Shots to show

Gianluca Moro[*]

23 April 2018

1 Introduction

The project implements a web page which can give a compact overview of the shots' statuses of a day. It is possible to:

- select a date;
- show all the shots present in that date and their status;
- check the shots to flag them as ready to be copied.

The web app is implemented in Php and uses a SQLite database as storage for flags status (by Timo Schröder).

2 Code overview

2.1 Libraries

The page uses the following libaries, stored in `web/js`:

- Twig (`https://twig.symfony.com/`, used in the Symfony framework, but can be used standalone too): is a templating library for web pages
- Kalendae (`https://github.com/Twipped/Kalendae`): is an utility do show a calendar - the usage of this library can be avoided using the standard HTML input date form (the solution used at the moment). Directory `archive` stores the used source code.

[*] Gianluca.Moro@unipd.it - at Max-Planck-Institut für Plasmaphysik, Greifswald, DE

- liblog.php: some logging functions (not used at the moment)

2.2 Main code

The Php source files are in the main directory:

- config.php: keeps some configuration variables

  $ipAllowedToCopy keeps a list of IPs allowed to check shots: if the connection comes from an allowed IP that user can check shots. If the list is empty, everyone can check shots.

  $shotsDirectory the directory storing MDSplus shots

  $mainW7XTreeName the tree which is used to get the list of shots: if a shot is present, it must be present in this directory.

  $statusDefinitions the map from status as recovered from database to its color, the presence (`true`) or absence (`false`) of checkbox, and a message shown on the web page legend.

  $dbFilename the SQLite file with the shots' database and the variable **$dbShots** containing the database (as a matter of fact, logically not a configuration variable).

  $listOfTrees the list of current trees, present in configuration as it was thought for a "default" list: in the current code it is always initialized to the corrent list, so its value as defined here is never used.

  $logFileName and **$logLevel** at the moment not used

- externalCmds.php: code to interface to database and filesystem

  getListOfTrees() get the list of all the trees, from the external command `w7x_gettrees`

  getTableOfStatus($listOfTrees, $listOfShots) for each tree in the 1^{st} parameter list and for each shot in the 2^{nd} parameter's list, a query to the database is done to get the status. The query is: `SELECT stat FROM shotdb WHERE shot=sName AND expt=tName;`

  saveShot($tree, $shot) this function checks the given shot: the real action is to set the status to "2" if is was "1". The query to set the status is: `UPDATE shotdb SET stat=2 WHERE shot=sName AND expt=tName;`.

getListOfShots($dirName, $shotDate) get the list of the shots in a given date.

- index.php: the main function is `buildHome`, which show a form to select the date, recover the list of shots for that day, recover all the statuses for the found shots and call the Twig template rendering with all the information.

  If there is a request to check (the user selected some shots and submitted them), then those shots are checked in the database (only if the IP is allowed)

2.3 The html interface

The files are in the `web` directory. The `web/js` contains the static files for web (images and javascript libraries), while the base `web` contains 2 Twig templates (extension phtml):

template.phtml a template loading needed Javascript libraries and with the header layout of the page.

home.phtml the main body, with different sections:

- a block which shows messages (green border) or errors (red border) if present.
- the date selection form: here the code using "Kalendae" is commented.
- a legend of the meaning of the colours
- the main table, showing the info (color) of the statuses and checkboxes
- the submit button if something has been checked

2.4 Other files

- phpinfo.php: shows Php configuration
- archive: source of kalendae library
- testauth: An authentication test page with kerberos
- testShots: a local simulation of a tree structure

3 Information on the infrastructure

- The source is in `mds-data-1:/home/gimo/shots`, and can be tested by running in that directory the command `php -S 10.44.4.11:4000` and accessing the web page by `http://10.44.4.11:4000/index.php`. The present documentation is in `mds-data-1:/home/gimo/shots-doc`.
- Server with MDSplus data: `mds-control`, `mds-data-1`, `mds-data-2`.
- The trees are stored in: `/w7x/vault/`*treename*, with the following filename format: `w7x_yymmddnnn`.
- The status is coded as integer:
  - <=0 - RED - NOT READY
  - 1 - RED - CHECK - ready to be checked
  - 2 - ORANGE - checked
  - >2 - GREEN - PROCESSED