AMAD[c], Julie BELLET[d], Julien TABAREAU[d], Fabien D'HONDT[b,c], Thomas FOVET[b,c] and Antoine LAMER[a,e]
[a]*InterHop, F-59000 Lille, France*
[b]*Centre National de Ressources et de Résilience Lille-Paris (CN2R), Lille, France*
[c]*Univ. Lille, Inserm, CHU Lille, U1172 – LilNcog - Lille Neuroscience & Cognition, F-59000 Lille, France*
[d]*CHU de Lille, Pôle d'anesthésie-réanimation, F-59000 Lille, France*
[e]*Univ. Lille, CHU Lille, ULR 2694 - METRICS: Évaluation des Technologies de santé et des Pratiques médicales, F-59000 Lille, France*

**Abstract.** Despite the increasing computerization of hospital information systems, segments of patient care are still in paper format. Data extracted automatically from the hospital databases for one specific project are thus supplemented by data collected manually. Data collection tools are usually developed entirely, which requires computer knowledge and is tedious, or automatically from metadata or drag and drop controls, which is limiting in terms of functionality. To facilitate this manual collection, we developed a free and open-source tool for creating forms that does not require advanced computer skills, offers rich features, and is quickly implemented, tested and deployed. It was implemented for 15 projects and supported thousands of daily users for a complex interactive study at the national level.

**Keywords.** Clinical research; Electronic data capture; Electronic health record; Data collection

## 1. Introduction

Despite the increasing computerization of health facilities, health data are still being collected manually to supplement the computerized databases when conducted observational retrospective or prospective studies. There are three main types of tools for manually capturing and storing data: spreadsheets, Electronic Data Capture (EDC) systems and handmade applications.

Spreadsheets are part of office suite software. In the same manner as text editors or presentation programs, spreadsheets are prevalent because they are automatically installed on the workstation and easy to handle [1]. However, they can be modified by a

[1] Corresponding Author, Niels Martignène, CN2R, 103 Boulevard de la Liberté, F-59000, Lille, France; E-mail: niels.martignene@protonmail.com.

faulty manipulation, they do not support the generation of standardized and interoperable data, and it is particularly difficult to manually handle for large datasets with high number of rows and columns.

EDCs collect data in an electronic Clinical Report Form (eCRF) and are generally composed of a graphical user interface component for data entry, a validation component and a database [2]. There are two main ways to design a form with an EDC: a metadata-driven approach and a GUI approach. In the first case, a table with the list of required variables, types, labels and constraints are loaded into the EDC to automatically structure the visual interface and the database. In the second case, graphical widgets are manually positioned on the interface by drag and drop. The widgets are then configured to take into account the characteristics of each variable. Although they have been reported as a mean of improving research efficiency [3], both approaches often offer limited functionality, and proprietary license is a barrier for scientific projects without sufficient funding [4].

The last alternative consists in developing handmade applications from scratch. For this, a web technology (such as PHP, Node.js, etc.) is generally associated with a database management system, which control the user interface and data storage, respectively. This approach allows to develop forms with unlimited functionalities, but requires advanced programming skills and is time consuming.

The objective of this project is to provide a free and open-source tool facilitating the developer and end-user work during the eCRF design stage. The tool has to allow creating forms on the fly, immediately ready to be tested and used, with a minimum of code and coding skills.

## 2. Methods

We performed extensive interviews among our peers in medical, pharmaceutical and environmental sciences to assess their needs in term of data capture tools. To bridge the gap between spreadsheets, EDCs and handmade applications, our solution had to answer the following design goals: (i) enable the creation of the project, the design of the interface, and the formatting of the data structure and storage with a minimum of code; (ii) facilitate the development and interactions between developer and end-user, (iii) support standard widgets but also rich functionalities, (iv) view and export the data easily, (v) provide training, (vi) generate shareable forms, (vii) support an offline mode and mobile/tablet format, and (viii) be free and open-source.

To bridge the gap between existing approaches, we have developed a framework which supports the generation the interface, data structure, and storage in one line using a common programming language rather than a meta-language or a graphical library.

We have developed Goupile project using free and open-source technologies. The front-end is built using HTML/CSS/Javascript, while the back-end is implemented in C++. Data is stored on the server as JSON (JavaScript Object Notation) objects in a SQLite database. The choice of SQLite was made for three reasons: (i) we wanted to compile and use Goupile as an autonomous single-file binary, (ii) we use relational tables for other functionality, such as user management, and (iii) SQLite is a solid battle-tested database engine with an extensive test suite. Each record does not necessarily have the same variables, which allows the form to evolve and store new variables. Some data can be stored client-side with IndexedDB (offline mode).

The standard widgets are implemented with predefined Javascripts functions: number, text, dropdown list, radio button, checkbox, date. Other more specific widgets were developed according to user needs: choice (a set of horizontally arranged buttons), binary (choice with yes/no answer), likert scale, and computed variables. All widgets are configurable with optional arguments dealing with the mandatory of the field, the number of decimals, or the minimal and maximal values, for example. Optional arguments are passed to the function between curly braces (see Figure 1).

```
function ( "variable_name", "Label shown to user", {
    option1 : value,
    option2 : value
} )
```

**Figure 1.** Function template for the creation of a widget

## 3. Results

Goupile is composed of five main components:

- an administration panel to create projects, users and assign projects to users in defining permissions (code, entry, export, read what was read by other),
- an application editor to define the architecture of a project (the forms and the relationships between them) and its general configuration (date of start, style with a CSS sheet),
- a form editor to code the sections and widgets (label, type, options),
- an entry panel to display instantly the form and its visual widgets, which allows the user to enter data,
- a data overview module to display the data recorded and provide export function (in csv or xlsx formats), audit trail displays information on entry (user and date); a dashboard displays the number of records and number of complete cases.

The project is available in two formats, a ready-to-use package for a single form and a single user, and a complete solution for multiple forms and with different configuration options [5].

### *3.1. Development of the form*

During the form design phase, the form editor may be used simultaneously with the entry panel: while coding the widget in the editor, the control is immediately displayed and usable right next to it (see Figure 2). At this stage, it is already possible to validate a record, test the logical structure of the form and visualize fictitious records.

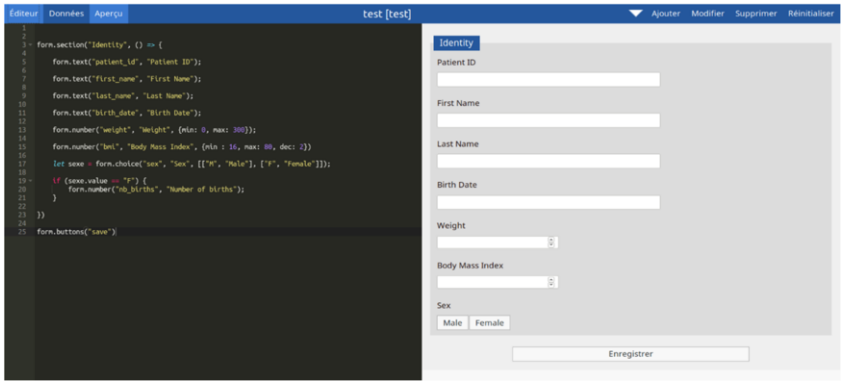


**Figure 2.** Form editor and entry panel. The left part of the interface is the form editor. The right part of the interface is the entry panel.

### *3.2. Daily use*

During the data collection phase, the entry panel may be used alone, or with the data overview/export part (see Figure 3).

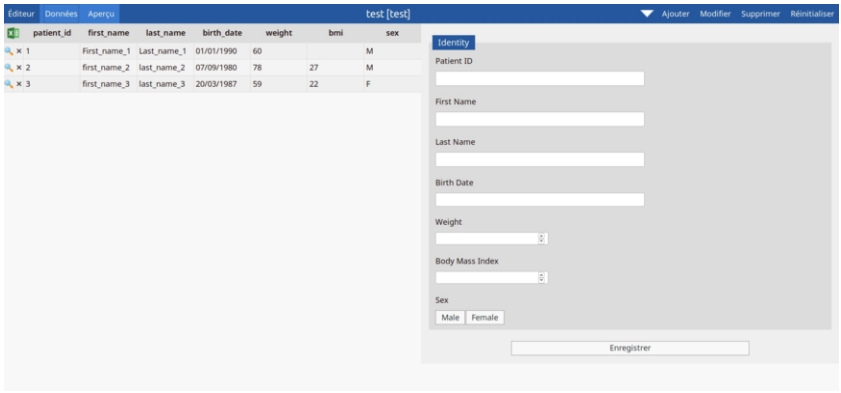


**Figure 3.** Data and form overview. The left part of the interface is the data overview. An export button allows exporting a CVS file. The right part of the interface is the entry panel.

### *3.3. Functionality*

The main features are the support of 1-to-many relationships, an offline mode, responsive design for tablets and smartphones, error management, data tracking. Advanced users can also customize the form with conditions, loops and all that can be programmed with Javascript. Finally, the data is hosted on a secure server.

### *3.4. Projects*

At this time, Goupile has been used for 15 projects, including 3 multi-center projects. The design of Goupile was flexible enough for us to implement the 17 modules of the MINI for DSM-V assessment online, despite the complex flow of the questions. We were also able to conduct a complex interactive study at the national level with Goupile, with

several neuropsychological assessments, where the users had to watch multiple videos and photos and answer to associated questions.

Several studies were developed on Goupile by non-programmers, who where able to do create most of the form each time. We typically intervened only at the end to fix minor mistakes and help to code the more complex conditions needed for some questions.

## 4. Discussion

We offer a tool that allows to quickly design and edit a form using ready-to-use Javascript functions, immediately visualize and test the form, and visualize/export collected data. The use of ready-made functions directly modifies the appearance of the form and the structure of the database, thus saving time compared to an application developed from scratch. These functions are developed in a common language (Javascript), rather than a meta-language or a GUI, which offers a number of customization possibilities for the eCRF. Finally, the hosting on a server, and the data quality rules are advantages compared to the spreadsheets. Goupile is provided free of charge, and the code is free and open-source (AGPL 3.0).

The tool was well received by the users, both for the form design and data collection phases. Further developments will address the support of direct mail, the use in clinical routine, and more powerful data monitoring capabilities.

## 5. Conclusions

Goupile presents a new paradigm for the quick development and implementation of an eCRF. Based on Javascript and ready-made functions, it allows users to easily design, edit, test and deploy an eCRF.

## References

[1] Anderson NR, Lee ES, Brockenbrough JS, Minie ME, Fuller S, Brinkley J, Tarczy-Hornoch P. Issues in biomedical research data management and analysis: needs and barriers. J Am Med Inform Assoc. 2007 Jul-Aug;14(4):478-88. doi: 10.1197/jamia.M2114. Epub 2007 Apr 25. PMID: 17460139; PMCID: PMC2244904.

[2] Shah J, Rajgor D, Pradhan S, McCready M, Zaveri A, Pietrobon R. Electronic data capture for registries and clinical trials in orthopaedic surgery: open source versus commercial systems. Clin Orthop Relat Res. 2010 Oct;468(10):2664-71. doi: 10.1007/s11999-010-1469-3. PMID: 20635174; PMCID: PMC3049639.

[3] Rorie DA, Flynn RWV, Grieve K, Doney A, Mackenzie I, MacDonald TM, Rogers A. Electronic case report forms and electronic data capture within clinical trials and pharmacoepidemiology. Br J Clin Pharmacol. 2017 Sep;83(9):1880-1895. doi: 10.1111/bcp.13285. Epub 2017 Apr 22. PMID: 28276585; PMCID: PMC5555865.

[4] Open-Source Electronic Health Record Systems for Low-Resource Settings: Systematic Review. JMIR Med Inform. 2017 Nov 13;5(4):e44. doi: 10.2196/medinform.8131. PMID: 29133283; PMCID: PMC5703976.

[5] (n.d.). https://framagit.org/interhop/goupile (accessed January 20, 2022).