General requirements on web-admin interface for DC service

The web-admin user interface is common administration UI for management of DC service, make operations, and visualize statistics and so on. The architecture is typically web UI with client-side and server side parts each are implemented on one framework library. The operations and stats data are correspond to the DC service functionality and requests description specified in requests and architectural documentation (see DC_application_requirements.docx and DC_application_architecture.docx documents). Common architecture is typically regular web-application with session-based authorization, users list, types of users, multi-language and so on.

General principles and rules

- The common graphical design principles are a minimalism and regular controls that is defined in terms of objects of the one javascript library that will be chosen (jquery and so on), no mixes.
- The server-side implementation must to use one MVC framework library (yii and so on), no mixes of clear php code.
- The graphical and HTML design need to be implemented according general templates, skins or related principles defined for framework to complete support fast update or change the template and so on…
- The web-application need to satisfy with the modern multi-language principles including easy way to switch interface language, edit textual messages, add more supported languages and so on. Only UTF-8 encoding support required for pages and database.
- All forms fields need to be validated by type and mandatory (if marked) on client and server side.
- During all operations execution time the “loader” visualization need to be displayed in upper layer to show that process of remote request still “in progress” state.
- The size of fonts, control elements, groups of items, padding, and spacing need to be as small as possible to get a more free space to locate more information on one screen.
- All controls like input fields, buttons, grid columns, lists, and so on need to have hover tooltips to display small help about how this control acts or what operation means and so on.
- All pages and dialogs need to have “help” button that opens modal dialog with “big help” with detailed description.
- Source code need to be well formatted and satisfy with requirements of PHP and JavaScript source code and naming conventions defined in http://pear.php.net/manual/en/standards.php and http://contribute.jquery.org/style-guide/js/ correspondently.

Root page

"GR-RP"

The root page is different for authorized and not authorized users. Not authorized user's root page is login page. Authorized user's root page is first top main menu item.

Main menu

"GR-MM"

Top main menu items for authorized users: **Sites, Resources, Batches, Stats, Configuration, Users**. When menu item is active it displayed as selected.

Path line

"GR-PL"

The path line – it is textual references in form of path to the current menu, submenu, action dialog position place and so on in the site's user interface tree. Looks like "/Sites/Add" or "/Resources/Edit URL/" and so on…

It located under the top main menu and formed as text html reference to allow making a jump to any of pages that lead to current.

Traceback log area

"GR-TLA"

All authorized user's pages need to have full width bottom area to display the log of operations that optionally can to include: command lines that executed on server-side, request and response jsons, error messages from stderror console and so on information. This trace-back area needs to have vertical and horizontal scroll, proportional font and control to hide/show it quickly. Possible it can to have tabbed way to view the different type of information the way as browsers developer's tools (not mandatory request).

Sites

"GR-S"

The sites manage page used to execute main operations with the Site objects and consists of several pages or/and dialogs depend on implementation and visual controls UI library used.

Sites list page.

"GR-S-SL"

Main work controls on the page: the filter and the submit button, the grid, the "New site" button.

Filter fields:

- **URL pattern** – string, 255 characters max, used as LIKE SQL query pattern to select root site's URLs. Mandatory not empty.
- **User Id** – select, filled according with user type. For "Administrator" type listed all Users names and additionally empty (first) item that supposes "ANY" user. For "User" type – only this user name listed. For "Viewer" type – only allowed users names listed.
- **Max items per page.** Select filled with values 10, 20, 30…100.

The grid columns:

- **N** – Number in list order. Numerical integer, value depends on pagination, start from 1.
- **Id** – Site Id. String, 32 characters length.
- **State** – the state. String symbolic names: Active – 1, Disabled – 2, Suspended – 3.
- **Root URLs** – string with list of root URLs. This is most wide column and need to have flexible width depend on window width. In case of CSV list is not fit in one line it need to be wrapped in several lines inside grid cell.
- **CollectedURLs** – numeric integer, total number of URLs collected.
- **NewURLs** – numeric integer, number of URLs in new state (not processed).
- **Resources** – numeric integer, number of URLs that resource is crawled.
- **Contents** – numeric integer, number of URLs that resource is processed and some content tags scraped.
- **DeletedURLs** – numeric integer, number of deleted URLs that are possible still not removed from the system physically.
- **Errors** – numeric integer, number of errors that happened during last crawling iteration.
- **Size** – numeric integer, the total size of all resources downloaded during last crawling iteration in bytes.
- Operations buttons/icons:
    - Resources – shift to the "Resources" main menu item, fills the "Site Id" filter field with this site Id value and activates the "Find" button. All another filter's fields are filled by default.
    - View – opens the "Site fields view" dialog/page.
    - Edit – opens the "Site properties edit" dialog/page.
    - Cleanup – opens the "Site cleanup" dialog/page.
    - Re-crawl – opens the "Site re-crawl" dialog/page.
    - Delete – opens the "Site delete" dialog/page.

On filter submit button the pattern string and selected user Id (or special value if "ANY") send to server-side. Server side script creates request based on cli utility specification, provides the json file created in temporary dir, parse response json and return some data that will be visualized with web-UI.

** If it is possible, as additional feature not for first implementation – the possibility to manage columns/fields of the DC-Site object that will be included in to the results grid as columns.*

Site fields view dialog/page

"GR-S-FV"

It visualizes all sites' fields returned for site as a result of "SITE_FIND" or "SITE_STATUS" requests. There are several types of fields need to be visualized, for example – simple text fields, lists, enumerations, Boolean and so on. The controls need to be chosen to better satisfy the nature and specific feature of information to display. For example, URLs supposes long strings (up to 4K character length), the lists need to show as many items as possible to not break the graphical design, the time and date marks need to be formatted according with common format than can be set up as configuration parameter and so on.

List of fields:

Main status area:

- **Description** – the short textual description of the meaning of the site collection. Possible, the collection name or keywords for memories.
- **Id** – the unique site Id, 32 characters md5 checksum representation. Json source field: "**id".**
- **Root URLs** – list of URLs strings of root entries for crawling process of site object. Need to be represented as list of items with vertical and horizontal scrolls, first three needs to be visible always. Json source field: "**urls**".
- **State** – state of site as an object of DC service, numeric integer {1 – Active, 2 – disabled, 3 - suspended}. Better to display as keyword or icon. Json source field: "**state**".
- **Iterations** – number of crawling iterations performed for this site, numeric integer. Json source field: "**iterations**".
- **Collected** – collected URLs counter. Mapped to **collectedURLs** field the response json URL object.
- **New** – new URLs counter. Mapped to **newURLs** field the response json URL object.
- **Crawled** – counter of URL crawled times. Mapped to **resoucres** field in the response json Site object.
- **Processed** – counter of URL processed times. Mapped to **contents** field in the response json Site object.
- **Deleted** – numeric integer, number of deleted URLs that are possible still not removed from the system physically. Mapped to **deletedURLs** field in the response json Site object.
- **Errors** – number of errors happened since site was created or last crawling iteration from scratch, numeric integer. Json source field: "**errors**".
- **Errors types** – lists type names of errors that happened since site was created or last crawling iteration from scratch, numeric integer mapped to string names. Json source field: errorMask. The errorMask in json is numeric bit set; the description of each bit is defined in the main architectural document file for DC service DC_application_architecture.docx in the **"Site.ErrorMask and URLs.ErrorMask specification"** section.

Limits area:

- **Priority** – the numeric value to sort sites for fetch URLs operation during crawling process, numeric integer, and higher value means higher priority. Higher priority means that in case of all another criterions are the same URLs of sites with higher priority will be taken more often. Json source field: "**priority**".
- **Filters** – list of filters patterns used during the crawling process of site object. Need to be represented as list of items with set of fields. Fields list: ***pattern, type, mode, cDate and uDate***.

Detailed description of filters field see at **"Site filter item view dialog**" description. The filters list item click need to open modal dialog to display all filter's fields with descriptions. Item in this list need to have fields: ***pattern, type, mode.*** Values of **type** and **mode** fields can be represented as icons. Json source field: "**filters**".

- **Max URLs** – approximately maximum number of URLs that can be collected for this site during the crawling process. This value is multiplied on number of hosts in DRCE **n-type** cluster and can be overflowed due to parallel crawling on one host. So, it is not strict limit but desired. Json source field: "**maxURLs**".
- **Max resources** – approximately maximum number of resources that can be processed (by the processing algorithms, for example – scraping) during the crawling process. This value is multiplied on number of hosts in DRCE **n-type** cluster and can be overflowed due to parallel crawling on one host. So, it is not strict limit but desired. Json source field: "**maxResources**".
- **Max errors** – approximately maximum number of errors (all kind) that can be reached during the crawling process (including processing). This value is multiplied on number of hosts in DRCE **n-type** cluster and can be overflowed due to parallel crawling on one host. So, it is not strict limit but desired. Json source field: "**maxErrors**".
- **Max resource size** – maximum size of resource downloaded from web-site to store it and process. If resource greater than this limit it is not stored in the file system storage and will not be included in farther processing. This value is absolute, bytes. Json source field: "**maxResourceSize**".

Processing area:

- **Processing delay** – number milliseconds to delay before next processing operation with resource, numeric integer. Json source field: "**processingDelay**".

HTTP area:

- **URL type** – defines default type of URL for URLs collected during crawling process, numeric integer. Values:
  - 0 - Regular, collect URLs and insert only for this site according filters;
  - 1 - Single, do not collect URLs,
  - 3 - Collect URLs, create sites and insert for all.

  Better visualization as icons with description. Json source field: "**urlType**".
- **Timeout** – number milliseconds to wait on http-request response, numeric integer. Json source field: "**httpTimeout**".
- **Delay** – number milliseconds to delay before http-request, numeric integer. Json source field: "**requestDelay**".

Properties area:

- **Properties list** – list of properties represented as key-value; string max length of name is 32 characters, max length of value is 4096 characters. Need to be represented as list of items that has two columns (or three column grid, first column is order number from 1) with vertical and horizontal scrolls and three items visible always. Json source field: "**properties**".

Dates area:

- **Creation date** – site creation date. Json source field: "**cDate**".
- **Touch date** – site touch date. Json source field: "**tcDate**".
- **Update date** – site update date. Json source field: "**uDate**".
- **Re-Crawl date** – site re-crawl date. Json source field: "**RecrawlDate**".

Traceback area:

- **Error code** – response error code, displayed as numeric value, possible some icons for set of values. Json source field: "**errorCode".**

- **Error message** – text message describes the error code, possible list of messages separated by semicolon. Need to be formatted pretty way to see if not empty. Both “Error code” and “Error message” need to be aligned as separated area to differ from all another fields section below. Json source field: “**errorMessage”.**

Bottom located controls – buttons “Close” and “Help”. The “Close” button hides the modal dialog and returns focus to the item in sites list on a grid. The “Help” button opens modal dialog displays contextual help content about site fields.

Site filter item view dialog

“GR-S-FI”

Displays the modal dialog with site filters fields with descriptions. Fields list: ***pattern, type, mode, cDate and uDate***. **Type** and **mode** field’s values can be represented as icons. Additional control elements – “Ok” and “Help” buttons. Help button opens general “Help” modal dialog to display help message for site filter item.

Site fields edit dialog/page

“GR-S-FE”

This modal dialog similar with the “Site fields view” – visualizes all sites’ properties returned for site as a result of “SITE_FIND” or “SITE_STATUS” requests, but gives possibility to edit and update them. It is possible to use for either this dialog the same code with some mode switches, for example – editable/read only or something like that. Key difference – it is possibility to enter/change value and to make update operation. As opposite to “Site fields view” – this dialog has three buttons – “Cancel”, “Update” and “Help”. The “Cancel” acts the same way as “Close” button, the “Help” button acts exactly the same way and the “Update” button – executes SITE_UPDATE operation and then SITE_STATUS operation for this site, closes the modal dialog and updates fields values in the sites list grid for this site. Also, it updates all fields inside the site’s data container on client-side to give the possibility to display actual updated fields for next “View” or “Edit” actions without reload data and additional requests.

Site cleanup dialog/page

“GR-S-CD”

This modal dialog contains fields used to tune site cleanup procedure. Depends on tune options chosen it may require SITE_UPDATE and SITE_STATUS operations to ensure that site was switched to proper mode. Tune options:

- Suspend site before cleanup – checkbox, not checked by default.
- Timeout value – the timeout value for wait on response on cleanup operation.

Additional controls on a dialog: buttons “Cancel”, “Cleanup” and “Help”. The “Cleanup” button starts process in sequence. Depends on tune options can do SITE_UPDATE, SITE_STATUS, SITE_CLEANUP, and SITE_STATUS operations or just a SITE_CLEANUP and SITE_STATUS. The same as the “Site fields edit” operation – it closes the modal dialog and updates fields values in the sites list grid for this site. Also, it updates all fields inside the site’s data container on client-side to give the possibility to display actual updated fields for next “View” or “Edit” actions without reload data and additional requests.

Site re-crawl dialog/page

"GR-S-RD"

This modal dialog contains fields to tune site re-crawl procedure. Depends on tune options chosen it may require complete "Site cleanup" operation before re-crawl action. Tune options:

- Cleanup site before re-crawl – checkbox, checked by default.
- Timeout value – the timeout value for wait on response on re-crawl operation.

Additional controls on a dialog: buttons "Cancel", "Re-crawl" and "Help". The "Re-crawl" button starts process in sequence. Depends on tune options can do SITE_UPDATE, SITE_STATUS, SITE_CLEANUP, and SITE_STATUS operations or just a SITE_CLEANUP and SITE_STATUS. The same as the "Site fields edit" operation – it closes the modal dialog and updates fields values in the sites list grid for this site. Also, it updates all fields inside the site's data container on client-side to give the possibility to display actual updated fields for next "View" or "Edit" actions without reload data and additional requests.

Site delete dialog/page

"GR-S-DD"

This modal dialog contains fields used to tune site delete procedure. Depends on tune options chosen it may require SITE_UPDATE and SITE_STATUS operations to ensure that site was switched to proper mode. Tune options:

- Delete task type – select, two items, titles "Asynchronous" and "Synchronous", values 2 and 1 correspondently. By default the "Synchronous".
- Timeout value – the timeout value for wait on response on delete operation.

Additional controls on a dialog: buttons "Cancel", "Delete" and "Help". The "Delete" button starts process in sequence: SITE_UPDATE (to disable site processing, set state=2 disabled, SITE_STATUS to get information about all nodes processed update and state is disabled, SITE_DELETE and SITE_STATUS to check is the site was really deleted on all hosts. The same as the "Site fields edit" operation – it closes the modal dialog and removes deleted site item from the sites list grid.

Resources

"GR-R"

This page displays search form with filter fields and grid to represent crawled pages. Each item of the grid is associated with some URL of web-server resource and page content that can be crawled or not at the moment.

The filter fields:

- **Site Id** – the unique site identifier; 32 character string. If specified the "Site root URL" field is ignored. Mapped to the **Site_Id** field in the sql request criterion. Mandatory.
- **Site root URL** – the site's root URL that can be used to calculate site Id or/and to find correspondent site; max 128 characters string. If not empty – the md5 checksum is calculated and filled the "**Site_Id**" field before request send. Optional.
- **Resource URL** – template for URL string; 256 characters max string. If empty – the "*" template value assumed for SQL select. Mapped to the **URL** field in the sql request criterion. Optional.
- **Resource Id** – unique resource Id; 32 characters max string. Mapped to the **URLMd5** field in the sql request criterion. Optional.
- **Status** – the select with values and titles: "" – "Any", 0 - Undefined, 1 - New, 2 - selected for crawling, 3 - crawling, 4 - crawled, 5 - selected to process, 6 - processing, 7 – processed. Mapped to the **Status** field in the sql request criterion. Mandatory.
- **Content-type** – the content-type name, 32 characters max string. Mapped to the **ContentType** field in the sql request criterion. Optional.
- **Crawl date from** – the date of resource crawled from; textual representation of the date in format "Y-m-d H:i:s" with calendar popup to get possibility to fill it from calendar window. Mapped to **UDate** field "from" for the select SQL criterion. Optional.
- **Crawl date to** – the date of resource crawled to; textual representation of the date in format "Y-m-d H:i:s" with calendar popup to get possibility to fill it from calendar window. Mapped to **UDate** field "to" for the select SQL criterion. Optional.
- **Max items per page**. Select filled with values 10, 20, 30…100. Mapped to the URL_FETCH **maxURLs** field and the LIMIT for sql query criterion.

Theresults grid columns:

- **Id** – the unique URL Id; 32 character string. Mapped to **urlMd5** field in the response json URL object.
- **URL** – the URL string of resource; 256 characters string max to display, but 4096 possible for copy operation. Flexible width. Mapped to **url** field in the response json URL object.
- **Status** – the URL status name, defined in URL class. Mapped to **status** field in the response json URL object.
- **Crawled** – the URL crawled counter, integer. Mapped to **crawled** field in the response json URL object.
- **Processed** – the URL processed counter. Mapped to **processed** field in the response json URL object.
- **Errors** – list of errors names as list of small descriptions corresponded with the errors bit set mask (defined in the main architecture specification for DC service application). Mapped to **errors** field in the response json URL object.
- **Size** – the resource size, byte. Mapped to **size** field in the response json URL object.
- **Total time** – the total time of crawling and processing. Mapped to **totalTime** field in the response json URL object.
- **Operations** buttons/icons:
    - View – opens the “Resource fields view” dialog/page.
    - Edit – opens the “Resource properties edit” dialog/page.
    - Cleanup – opens the “Resource cleanup” dialog/page.
    - Re-crawl – opens the “Resource re-crawl” dialog/page.
    - Re-process – opens the “Resource re-process” dialog/page.
    - Download – opens the “Resource download” dialog/page.
    - Delete – opens the “resource delete” dialog/page.

On filter submit button form fields send to server-side. Server side script creates request based on cli utility specification, provides the json file created in temporary dir, parse response json and return some data that will be visualized with web-UI.

On top or right of the search activation button the “New” button or activation item need to be placed with the action of open “New resource dialog/page”.

Resource fields view dialog/page

“GR-R-FV”

It visualizes all URL object fields that returned as a result of “URL_FETCH” or “URL_STATUS” requests. There are several types of fields need to be visualized, for example – simple text fields, lists, enumerations, Boolean and so on. The controls need to be chosen to better satisfy the nature and specific feature of information to display. For example, URLs supposes long strings (up to 4K character length), the lists need to show as many items as possible to not break the graphical design, the time and date marks need to be formatted according with common format than can be set up as configuration parameter and so on. The complete list of fields can be taken from jsons examples in appendix B of the http://hierarchical-cluster-engine.com/docs/pdf/DC_public_client_API.pdf document for URL_STATUS and URL_FETCH operations.

Resource fields edit dialog/page

“GR-R-FE”

This modal dialog similar with the “Resource fields view” – visualizes all resource’s properties returned

for URL object as a result of “URL_FETCH” or “URL_STATUS” requests, but gives possibility to edit and update them. It is possible to use for either this dialog the same code with some mode switches, for example – editable/read only or something like that. Key difference – it is possibility to enter/change value and to make update URL object operation. As opposite to “Resource fields view” – this dialog has three buttons – “Cancel”, “Update” and “Help”. The “Cancel” acts the same way as “Close” button, the “Help” button acts exactly the same way and the “Update” button – executes URL_UPDATE operation and then URL_STATUS operation for this URL, closes the modal dialog and updates fields values in the urls/resources list grid. Also, it updates all fields inside the resource’s data container on client-side to give the possibility to display actual updated fields for next “View” or “Edit” actions without reload data and additional requests.

New Resource dialog/page

“GR-R-NW”

This modal dialog similar with the “Resource fields edit” – visualizes all resource’s properties initially set in default values for the correspondent type of user account and gives possibility to edit and create a new URL object representation. All behavior of fields editor the same as for edit dialog but “Submit” action lead to execute URL_NEW request and after that the URL_STATUS if success and open the “Resource fields edit dialog/page”.

Both New and Edit dialog/page need to have a possibility to manage the raw content, processed content instances and to view the extended textual data like HTTP request headers, cookie and response headers. Because the size of raw content and each of processed content instances can to have huge size – the better to not include them in to the visualization and editable controls by default and hide area reserved for visualization until user not opened it. After open action performed – additional URL_CONTENT request executed and returned field’s data visualized.

Resource cleanup dialog/page

“GR-R-CD”

This modal dialog contains fields used to tune resource cleanup procedure. Depends on tuning options chosen it may require URL_UPDATE and URL_STATUS operations. Tuning options sets the State and the Status fields after cleanup:

- State – select filled with: “Enabled”, “Disabled”, “Error”; the “Enabled” selected.
- Status – select filled with: “New”, “Undefined”; the “New” selected.

Additional controls on a dialog: buttons “Cancel”, “Cleanup” and “Help”. The “Cleanup” button starts process in sequence and the URL_CLEANUP, URL_UPDATE and URL_STATUS operations performed. The same as the “Resource fields edit” operation – it closes the modal dialog and updates fields values in the resources list grid. Also, it updates all fields inside the resource’s data container on client-side to give the possibility to display actual updated fields for next “View” or “Edit” actions without reload data and additional requests.

Resource re-crawl dialog/page

“GR-R-RC”

This modal dialog contains warning message: “After re-crawl process resource content will be fetched from web-server and processed by processor according the site configuration. Content will be updated. Real time when re-crawl process will be started depends on many factors.”.

Additional controls on a dialog: buttons “Cancel”, “Re-crawl” and “Help”. The “re-crawl” button starts process in sequence: URL_CLEANUP, URL_UPDATE (set State=0, Status=1) and URL_STATUS operations performed. The same as the “Resource fields edit” operation – it closes the modal dialog and updates fields values in the resources list grid. Also, it updates all fields inside the resource’s data container on client-side to give the possibility to display actual updated fields for next “View” or “Edit” actions without reload data and additional requests.

Resource re-process dialog/page

“GR-R-RP”

Completely the same as the “Resource re-crawl” but warning message is “After re-process resource’s processed content will be updated. Real time when re-crawl process will be started depends on many factors.” And during the URL_UPDATE operation the Status=4 set instead the Status=1).

Resource download dialog/page

“GR-R-DW”

This modal dialog contains fields used to tune resource related contents download procedure. Depends on tuning options chosen it require URL_CONTENT operation with different request fields values. Tuning options sets contents types that can be downloaded:

- Raw content (include fixed by the tidy lib), Processed content, HTTP request, HTTP headers, HTTP cookies, HTTP meta data – checkboxes; the “Processed content” is checked by default.
- Crawled – select filled with: “Last”, “First”, “All”; the “Last crawled” selected by default. Mapped to bits defined for the URL_CONTENT request object as CONTENT_TYPE_RAW_LAST, CONTENT_TYPE_RAW_FIRST, CONTENT_TYPE_RAW_ALL correspondently.

Additional controls on a dialog: buttons “Cancel”, “Download” and “Help”. The “Download” button starts process in sequence and the URL_CONTENT operation then decode all requested contents, create directory with name of resource Id in the temporary directory, store contents of correspondent type as disk file with names: “raw.bin”, “raw.bin.tidy”, “processed”, “request”, “headers”, “cookies” and “meta”; zip this directory as file named as resource Id with .zip extension; delete directory; return zipped data back to browser as regular zip archive MIME type to get possibility for browser to process it regular way. Then close modal dialog.

Resource delete dialog/page

“GR-R-DD”

This modal dialog contains warning message: “After this operation all resource-related data including contents and URLs will be deleted!”.

Additional controls on a dialog: buttons “Cancel”, “Delete” and “Help”. The “Delete” button starts process in sequence: URL_DELETE, close the modal dialog and delete correspondent record from resources list grid.

Sites and resources statistics

"GR-SS"

Sites and resources statistics are the periodical collected data visualization and tracking about number of units like URLs, resources, contents, tags, batches and so on that was processed, collected, analyzed and performed another actions per time unit, total, average and so on. Statistical data represented as summary of counters and items inside the DC service for all hosts and nodes.

Site stats

The site statics collected from the DC service by the SiteFind query per User with interval of time configured in general configuration as a periodic process (by default 5 minutes) and accumulated inside the administration system database with sharding a table per site. Proposed database schema fields:

URL stats

Displays the page with filter and results data grid. The filter has date range fields with popup calendar to set time range and the module name selector (select). The results grid columns: Date, Module Name, Comment.

Batches

To be continued…

Processing

Add scraping template dialog/page

To be continued…

Delete scraping template dialog/page

To be continued…

Configuration

“GR-CF”

This page contains select control to choose type of ini-file and multiline textbox to edit the ini-files of DC service. The path for each ini-file is configured by main settings. Files are managed and listed in select: “dc-admin.ini”, “dc-admin_log.ini”, “dc-client.ini”, “dc-client_log.ini”, “dc-daemon.ini” and “dc-daemon_log.ini”; by default the “dc-daemon.ini” selected. If server-side script has no permissions to modify ini file – the “Save” button is not active. If selected another file and current textbox text was changed the confirmation modal dialog appears with text: “The configuration data was changed and will be lost. Press “Save” button to store changes.” and buttons: “Ok” and “Cancel”. The behavior is regular. Additional controls – buttons: “Cancel” (refreshes data from source), “Save”, and “Help”. The behavior is regular.

Service

“GR-SV”

This page contains management items for DC service process management and visualizes current state. The representation control is multiline text field that displays the results of a Linux “top” command with parameters (full command line configured by main configuration settings); supposes some several lines with information about the DC’s process state inside the Linux OS. When the dialog opened the command executed and textual output is filled. Additionally, the admin request type “STATE” is used for only the “AdminInterfaceServer” class to get stat data. If admin request fault information message need to be displayed in the modal popup window. Additional controls on a dialog – buttons: “Start”/”Stop” (depends on service state), “Kill” and “Help”. “Start” and “Stop” buttons are displayed depends on a service state – not started and started correspondently and uses the start and stop command lines (configured in main configuration). The “Kill” button uses the kill command line also configured in main configuration; this button action need the confirmation modal dialog with text “This operation will kill the DC’s service as process! All managed crawling processes and management data will be lost! Contimue?” and buttons “Ok” and “Cancel”. After execution of correspondent commands the stdout and stderror printouts are displayed in the “Traceback log area; “top” command need to be executed and textbox need to be refreshed with results. Behaviors of rest are regular.

Users and accounts

"GR-US"

It is general user's management CRUD in free form. There are three types of users by general permissions set:

1. Administrator – can to view and to modify any data that belongs to any users. Can to be owner of sites as type "User".
2. User – site owner, relations – one user can manage several sites and each site has one owner. Can to view and to modify only own data.
3. Viewer – can only view data of sites of users that has allowed/assigned for him.

Account block

Add possibility to block user account. If user is blocked – no login possibility and notification message reports about of cause of that blocked state. For admin UI block action provides the text field to enter notification message.

Account aging

The aging is a possibility to perform automated change of the state of the user's account from "Active" to "Blocked" and then to delete it and all related data structures including sites inside the DC service. It is supposes the support of TTL value that initialized when account created and can be updated by user with correspondent permission right. All automated processes need to be performed centralized way by the periodic processes automation ("GR-PP"). The "Account delete" operation needs to perform delete all user's sites that are belongs for that user only. If site belongs to another users - only relation with this user need to be removed. If delete all user's sites operation on DC service is not finished successfully it need to be repeated. Number of tries is configured in the main configuration. If number of tries exceeded – notification message need to be sent for administration e-mail and account stays in blocked state (not deleted).

Account types

Support of account types: "paid" and "free". Additions for registration, creation, filters and another kind related operations.

Account limitations

The accounts limitations it is a based on account types – set of templates for user's account, site and another related functional entities. The aim is to have a possibility to limit all possible resources for different account types. For the Account entity they are: max TTL in minutes and max sites number. For the Site entity there are main crawling and processing limits from DC service like: MaxURLs, MaxURLsFromPage, MaxResources, MaxErrors, MaxResourceSize and additional from the management system like total max size of data including content on disk and in the database key-value storage (the estimation metrics calculation algorithm need to be designed later). The functionality includes the possibility to set initial values from named template (stored in the main configuration) and to change them. To simplify the UI - fields that cannot be changed for limited account will not be updated during the regular operation with warning message, but not block operation at all.

Notifications

Displays the page with filter and results data grid. The filter has date range fields with popup calendar to set time range, type selector with values: e-mail, http. The results grid columns with notification item fields. Functionality supposes full CRUD for items, detailed definition to be continued.

Data delivery

Displays the page with filter and results data grid. The filter fields: Site Id input with autofill possibility for my sites (by site root URL domain pattern or URLMd5 Id); selector (select) of type with values: None, Archive, http, ftp and state field with values: Active, Inactive. The results grid columns with data delivery item fields. Functionality supposes full CRUD for items, detailed definition to be continued.

General possibilities – to collect archives of data on one point host where TR service's site located is and optionally send per item or per archive to the destination like http or ftp by customized request string. Supposes UI to setup and configure delivery item per site with checks to not create duplicated items or incompatible for the same site. Each item supposes own schedule and implementation of isolated execution by process spawning, logging and possibility to limit number of spawned processes simultaneously running in time.

Periodic processing automation

“GR-PP”

It is server side module that supposes automated start by cron. The purpose is to involve and execute some functionality configured to be used by internal schedule as periodic processing. Suppose usage of the schedule table in the main database as well as common call method for functional processing modules implemented as a part of the server side system. After execution record in the schedule table stored on limited period (limitation defined in the main system configuration). Each record can to have comment that can be set after execution of the functional processing module. This way some textual message can be noted, for example about number of processed items, some names or Ids that can to help to get track of events in the system. The module management performed by main system configuration (list of active modules and their schedule).

Modules

Displays the page with filter and results data grid. The filter has date range fields with popup calendar to set time range and the module name selector (select). The results grid columns: Date, Module Name, Comment.

This subsystem will be used to schedule the statistical data collector module, the notification messages module, the account aging module and so on.

History

Displays the page with filter and results data grid. The filter has date range fields with popup calendar to set time range and other fields. Description to be continued.

Configuration option parameters

"GR-CO"

Server-side scripting need to have common general options that can be configured by separated dedicated configuration file/script and can be changed as a part of installation procedure or later in runtime period. List of configuration options:

- **Date time visualization format** – set as string in PHP language format specification, for example "Y-m-d H:i:s"
- **API Utilities dir** – directory for API utilities to use to make full path to run DC management utilities.
- **Site errors types names** – String names enumeration array for site errorMask visualization.
- **Site state names** – string names enumeration for the site state title visualization.

Service configuration

“GR-SC”

This page displays vertically sectioned grid for each of service’s internal classes. Each class lists own configuration variables as name-value pairs in columns. Third column with the “Update” button gives possibility to open simple input dialog with one field and two buttons “Ok” and “Cancel”. If “Ok” is pressed – the configuration update request used to change current value and refresh grid with new values performed. The configuration operation requests performed by the dtm-admin.py call with GET and SET commands, see the dc-daemon_config_vars.sh script as example of get operation for all classes. The scripts templates for this operation need to be defined in the general configuration to get possibility easily to change list of classes and so on.

Service state

“GR-SS”

This page displays vertically sectioned grid for each of service’s internal classes. Each class lists own statistical variables as name-value pairs in columns. Page is current snapshot of values only with refresh button on the top.

Service statistics

"GR-ST"

This page displays graphical representation of the time tracked state variable. User can to get it only by reference from the "SS" page. On top of the page filter gives the possibility to choose variable name as pair of class and variable name (list), the report type:

- Natural values;
- Moving average;
- Natural and Moving average;
- Natural delta;

(list) and the time range by set of different methods:

- Simple "from-to" range with calendar popup;
- Last day;
- Last week;
- Last month;

With possibility to shift backward and forward by chosen standard time range (day, week and month) as well as custom time range with precision in minutes.

This functionality supposes accumulation of the service's stats in the dedicated database on side of web administration interface. To collect the variables information common database structure will be used with fields: CDate(Timestamp) and Data(BLOB). The "CDate" field is unique and primary key. The "Data" field contains response of regular GET command in json format. Accumulation script start is managed by cron and makes one snapshot of statistical variables per start. The number of variables as well as number of classes that can be managed and tracked is different and can be changed during service usage, so if value is not present in the json data of some record it needs to be treated as absent or zero value to minimize negative influence on graphical representation.

Configuration

"GR-CF"

E-mail notification templates

Any notification message need to have named template stored and managed with full CRUD in the main database. To send notification message the target functional module need to get template context for the subject and the body and substitute correspondent macro.

Messages translation

Depends on multi-language principles – the possibility to enter and to edit multi-language messages contents.

Appendix I

Requests and responses protocol

Sites list

Site fields view

Site cleanup

Site re-crawl

Site delete

Stats

Site fields view

Service
Stop