Dynamic Report Push

V3.0.2

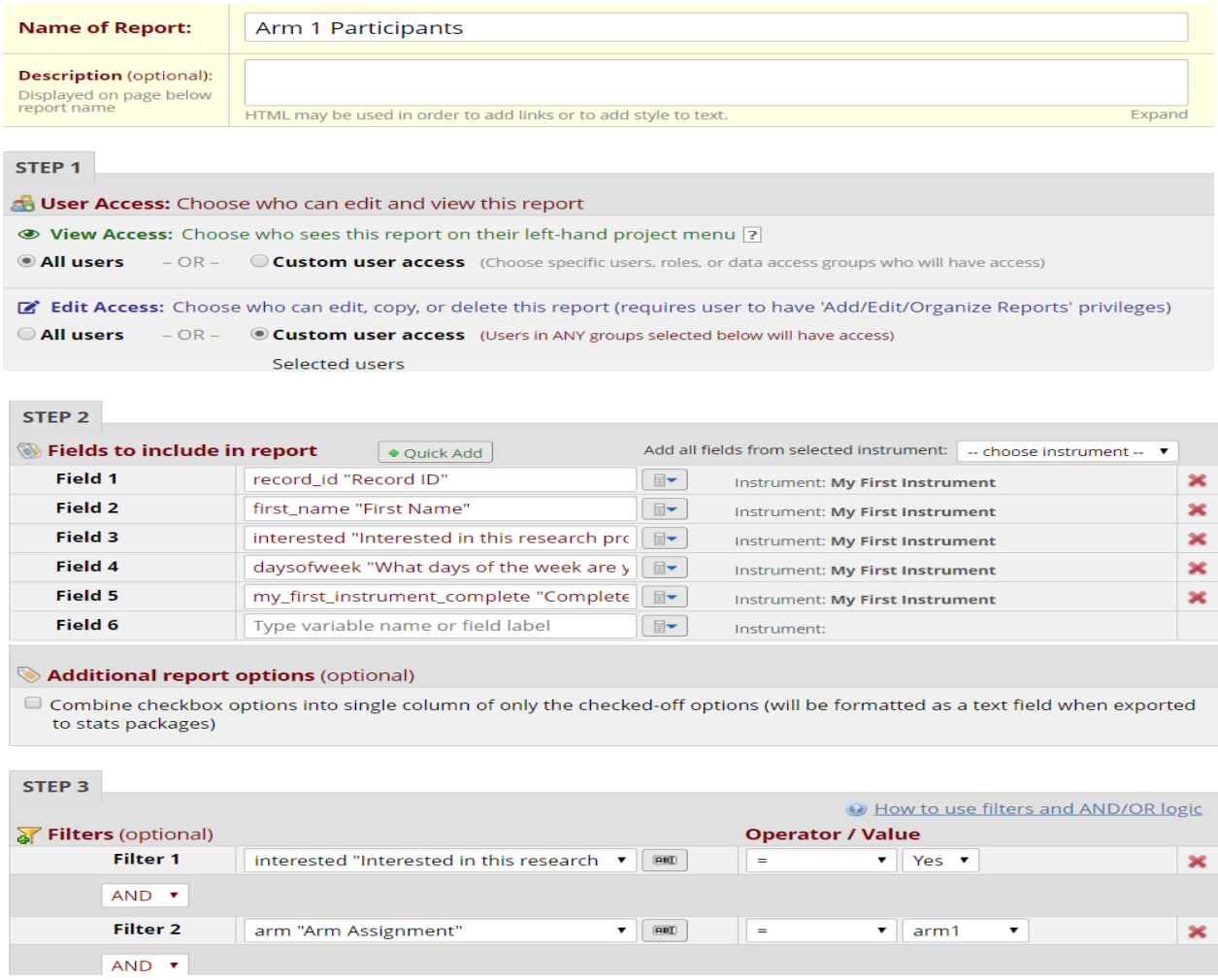
Description-

The module allows you to use the reporting tool and its built-in logic to select variables to push from the current REDCap project into one or more receiving project(s). Variable names must be the same. Once successfully configured, users can select a button that triggers the push. A data entry trigger can also be used. The configuration module will allow you to add an event name when going from a classic project to longitudinal. You can also delete the event name and/or repeating instrument fields when going from longitudinal to classic projects.

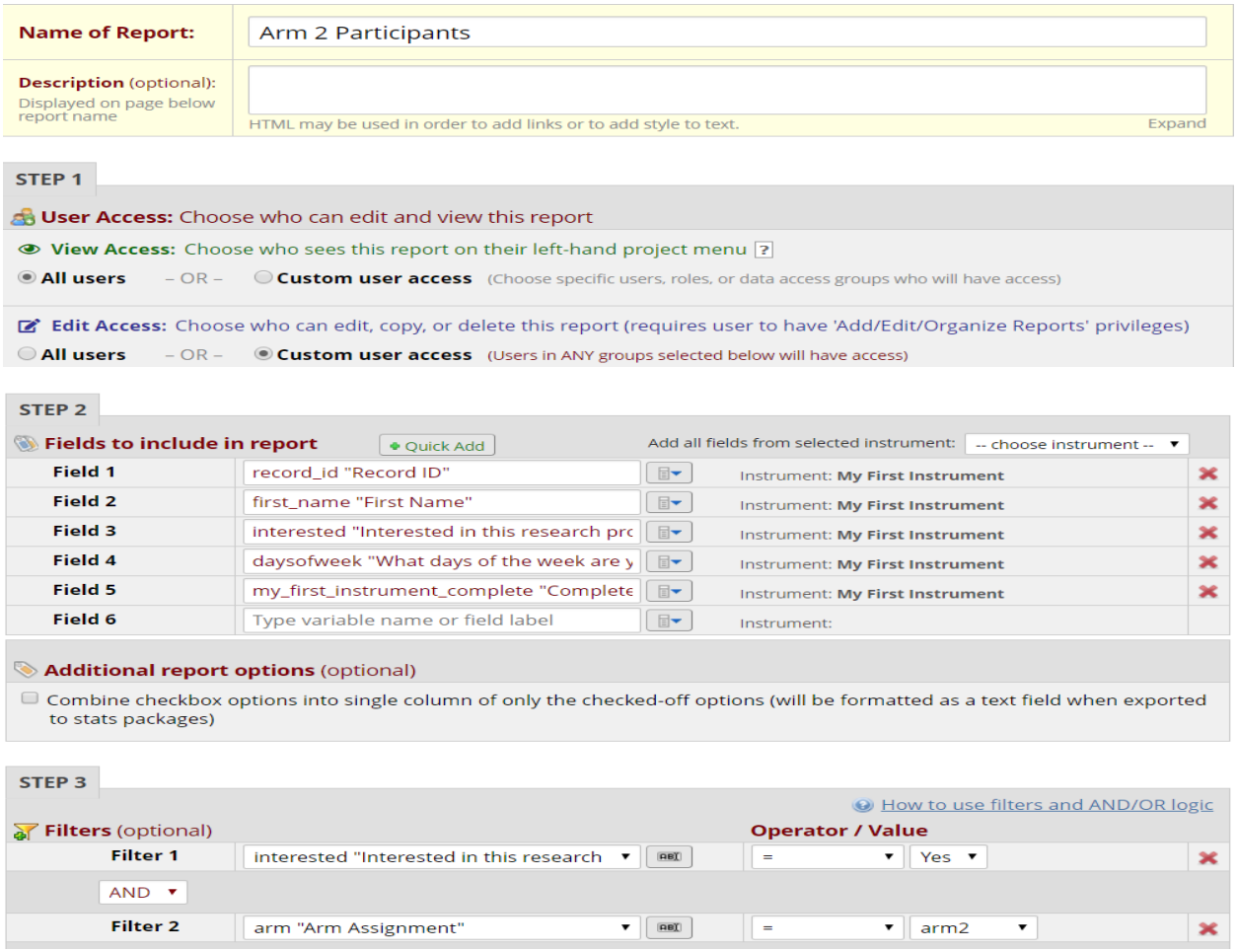
Configuration-

The REDCap builder must first create the report(s) and have tokens for all projects.

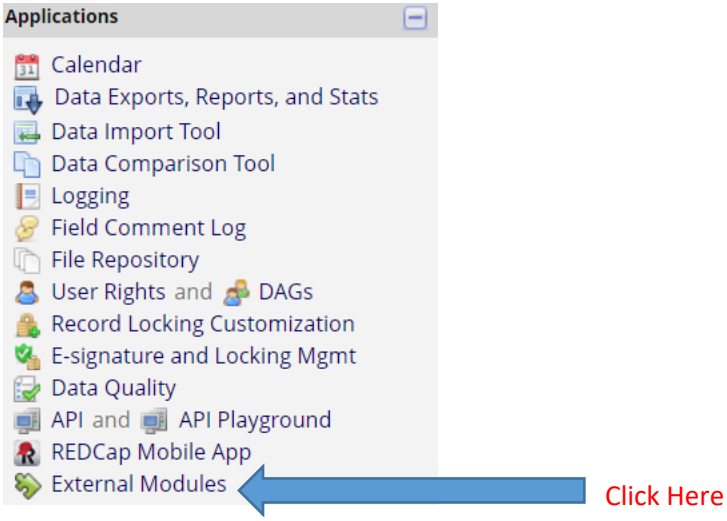
Report 1



Report 2

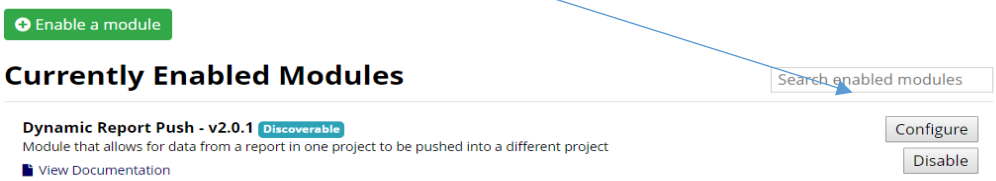


Click on External Modules and then Configure.

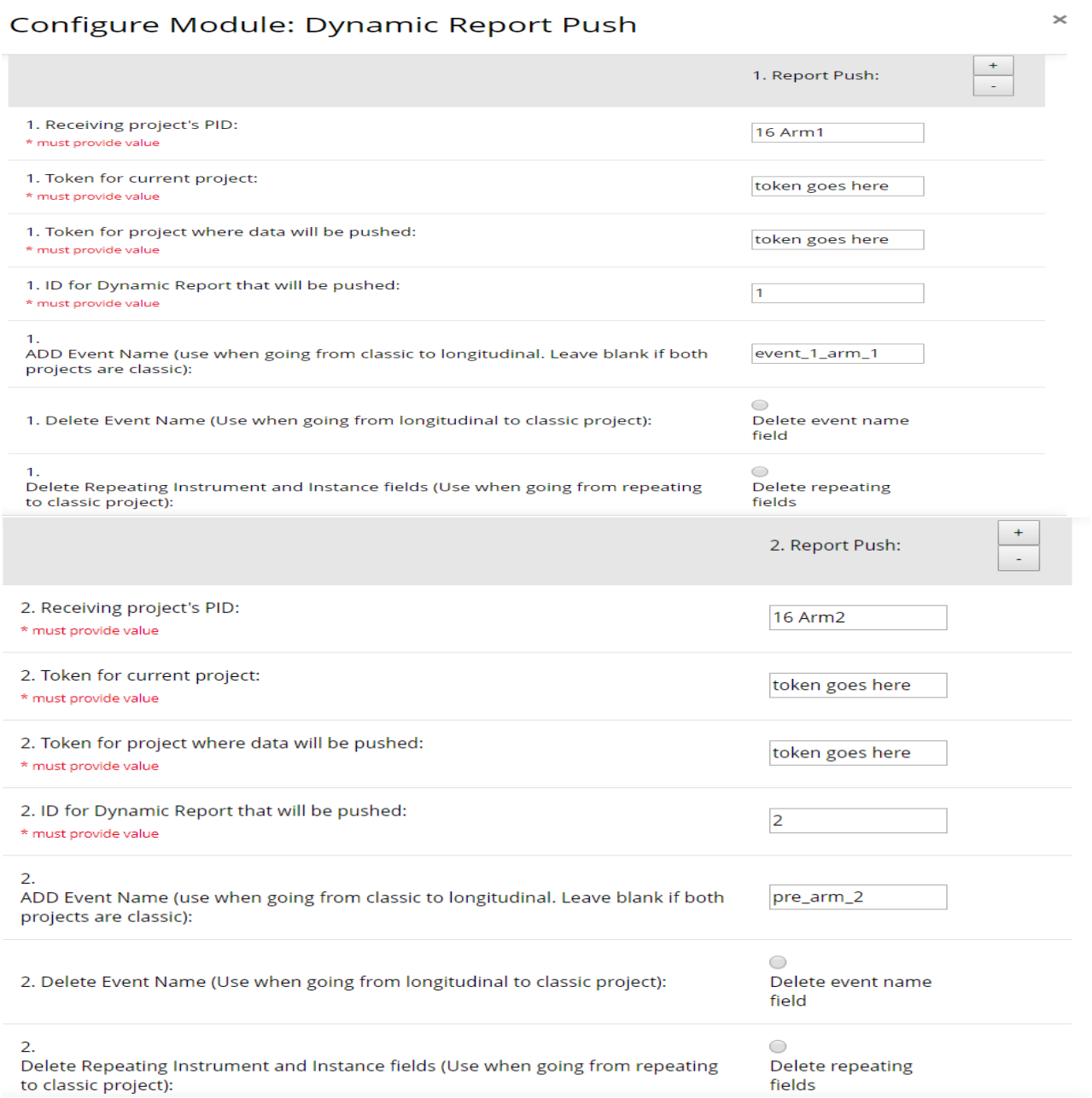


Click Here

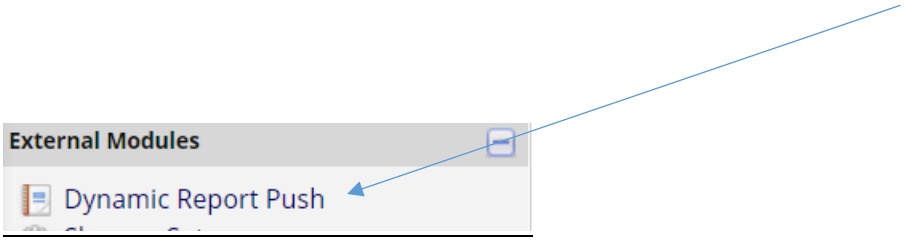
Now Click Configure



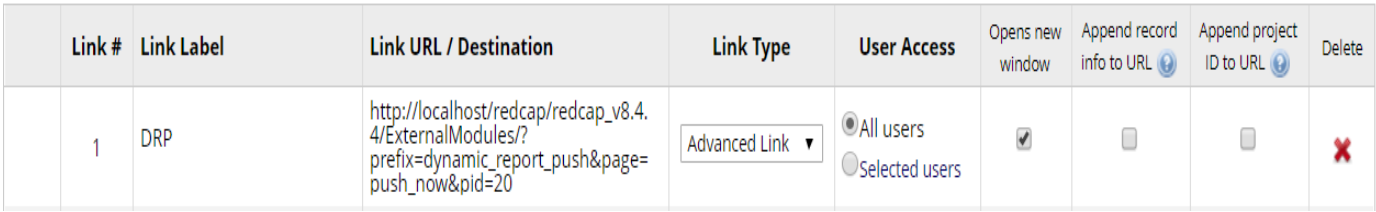
Enter PID or Project Name to keep documentation of the linked project. Recording this information is important in case the builder is not available and a different person needs to follow the data flow.



Super users and project builders can click the following button to transfer report information into the linked project.



To give all users the ability to transfer data you will need to create a bookmark.

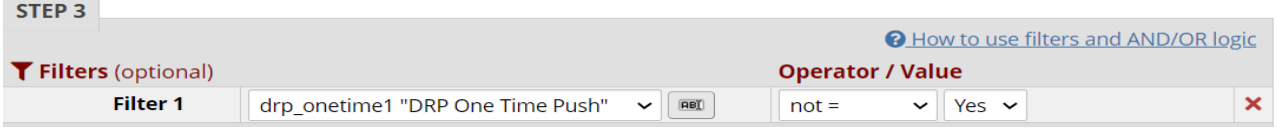


Changes in version 3.0.2

The following revisions should help with larger and more complicated projects.

Builders can configure reports that move data into different loops of the module (DRP1 and DRP2). A use case for this is when it is necessary to trigger some fields each time users save a record but others on demand.

Builders can configure a one-time push to limit the process from running each time. You will need to create a field in DRP1 and or DRP2 and select it in the dropdown. Include logic in your report to drop off records once the push value is = '1'. Example uses drp_onetime1 but version 3.0.2 allows you to use different fields.



Words of caution.

ID Creation-It will be less risky if you do not create records in the linked project that is receiving data. However, if that is necessary, be sure to think through how to avoid overwriting data in the second project.

Record Locking-Best practice will be to lock records in both projects when locking data. If a change needs to be made then unlock the linked record in both projects and lock it back after the change. The module can stop working if a record is locked in the receiving project and you are trying to update data.

Contact for issues-

Gary Rains

IT Analyst for Duke University School of Medicine

Gary.Rains@duke.edu