General requirements on web user interface for public Tags Reaper service

The web user interface is common UI for execution of most important tasks with TR service, make operations, and visualize statistics and so on without explicit authorization or registration in demo mode. Also, set of regular pages with basic functionality for account registration, password recover, login, logout and so on. The architecture is typically web UI with client-side and server side parts each are implemented on one framework library. All dependent requirements the same as for the administrative UI defined in the DC_web_ui_general_requirements.docx document.

Since the common not authorized pages needs in some hidden authorization the unique user Id generator algorithm based on the client-side browser's data criterions (like IP address, graphical resolution, user agent, time mark of the first visit of TR web site and so on) need to be defined. Also, some leasing on fixed time period principles can be used. All request URIs and forms fields values need to include long length string unique identifier that encodes some of key identification criterions and must be resolved in to the integer user Id used by the backend and the DC service internally with protected by some cipher well-formed unambiguous algorithm or cross-reference table without the possibility to reconstruct the integer user Id by string request or session Id used in the URL.

All possible not authorized requests that will lead to interaction with the DC service need to be limited in time. The limitation can be related with the abstract or real not authorized user some kind represented inside an administration structures or/and database. In case of limit is reached the request needs to return standard error that need to be handled by client side with correct information warning message at least.

All public pages available without an explicit authorization need to be multi-language with the possibility to switch the language of interface easy way in one click. The messages definitions can be done on any principles as include files, database and so on. Only UTF-8 encoding need to be supported.

Demo test start page and redirect behavior for registration

"GR-SP"

Since the service site has many pages that are cross-linked several requests and actions with active demo-test functionality that supposes an interaction with the DC service need to be indirectly authorized. Also user needs to have a tiny registration that must to be done once. To identify the user as a device client system the string identifier need to be generated and provided in any protected request. Also, identifier stored on server side as a session value. For fast tiny registration this identifier generated with small TTL value, for example two days. This identifier will be used for a public API calls also.

If user is not registered – i.e. the identifier is not provided with request data, not set in server-side session variables, provided or set but not found as registered or found but is expired, user must be redirected on tiny registration form. If identifier expired the session variable value deleted. The registration form can be filled directly or using a public accounts like gmail, FB, Hotmail and so on as an alternate way to pass registration form. Mandatory registration form fields are: e-mail address, name and company. After complete form filling and submit user must be redirected on target page with string identifier added to the request data and set as a session variable value automatically.

Single resource processing page

"GR-SR"

The single resource processing page is functionally complete for demonstrate the crawling and/or the processing (scraping) of one resource. The resource can be specified by user as the URL link or as the complete web page body formatted in HTML. The common functionality need to be split on three stages:

1) request form input,
2) request processing progress,
3) results data visualization.

Request form input shows all fields that need to be filled by user to make request and activation element (the button or another kind) to make request and start processing. Request processing progress visualization need to display some progress animation until timeout or request finish state reached. In case of timeout of AJAX request is reached – correspondent error message in common messages design need to be displayed with the possibility to repeat action. The activation element needs to be disabled on request processing time to avoid multiple execution of the same request. After timeout reached the initial stage form 1) need to be shown with all entered data filled. In case of request error response received, the behavior is the same but message text. If response is success and correspondent data is received – the results data visualization form shown.

All kind of request interaction needs to be protected by captcha.

Request form

The request form has different set of fields depends on source data type (URLs or raw html content) that can be switched by radio button, tab or another way. By default is “URLs list” multi-line – three visible long length text input control (up to 128 characters or more, cause URLs are often very long) named as “Resource URL(s)”. All another possible fields are hidden and set in default state, but can be shown by some activation item depends on visualization tool (tabbed dialog, sliding panes, scrolled grid, hidden pane, and so on). Request form fields filling need to be saved from request to next request and must be pre-filled in any state of request (timeout or error).

The “Resource URL(s)” textarea can to have more than one URL. The URLs number needs to be identified (parsing delimited with new line) and validated on client and server side.

All kind of validation and limitations can be done for all fields on client and server side. The limits for validation and checks need to be set as arrays in main configuration or the same way to have one point change access with administration account. Only one set of limits and default values need to be defined in the system for that kind of interface and titled as “Not authorized requests limits” and “Not authorized requests defaults”.

Any kind of client or server side validation error needs to lead to displaying of the error or the warning message with detailed explanation. In case of cause of validation failure request form field can be identified – it need to be done active, focused and name highlighted to have a possibility of a clear understanding from user.

Request processing progress

This is just visualization tool that shows some animation while request time is going on. No reciprocal interaction with actual processing supposed. When request is finished, the progress animation hides out.

Results data visualization

The results data visualization view consists on multiline textarea filled with the raw HTML data (and if it was input request data also) and some kind of two column grid filled with tags names and values detected as a result of the scraping. Under the grid some area filled with extended textual information about the HTTP request, timing and so on. (Complete set of possible data need to be taken from the real-time request response json defined in DC_public_client_API.docx).

Also, the possibility to return to the request form and to make next one request required.

Multi resource processing page or site-processing

"GR-MR"

The multi resource processing page is functionally complete for demonstrate the crawling and/or the processing (scraping) of the Site in terms of the DC service internal objects.

The same way as the "Single resource processing page" it consists of three common functionality stages:

1) request form input,
2) request processing progress,
3) results data visualization.

But, cause the request initiate more complex processing that lead to create the DC-Site structure and the DC service continue all regular processing in background – the management of processing bit more complex.

Request form input shows all fields that need to be filled by user to make request and activation element the same way as for the "Single resource processing".

Request processing progress visualization need to display some progress animation until timeout or request finish state reached. In case of timeout of AJAX request is reached – correspondent error message in common messages design need to be displayed with the possibility to repeat action. The activation element needs to be disabled on request processing time to avoid multiple execution of the same request. After timeout reached the initial stage form 1) need to be shown with all entered data filled. In case of request error response received, the behavior is the same but message text. If response is success and correspondent data is received – the results data visualization form shown.

All kind of request interaction needs to be protected by captcha.

Request form

The set of fields includes only the required for the SiteNew operation (see the SITE_NEW operation request and support json). The Site_Id needs to be generated include information about user unique Id cause simultaneous multi-user usage. The same way, all fields but the list of root URLs must be set by default and hidden with easy possibility to show up in one click. The root URLs list is multi-line – three visible long length text input control (up to 128 characters or more, cause URLs are often very long) named as "Root URL(s)". All another possible fields are hidden and set in default state, but can be shown by some activation item depends on visualization tool used (tabbed dialog, sliding panes, scrolled grid, hidden pane, and so on). Because the request produces SiteNew operation and the DC-Site object, only one site per unauthorized user can be created. So the SiteStatus operation needs to be done using the generated unique user's string identifier and server-side session stored DC-Site_Id before execution of the SiteNew. If site already exists – the request form is filled with the current values of the DC-Site fields from the SiteStatus operation response. If site not exists – the SiteNew operation performed. The DC-Site need to be created using special user's account and site type "Temporary" with the TTL defined in global TR service settings. The TR service itself manages this kind of sites and complete removes them and their data if TTL expired. So, the regular situation for user is refresh this page and get ready to

create site or data from site that is already exists and in some progress state. User must to have no possibility to identify the DC-Site Id, TR's numeric user Id or another kind of internal DC's and TR's data identifiers. So, all data that need to be represented as site's fields need to not to have Ids in visual or hidden form on client side and only can be completed with them by resolving of the textual user'd Id of TR service.

The "Root URL(s)" textarea can to have more than one URL. The URLs number needs to be identified (parsing delimited with new line) and validated on client and server side.

All kind of validation and limitations can be done for all fields on client and server side. The limits for validation and checks need to be set as arrays in main configuration or the same way to have one point change access with administration account. Only one set of limits and default values need to be defined in the system for that kind of interface and titled as "Not authorized requests limits" and "Not authorized requests defaults".

Any kind of client or server side validation error needs to lead to displaying of the error or to the warning message with detailed explanation. In case of cause of validation failure request form field can be identified – it need to be done active, focused and name highlighted to have a possibility of a clear understanding from user.

Request processing progress

This is not just visualization tool that shows some animation while request time is going on. According with the timing that can be changed in real-time by user as a hidden additional property of the "Results data visualization" – it need to be refreshed. The refresh periodic need to have minimal limitation defined in general TR's service configuration and set by default as 15 sec.

Results data visualization

The results visualization representation needs to be done in form of some kind grids possible subdivide the page on two areas. First area – it is site's stats data counters, that are grouped to have better visual representation. All counters can be taken from the DC-Site status operation result. Possible groups are:

- Status: State, Iteration, ErrorMask (with detailed description in popup), Errors, Size, AVGSpeed
- Resources: NewURLs, CollectedURLs, DeletedURLs, Resources, Contents
- Timing: CDate, TcDate, UDate, RecrawlDate, RecrawlPeriod
- Limits: MaxURLs, MaxURLsFromPage, MaxResources, MaxErrors, MaxResourceSize, RequestDelay, ProcessingDelay, HTTPTimeout,

The second area represents the resources state and is a grid with columns filled from the URLFetch request with criterions to select only this site URLs, without tags data in all state that are sorted by CDate by default and no more than default limit defined in the general TR service configuration, set by default as 20. The grid need to have both scrolls to fit the page width and to not to show more than five records.

Also, the possibility to return to the request form and to make next one request required. Because the different state of the DC service at moment of time possible, after return to the request form page it is pre-filled with default values or with the result of the SiteStatus response as well as the SiteNew or the SiteUpdate operation performed.

In case of site already exists two more operations are need to be available – to suspend and to cleanup it. The actions activator items can be a buttons or another kind. The suspend action performs the SiteUpdate request to the DC service for the correspondent site and set the DC-Site.State in SUSPEND(3) value. If site exists and is suspended, the “suspend” item need to be changed to “activate” and to perform the SiteUpdate request to the DC service and set the DC-Site.State in ACTIVE(1).

The cleanup operation needs to performe the SiteCleanup operation. Possible the site need be suspended before and some delay from 15 to 60 seconds need to be done.

After any kind of that extended operations the SiteStatus operation done as regular page refresh.

Logging and tracking

"GR-LT"

All kind of not authorized and authorized user's activity need to be logged and tracked by the service's features inside. So, some dedicated schema needs to be supported and managed as default behavior of the TR service. Proposed structure is two level data representation: the table with only time stamp, URL and UId and the table with complete form data compressed. Management of those structures needs to include the search page and periodical cleanup. This functionality extends the default web administration interface as addition that became available as result of union of web administration backend and TR service's backend. Some configuration and site's script deployment settings possible need to be done once.

The search page

The search page needs to have filter with fields: date range, user unique Id and operation type. The operation types can be identified by first 64 characters of URU (text index for correspondent field) and some set of pre-defined by the operation sense can be provided to choose as value. The operation type field can be just single line text field with auto-complete popup with list of some pre-defined named operations. Each operation supposes some unique URI parameters or path pattern.
The result of the search request is records set visually represented as some kind of grid with pagination. Grid columns the same as request form fields with possibility to order by. The line of grid is clickable and opens the show **data page** as separated. The maximum number of records or time range interval also can be limited with general configuration settings. Default values are one month time range, 100 records per page and 1000 records per request pagination shift possibility.

The data page

The data page is visualization of the data that is associated with the user's URI that is used for action from client side and identified with unique UId. Common default visualization is multiline textarea. Depends on specific actions and data type format identification some additional transformation can be done as json or xml formatting, make URL in content clickable and so on. All possible additional settings of visualization form can be changed as additional visualization parameters hidden by default.

Periodical cleanup

The action need to be performed as single session start module that scheduled and spawned by regular periodic processes management system of web administration. The parameters are the TTL of records and number of items that can be deleted per one start. Parameters are configured by main configuration. For more detailed description see specification in the DC_web_ui_general_requirements.docx.