Journal of open research software

CastorEDC API: A Python Package for Managing Real World Data in Castor Electronic Data Capture

SOFTWARE METAPAPER

REINIER CORNELIS ANTHONIUS VAN LINSCHOTEN
SEBASTIAAN LAURENS KNIJNENBURG
RACHEL LOUISE WEST**
DESIRÉE VAN NOORD**

*Author affiliations can be found in the back matter of this article
**These two authors share last authorship

ubiquity press

ABSTRACT

Real world data is being used increasingly in medical research. Castor Electronic Data Capture is a secure and user-friendly platform for managing study data. Integrating data from several databases into a single Castor database is complex. We developed CastorEDC API, a free and open source Python package which can be used to interact with the API of Castor, and through which data can be imported from multiple sources into a Castor database. The importer reads, cleans, validates and imports data while accounting for differences in column structure and variable coding between databases.

https://github.com/reiniervlinschoten/castoredc_api.

CORRESPONDING AUTHOR:
Reinier Cornelis Anthonius van Linschoten
Department of Gastroenterology & Hepatology, Franciscus Gasthuis & Vlietland, Rotterdam, Netherlands; Department of Gastroenterology & Hepatology, Erasmus MC, Rotterdam, Netherlands
r.vanlinschoten@erasmusmc.nl

KEYWORDS:
Big data; Data management; Castor EDC; Clinical research; Clinical trials; Real world data; Administrative data

TO CITE THIS ARTICLE:
van Linschoten RCA, Knijnenburg SL, West RL, van Noord D 2023 CastorEDC API: A Python Package for Managing Real World Data in Castor Electronic Data Capture. *Journal of Open Research Software*, 11: 12. DOI: https://doi.org/10.5334/jors.436

(1) OVERVIEW

INTRODUCTION

There is a paradigm shift in the collection of data for medical research [1, 2]. As manual data collection through case report forms is an expensive and time-consuming task [3], there is substantial interest in real world data (RWD) which is a rich source of information for scientific research. RWD is data on patient health status and/or delivery of healthcare which is routinely collected during care [4]. Using RWD for trials can reduce costs, but there are also scientific benefits, for example new data collection options and the possibility to answer research questions for which a trial was otherwise not feasible [5]. An important challenge for RWD however is the management, processing and merging of large and divergent datasets [5].

Castor Electronic Data Capture (EDC) is a cloud-based clinical data management platform allowing researchers to securely and efficiently manage their clinical research data [6]. While there are a few options for directly integrating RWD into a Castor database, there are limitations.

Importing simple RWD is possible using the user interface of Castor EDC, but this does not allow for import of survey data and does not account for differences in columns and variable coding between databases. Moreover, importing data through the user interface is limited to a maximum of 25.000 data points and cannot be used programmatically. It is possible to directly link the Castor database to a RWD database, such as an electronic health record [7], but this needs to be configured separately for each database.

Castor EDC offers a public HyperText Transfer Protocol Secure (HTTPS) Application Programming Interface (API) that conforms to the Representational State Transfer (REST) architecture style. This API can be used to securely interact with the study database and import data [8]. A prior Python project implemented some of the endpoints in the API, but has little support for differences between databases, a small testing suite and can only be used with considerable knowledge of Python [9]. For R there is an open source package available to interact with the API, but this package currently only supports reading data from the API, not writing data into the database [10].

We aimed to develop an open source Python package to make integration of RWD accessible for clinical researchers, that can be used with little knowledge of programming or Python and which can be easily integrated with other data science tools such as pandas or R.

IMPLEMENTATION AND ARCHITECTURE

The software package consists of three separate modules: the client (CastorClient), a local representation of the researcher's database (CastorStudy) and a set of functions which can be used to import RWD or other study data (importer).

CastorClient

CastorClient supplies the key functionality of the package. It authenticates the user to the Castor API and has defined functions to interact with all the endpoints defined in the API. It allows for direct querying of database endpoints, but requires some knowledge of programming in Python. Examples of endpoints are creating a new survey package for a study participant, updating study data for a participant or retrieving information about the study. For an example, see Code Snippet 1.

CastorStudy

CastorStudy is a local representation of the researcher's study database. It downloads the raw study data through the API using the CastorClient module in the background and then augments and formats the study for easy searching. The local representation is split into two parts: one part defines the structure of the study and the other part defines the data collected in the study. For a visualisation of the study structure, see Figures 1 and 2.

Code Snippet 1 Example use of the CastorClient.

```
from castoredc_api import CastorClient
# Create a client with your credentials
c = CastorClient('MYCLIENTID', 'MYCLIENTSECRET', 'data.castoredc.com')
# Link the client to your study in the Castor EDC database
c.link_study('MYSTUDYID')
# Then you can interact with the API
# Get all records in the study
c.all_records()
# Create a new survey package
c.create_survey_package_instance(survey_package_id="SURVEY-PACKAGE-ID",
                                 record_id="TEST-001",
                                 email_address="example@fakemail.com",
                                 auto_send=True)
```

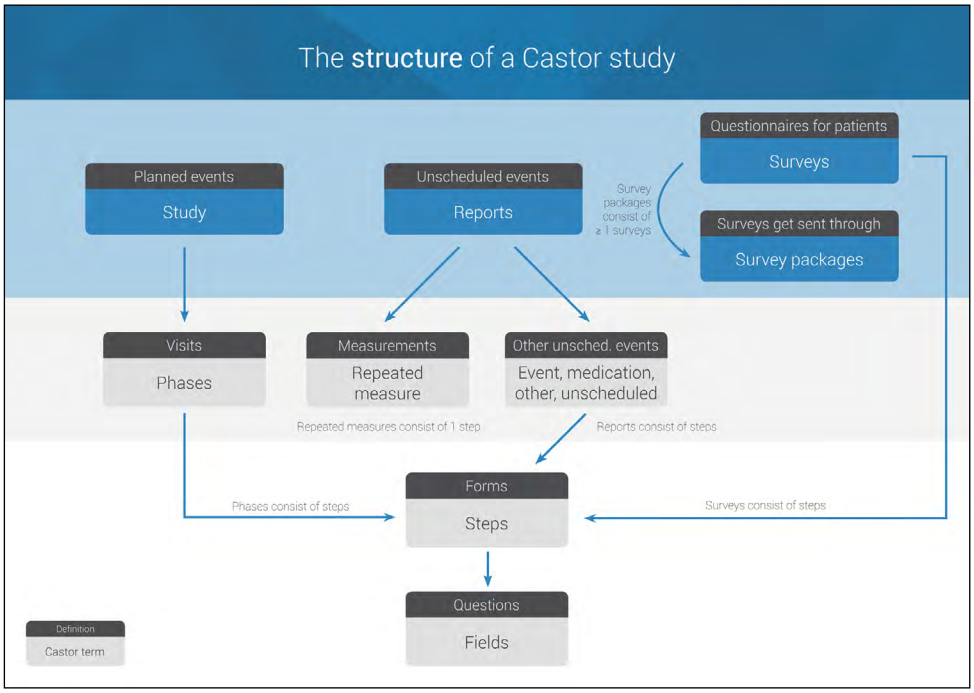

Figure 1 Overview of the Castor study structure.

Code Snippet 2 Example use of the CastorStudy.

```
from castoredc_api import CastorStudy
# Link to Castor database
study = CastorStudy('MYCLIENTID', 'MYCLIENTSECRET', 'MYSTUDYID', 'data.castoredc.com')

# Map only the study structure locally
study.map_structure()
# Find the field labelled “med_name”
field = study.get_single_field(“med_name”)

# Map the study data locally (also maps structure)
study.map_data()
# Get all data points in the study
data_points = study.get_all_data_points()

# Export your study to pandas dataframes or CSV files
study.export_to_dataframe()
study.export_to_csv()
```

The structure of a study is defined at the upper level by forms. These can consist of different types: study, report or survey. A study form is a planned phase in the study, a report form is an unscheduled or repeated event, and a survey form is a survey or electronic patient-reported outcome measure. Each form consist of a collection of fields which are the variables specified by the study protocol to be collected. Forms and fields can be considered the templates in which data per record is filled.

The data of the study is defined at the upper level by records. Each record is a study patient and is linked to a collection of form instances. These are instances of the study, report and survey forms and contain information on among other things when forms were created and filled in. Each form instance contains a collection of data points, which are instances of fields. These hold values and information for each variable in the study per record.

Mapping the study structure locally allows the researcher to quickly find specific fields or forms and is used when validating data to be imported. After the study structure is mapped, a representation of the study is created in Python. This only contains forms, steps and fields, and can be seen as the case report forms of the study and does not include data. When importing data, the structure is used to assess if fields exist in the study and validate values. Moreover, one can search for specific fields or forms in the structure to get information on these fields, see Code Snippet 2.

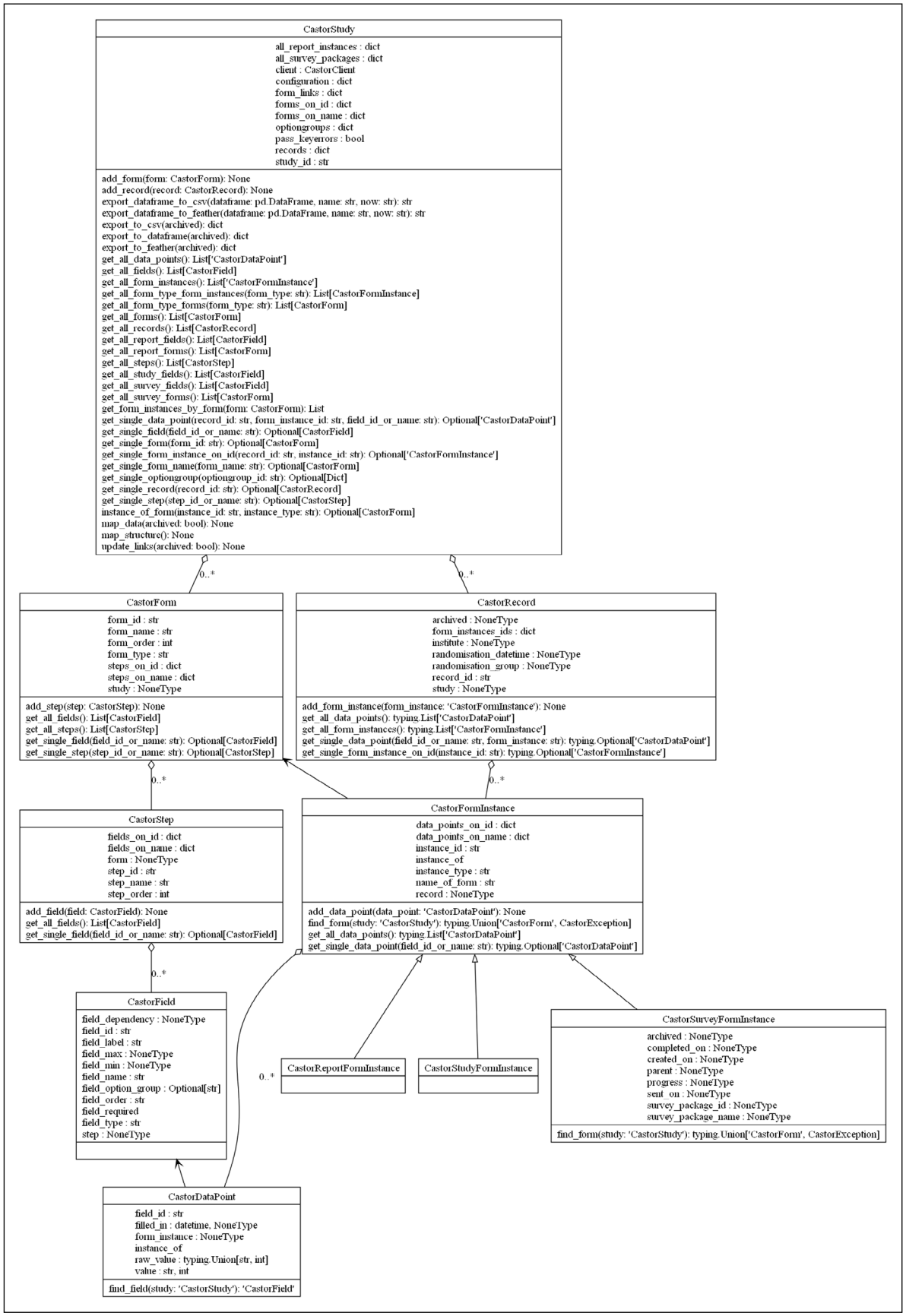

Figure 2 Unified modelling language (UML) diagram of the CastorStudy implementation.

One can also map the study data, which first maps the study structure and then exports all data from the Castor database, linking it to the structure and allowing the data to be accessed as Python objects. These functions are used to export data from the API to data analysis tools such as Python (pandas) and R [11]. For examples, see Code Snippet 2.

Importer

The importer can be used to read, clean, validate and import RWD into the Castor database. The importer is started by calling the function *import_data* with a set of configuration options. First the simple case of uploading data is shown, thereafter the complex options are described. In all cases the main input is the data to be imported, a xls(x) file with records as rows and variables as columns (Table 1).

This is combined with a second xls(x) file that maps the columns in the data file to the fields in the Castor database (Table 2). Headers "other" and "castor" are obligatory. For special cases, for example checkbox, radio or dropdown fields that define an 'other' option, see the software documentation [11].

There are three other options that need to be specified when uploading data. These are the study, whether the data is labelled and what the target is of the import. The study is defined by a CastorStudy (see above). Labelled data defines whether data in the data file contains labels (woman/man) or values (0/1) for data points that use an option group. The target defines whether the data to be imported are for a study, report or survey form. When importing report or survey data, it is also necessary to define the name of the report or survey. When uploading survey data a fall-back (fake or researcher owned) e-mail address needs to be supplied. While the surveys are not sent to this e-mail address, the database does not accept survey instances without a fall-back e-mail address. For examples of each data type, see Code Snippet 3.

PATIENT	MEDICATION	STARTDATE	STOPDATE	DOSE	UNITS
110001	Azathioprine	05-12-2019	05-12-2020	0.05	g/day
110002	Vedolizumab	17-08-2018	17-09-2020	300	mg/4 weeks
110003	Ustekinumab	19-12-2017	03-06-2019	90	mg/8 weeks
110004	Thioguanine	25-04-2020	27-05-2021	15	mg/day
110005	Tofacitinib	01-03-2020	31-12-2999	10	mg/day

Table 1 Example of a data file.

OTHER	CASTOR
patient	record_id
medication	med_name
startdate	med_start
stopdate	med_stop
dose	med_dose
units	med_units

Table 2 Example of a link file.

Code Snippet 3 Example use of simple import.

```
from castoredc_api import CastorStudy
from castoredc_api import import_data
# Link to Castor database
study = CastorStudy('MYCLIENTID', 'MYCLIENTSECRET', 'MYSTUDYID', 'data.castoredc.com')
# Import labelled study data
imported_data = import_data(data_source_path="studydatafile.xlsx",
                            column_link_path="studylinkfile.xlsx",
                            study=study,
                            label_data=True,
                            target="Study")
# Import non-labelled report data
imported_data = import_data(data_source_path="reportdatafile.xlsx",
                            column_link_path="reportlinkfile.xlsx",
                            study=study,
                            label_data=False,
                            target="Report",
                            target_name="Medication")
# Import labelled survey data
imported_data = import_data(data_source_path="surveydatafile.xlsx",
                            column_link_path="surveylinkfile.xlsx",
                            study=study,
                            label_data=True,
                            target="Survey",
                            target_name="Example Survey Package",
                            email="example@fakemail.com")
```

Before uploading the data file to the Castor database, the package prepares the data for import and validates all values. Every column in the data file is checked to see whether a target field exists and is part of the target. Every record is checked to determine whether it exists in the study. All labelled data is translated to the corresponding values if applicable. For all columns the values to be imported are checked against minimum or maximum allowed values (numeric and year fields), the allowed option groups (dropdown, radio and checkbox fields), and the correct formatting (date, datetime and time fields). When one or more errors are encountered in the data file, the process is aborted and the program outputs a file indicating for which values an error was found.

The import_data function can be supplied with some extra configuration options for complex situations. These are translating variable names between databases, merging multiple variables into one, formatting options and an option for asynchronous interaction with the API. See the software documentation [11].

QUALITY CONTROL

The software has been thoroughly tested and there is a testing suite with more than 500 tests. Tests coverage is 98% and covers almost all aspects of the CastorClient, CastorStudy and importing suite. As a large proportion of the tests interact with the Castor database, tests are currently run through a Github hosted runner. Other users can run tests on their local machine for development after contact with the code owner.

(2) AVAILABILITY

OPERATING SYSTEM

Platform Independent

PROGRAMMING LANGUAGE

Python >= 3.8;

ADDITIONAL SYSTEM REQUIREMENTS

No specific requirements

DEPENDENCIES

Pandas >= 1.3.1; numpy >= 1.21.1; openpyxl >= 3.0.7; tqdm >= 4.62.0; httpx >= 0.19.0;

LIST OF CONTRIBUTORS

van Linschoten, Reinier Cornelis Anthonius[a, b]
Knijnenburg, Sebastiaan Laurens[c]
Lutro, Andreas[c]
[a]Department of Gastroenterology & Hepatology, Franciscus Gasthuis & Vlietland, Rotterdam, Netherlands
[b]Department of Gastroenterology & Hepatology, Erasmus MC, Rotterdam, Netherlands
[c]Castor, Amsterdam, The Netherlands

SOFTWARE LOCATION

Archive

Name: CastorEDC API
Persistent identifier: https://pypi.org/project/castoredc-api/
Licence: MIT
Publisher: Reinier Cornelis Anthonius van Linschoten
Version published: v0.1.9.1
Date published: 13/07/2023

CODE REPOSITORY

Name: CastorEDC API
Identifier: https://github.com/reiniervlinschoten/castoredc_api
Licence: MIT
Date published: 13/07/2023

LANGUAGE

English

(3) REUSE POTENTIAL

The described software originated from the IBD Value study [12]. This longitudinal multicentre non-randomised cluster trial combines patient-reported outcomes and RWD to study variation in quality of care in inflammatory bowel disease and the effect of a uniform care pathway on quality of care. CastorEDC API has been successfully used to merge data from eight hospitals, two different electronic health records and a registry with patient-reported outcomes into a single database in Castor EDC. Moreover, the Erasmus Medical Centre is testing this software for importing data into Castor EDC. The scope of the software is not limited to data from electronic health records or patient-reported outcomes. It allows creating automatic scripts to transform, validate and import differing types of data from large and diverging databases, such as data from wearable devices or lifestyle and environmental data, into a single Castor study database.

The use of test-driven development and continuous integration to ensure high code quality and proper code formatting led to a modular codebase which can be easily extended. As validity and reliability of data are of utmost importance in scientific research, a large testing suite was pivotal for ensuring correct functionality, both during development and when refactoring and cleaning up the code base.

While the software has been extensively tested and is ready for production, there remain some possible improvements. Most important among them are full asynchronous export of study data and more extensive configuration options for differences in data structures between the data file and the Castor database. When

exporting study data for a large study, the API requests for the raw data take considerable time and block the program until the Castor database responds. Currently export happens partially asynchronously, but full asynchronous requests can speed this up considerably. Moreover, there is no support for mapping between the columns in the data file and the fields Castor database other than one-to-one or many-to-one. These improvements are being tracked on the project at Github and users of the software are encouraged to contribute or submit other feature requests [11].

DATA ACCESSIBILITY STATEMENT

The described software is free to use and open source accessible on https://github.com/reiniervlinschoten/castoredc_api.

ACKNOWLEDGEMENTS

We would like to thank Andreas Lutro and Nick Groenen for code review and Christiaan Oudmaijer for his help in testing the software.

FUNDING INFORMATION

Development of this software was funded partly by the Franciscus Research Foundation (Grant Number N/A) and Castor EDC (Grant Number N/A). The funders had no role in design of the software package.

COMPETING INTERESTS

Drs. van Linschoten has nothing to disclose.
Dr. Knijnenburg is the Principal Innovation Engineer of Castor EDC.
Dr. van Noord reports personal fees from Janssen and Takeda outside the submitted work.
Dr. West reports personal fees from AbbVie, Janssen and Pfizer outside the submitted work.

AUTHOR CONTRIBUTIONS

RCAvL designed and created the software with help from SLK and supervision of DL and RLW. RCAvL wrote the first draft of the manuscript and all co-authors critically revised the manuscript.

Rachel Louise West and Desirée van Noord share last authorship.

AUTHOR AFFILIATIONS

Reinier Cornelis Anthonius van Linschoten orcid.org/0000-0003-3052-596X
Department of Gastroenterology & Hepatology, Franciscus Gasthuis & Vlietland, Rotterdam, Netherlands; Department of Gastroenterology & Hepatology, Erasmus MC, Rotterdam, Netherlands

Sebastiaan Laurens Knijnenburg orcid.org/0000-0002-2475-6254
Castor, Amsterdam, The Netherlands

Rachel Louise West
Department of Gastroenterology & Hepatology, Franciscus Gasthuis & Vlietland, Rotterdam, Netherlands

Desirée van Noord orcid.org/0000-0003-1578-9123
Department of Gastroenterology & Hepatology, Franciscus Gasthuis & Vlietland, Rotterdam, Netherlands

REFERENCES

1. **Eapen ZJ, Lauer MS, Temple RJ.** The Imperative of Overcoming Barriers to the Conduct of Large, Simple Trials. *JAMA*. 2014; 311(14): 1397–8. DOI: https://doi.org/10.1001/jama.2014.1030
2. **Lauer MS, D'Agostino Sr. RB.** The Randomized Registry Trial — The Next Disruptive Technology in Clinical Research? *N Engl J Med*. 2013; 369(17): 1579–81. DOI: https://doi.org/10.1056/NEJMp1310102
3. **Jeannic AL, Quelen C, Alberti C, Durand-Zaleski I, Comparec Investigators.** Comparison of two data collection processes in clinical studies: electronic and paper case report forms. *BMC Med Res Methodol*. 2014; 14. DOI: https://doi.org/10.1186/1471-2288-14-7
4. **U.S. Food & Drug Administration.** Real-World Evidence. [cited 2023 11-07-2023] Available from: https://www.fda.gov/science-research/science-and-research-special-topics/real-world-evidence
5. **Mc Cord KA, Al-Shahi Salman R, Treweek S, Gardner H, Strech D, Whiteley W,** et al. Routinely collected data for randomized trials: promises, barriers, and implications. *Trials*. 2018; 19(1): 29. DOI: https://doi.org/10.1186/s13063-017-2394-5
6. **Ciwit BV.** Castor Electronic Data Capture. Amsterdam; 2016.
7. **Ciwit BV.** Enable importing EHR data for your study 2018 [cited 2021 26-11-2021]. Available from: https://helpdesk.castoredc.com/article/260-enable-importing-ehr-data-for-your-study.
8. **Ciwit BV.** Castor EDC API 2016; [cited 2021 26-11-2021]. Available from: https://data.castoredc.com/api.
9. **Potters W.** castorapi 2019; [cited 2021 26-11-2021]. Available from: https://github.com/wouterpotters/castorapi.
10. **Ciwit BV.** Castor API R Package 2016; [cited 2021 26-11-2021]. Available from: https://github.com/castoredc/castoredc.

11. **van Linschoten RCA.** CastorEDC API; 2021 [cited 2021 26-11-2021]. Available from: https://github.com/reiniervlinschoten/castoredc_api.
12. **van Linschoten RCA, van Leeuwen N, Nieboer D, Birnie E, Scherpenzeel M, Verweij KE,** et al. Value-based care pathway for inflammatory bowel disease: a protocol for the multicentre longitudinal non-randomised parallel cluster IBD Value study with baseline period. *BMJ Open*. 2022; 12(1).DOI:https://doi.org/10.1136/bmjopen-2021-050539

TO CITE THIS ARTICLE:
van Linschoten RCA, Knijnenburg SL, West RL, van Noord D 2023 CastorEDC API: A Python Package for Managing Real World Data in Castor Electronic Data Capture. *Journal of Open Research Software,* 11: 12. DOI: https://doi.org/10.5334/jors.436

Submitted: 30 June 2022 **Accepted:** 13 October 2023 **Published:** 20 October 2023