Workbooks.com

Workbooks API and the Process Engine

www.workbooks.com

Agenda

- Introduction
  - What you can do with the API
- How to use the API
  - External access or the Process Engine?
- The Process Engine
- Get, Create, Update and Delete with the API
- Process types
  - Scheduled Processes, Process Buttons, Web Processes
- How Processes are run

Introduction: Why use the API?

Some examples:

- Email-to-Case: monitor a mailbox, create and update support cases
- MailChimp, Constant Contact, dotMailer, HubSpot …
- Sagelink, OneSaaS
- Outlook Connector
- Mobile Client
- Creating many order line items to reflect a delivery schedule
- Calculate field values
- Sales Lead categorisation, analysis and allocation

API not required:

- Simple lead or case capture (use web-to-case)
- Generating a PDF (use PDF templates) or a templated email
- Simple workflow using custom page layouts and assignment
- Data Import
- Reporting

Introduction: What is the API?

- API – 'Application Programmatic Interface'
  - i.e. an interface enabling software to interact with Workbooks.
- A set of web services delivered over SSL (https)
- Stateless, client/server
- RESTful – create, read, update, delete
- Batched
- JSON

How to call the API:
Wire Protocol or Binding?

- Wire Protocol = JSON-encoded HTTP requests
  - Can be complex
  - Documented at
    - http://www.workbooks.com/api-developer-guide
  - No restriction on which language is used.
- Bindings hide much of the complexity
  - PHP binding on github (please feel free to contibute) at
    - https://github.com/workbooks/client_lib/tree/master/php
    - PHP used by the process engine
      - PHP is widely-understood and open-source.
      - Lots of systems have documented PHP APIs.
  - Others to come.

How to call the API: Where to run your code?

- Workbooks-hosted
  - The “Process Engine”.
  - Simpler, automates invocation, authentication and logging.
  - Not available if you are using a ‘Free’ licence.
- Externally
  - Host your code yourself.
  - Connect to Workbooks explicitly over HTTPS.
  - Authenticate using API Key or Username, password and database ID.
  - A little more flexible.
  - The API is available to all Workbooks users.
- The Process Engine is used in this presentation for simplicity.

Introducing the Process Engine

- Some glossary:
  - Script – a unit of code.
  - Processes invoke Scripts.
  - Process types:
    - Scheduled Process
    - Web Process
    - Process Button / on-Save Process
    - Test Process
  - Processes run on behalf of a user, with constraints.

Example: Hello World!

- Process Engine example - hello world as a web process

Example: Hello “name”

- php in 5 minutes
  - tags
  - variables
  - output
  - function calls
  - where to find out more

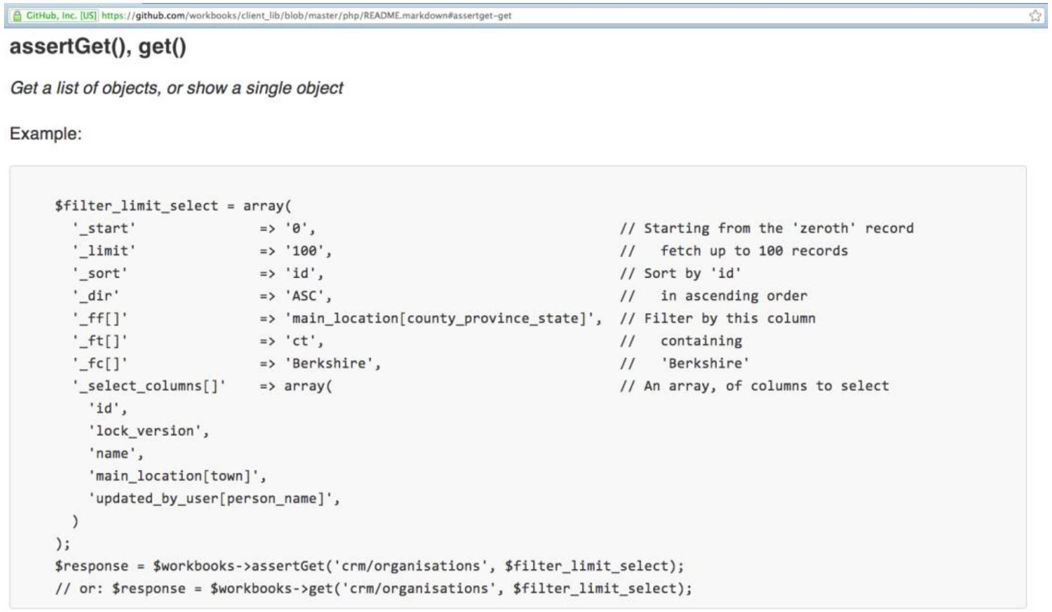
- Show a form asking for a name.
- Echo that name back.

The runtime environment

- $workbooks
  - A handle to a variety of useful methods.
- Logging
  - Automatic logging for API
  - Inputs, environment variables, parameters…
- Parameters and including scripts

Using the API: Get records

- Retrieve Parameters: all optional:
  - start, limit
  - sort, direction
  - filter
  - column selection (speed)
- Response:
  - an array of hashes
- Errors:
  - **assert**Get - raise exception



GitHub, Inc. [US] https://github.com/workbooks/client_lib/blob/master/php/README.markdown#assertget-get

assertGet(), get()

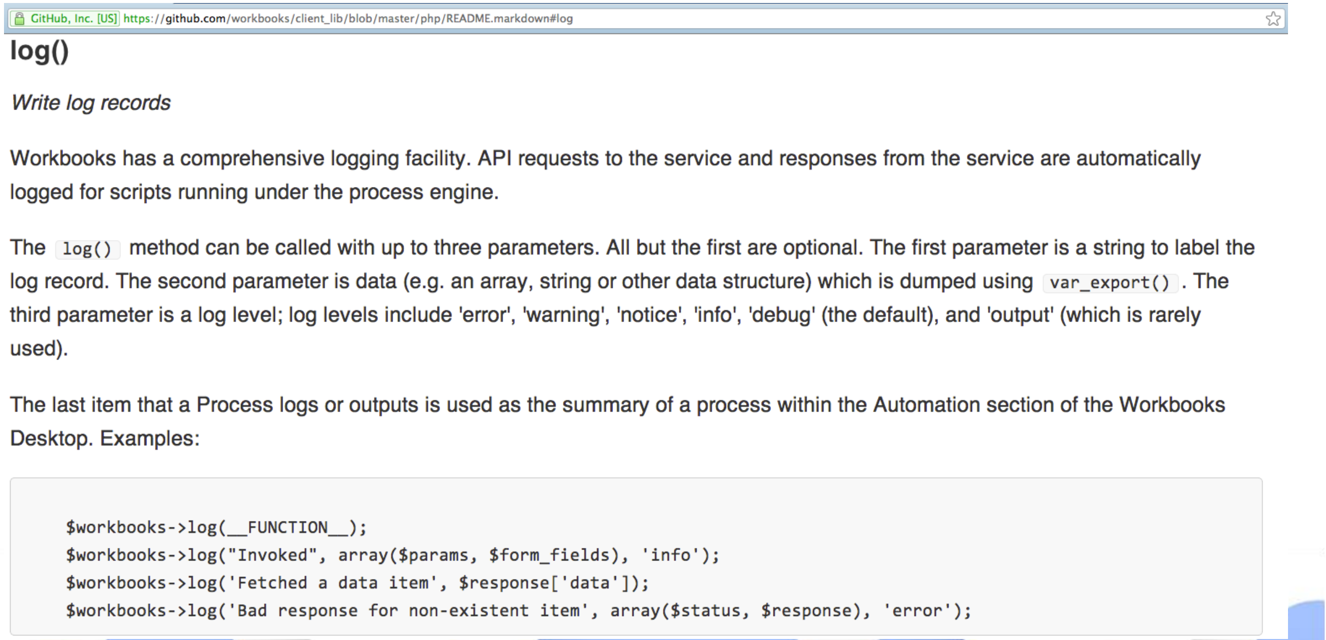
Get a list of objects, or show a single object

Example:

```
$filter_limit_select = array(
  '_start'               => '0',                                   // Starting from the 'zeroth' record
  '_limit'               => '100',                                 //   fetch up to 100 records
  '_sort'                => 'id',                                  // Sort by 'id'
  '_dir'                 => 'ASC',                                 //   in ascending order
  '_ff[]'                => 'main_location[county_province_state]', // Filter by this column
  '_ft[]'                => 'ct',                                  //   containing
  '_fc[]'                => 'Berkshire',                           //   'Berkshire'
  '_select_columns[]'    => array(                                 // An array, of columns to select
    'id',
    'lock_version',
    'name',
    'main_location[town]',
    'updated_by_user[person_name]',
  )
);
$response = $workbooks->assertGet('crm/organisations', $filter_limit_select);
// or: $response = $workbooks->get('crm/organisations', $filter_limit_select);
```

Logging

- API calls log automatically.
- Use $workbooks->log() often.
- Last line treated as the "summary".


GitHub, Inc. [US] https://github.com/workbooks/client_lib/blob/master/php/README.markdown#log

log()

Write log records

Workbooks has a comprehensive logging facility. API requests to the service and responses from the service are automatically logged for scripts running under the process engine.

The `log()` method can be called with up to three parameters. All but the first are optional. The first parameter is a string to label the log record. The second parameter is data (e.g. an array, string or other data structure) which is dumped using `var_export()`. The third parameter is a log level; log levels include 'error', 'warning', 'notice', 'info', 'debug' (the default), and 'output' (which is rarely used).

The last item that a Process logs or outputs is used as the summary of a process within the Automation section of the Workbooks Desktop. Examples:

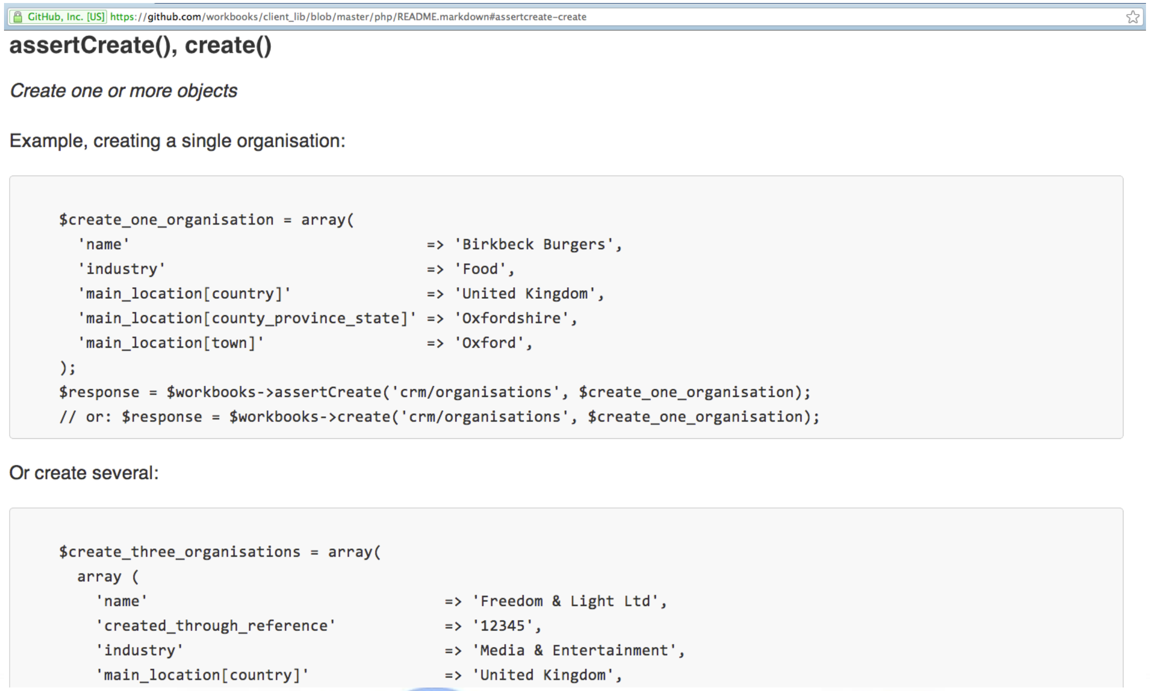
```
$workbooks->log(__FUNCTION__);
$workbooks->log("Invoked", array($params, $form_fields), 'info');
$workbooks->log('Fetched a data item', $response['data']);
$workbooks->log('Bad response for non-existent item', array($status, $response), 'error');
```

More about fetching data

- PDF
- Report – as CSV or as a hash
- Metadata API
  - Discover the set of fields, including custom fields
- Do not assume field order or record order without sort
- Field sizes are important
- Filters
  - Comparison operators
  - OR can be slow
  - Complex comparisons

Create

- Batch
  - up to 100 objects in a single request.
- Response
  - id, lock_version
- Create main objects
- Create relationships
- Picklists and IDs



GitHub, Inc. [US] https://github.com/workbooks/client_lib/blob/master/php/README.markdown#assertcreate-create

assertCreate(), create()

Create one or more objects

Example, creating a single organisation:

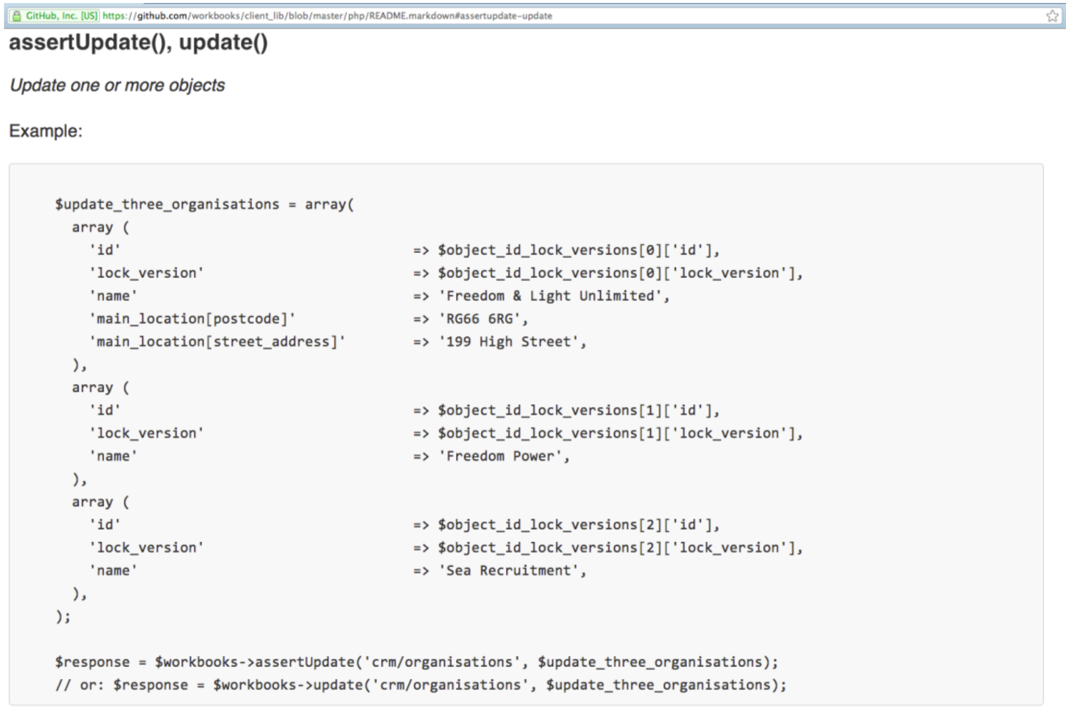
```
$create_one_organisation = array(
  'name'                                 => 'Birkbeck Burgers',
  'industry'                             => 'Food',
  'main_location[country]'               => 'United Kingdom',
  'main_location[county_province_state]' => 'Oxfordshire',
  'main_location[town]'                  => 'Oxford',
);
$response = $workbooks->assertCreate('crm/organisations', $create_one_organisation);
// or: $response = $workbooks->create('crm/organisations', $create_one_organisation);
```

Or create several:

```
$create_three_organisations = array(
  array (
    'name'                                 => 'Freedom & Light Ltd',
    'created_through_reference'            => '12345',
    'industry'                             => 'Media & Entertainment',
    'main_location[country]'               => 'United Kingdom',
```

Update

- Required:
  - id
  - lock_version
  - _can_modify
- Returns
  - An array of affected objects/errors
- Stale object error:
  - lock_version out of date



GitHub, Inc. [US] https://github.com/workbooks/client_lib/blob/master/php/README.markdown#assertupdate-update

assertUpdate(), update()

Update one or more objects

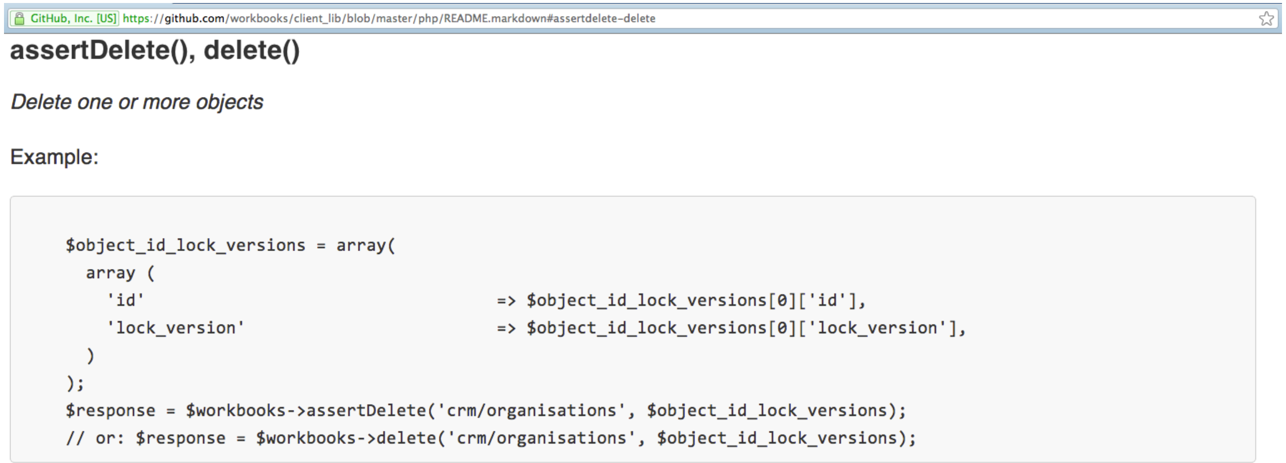
Example:

```
$update_three_organisations = array(
  array (
    'id'                                      => $object_id_lock_versions[0]['id'],
    'lock_version'                            => $object_id_lock_versions[0]['lock_version'],
    'name'                                    => 'Freedom & Light Unlimited',
    'main_location[postcode]'                 => 'RG66 6RG',
    'main_location[street_address]'           => '199 High Street',
  ),
  array (
    'id'                                      => $object_id_lock_versions[1]['id'],
    'lock_version'                            => $object_id_lock_versions[1]['lock_version'],
    'name'                                    => 'Freedom Power',
  ),
  array (
    'id'                                      => $object_id_lock_versions[2]['id'],
    'lock_version'                            => $object_id_lock_versions[2]['lock_version'],
    'name'                                    => 'Sea Recruitment',
  ),
);

$response = $workbooks->assertUpdate('crm/organisations', $update_three_organisations);
// or: $response = $workbooks->update('crm/organisations', $update_three_organisations);
```

Delete

- Required:
  - id
  - lock_version
  - _can_delete



GitHub, Inc. [US] https://github.com/workbooks/client_lib/blob/master/php/README.markdown#assertdelete-delete

assertDelete(), delete()

Delete one or more objects

Example:

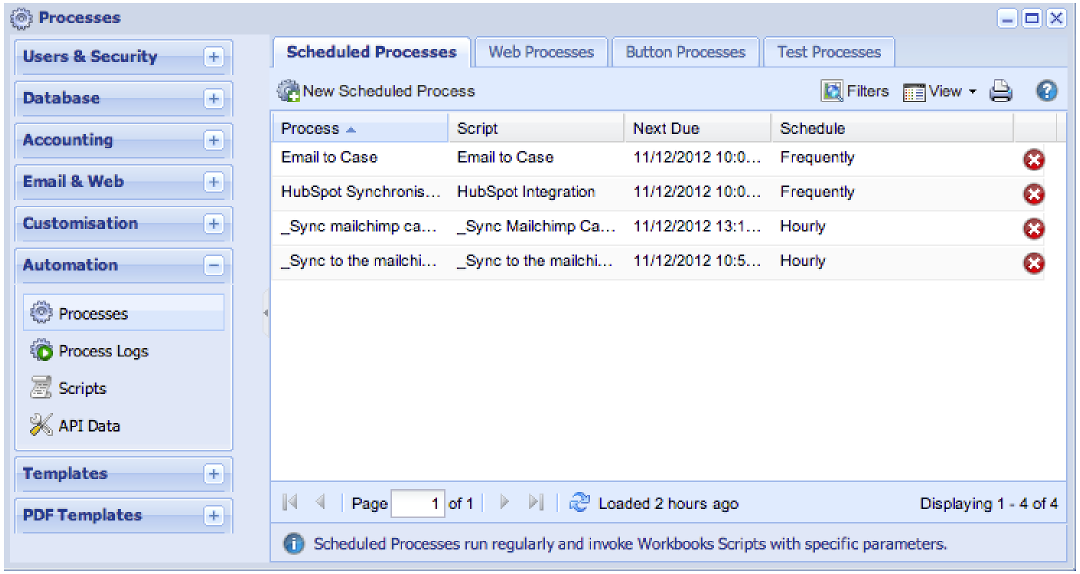
```
$object_id_lock_versions = array(
  array (
    'id'                                  => $object_id_lock_versions[0]['id'],
    'lock_version'                        => $object_id_lock_versions[0]['lock_version'],
  )
);
$response = $workbooks->assertDelete('crm/organisations', $object_id_lock_versions);
// or: $response = $workbooks->delete('crm/organisations', $object_id_lock_versions);
```

Other Useful APIs

- Sending email
  - e.g. send a report.
  - uses the user’s Workbooks email settings to deliver.
- API Data
  - Useful to hold process ‘state’ between invocations.
  - Do work in small batches.

Processes

- Processes invoke Scripts
- Scheduled
- Web
- Button (& onSave)
- Report
- Test



Process	Script	Next Due	Schedule
Email to Case	Email to Case	11/12/2012 10:0...	Frequently
HubSpot Synchronis...	HubSpot Integration	11/12/2012 10:0...	Frequently
_Sync mailchimp ca...	_Sync Mailchimp Ca...	11/12/2012 13:1...	Hourly
_Sync to the mailchi...	_Sync to the mailchi...	11/12/2012 10:5...	Hourly

Scheduled Processes run regularly and invoke Workbooks Scripts with specific parameters.

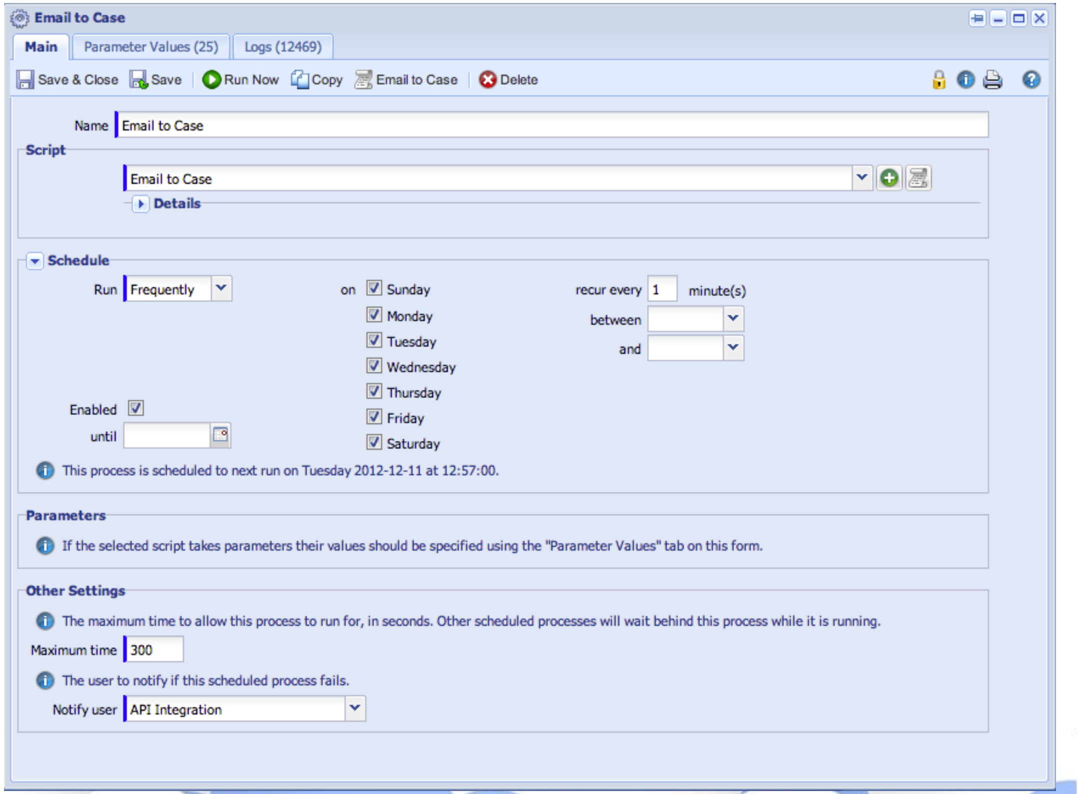
- Processes run as a user
  - User requires DB access.
  - Capabilities matter.

Test Process

- Useful for debugging simple scripts
- Created when first used
- Prompt for parameters

Scheduled Process

- Restriction: only one per database can run at a time
  - Duration should be small
- Exit Code matters
  - 0 => OK
  - 1 => Retry later
  - 2 => Failure
- Upon failure:
  - Process disabled
  - User notified



Name: Email to Case

Script

Email to Case

Schedule

Run: Frequently

on: ☑ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday

recur every 1 minute(s)

between

and

Enabled ☑

until

This process is scheduled to next run on Tuesday 2012-12-11 at 12:57:00.

Parameters

If the selected script takes parameters their values should be specified using the "Parameter Values" tab on this form.

Other Settings

The maximum time to allow this process to run for, in seconds. Other scheduled processes will wait behind this process while it is running.

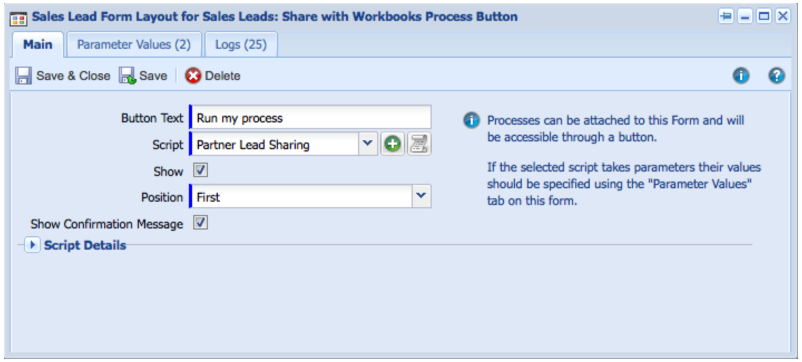
Maximum time: 300

The user to notify if this scheduled process fails.

Notify user: API Integration

Process Button

- Added via Custom Form Layout.
  - Automation tab
- Buttons appear on record toolbar.
- Process invoked after successful validation and save of the record.
- Process completes before form reloads.
- Form fields passed to the process in $form_fields array.
- Summary shown as an alert message unless turned off.
- Button processes can be attached to the record save action to run every time.



Sales Lead Form Layout for Sales Leads: Share with Workbooks Process Button

Button Text: Run my process

Script: Partner Lead Sharing

Show ☑

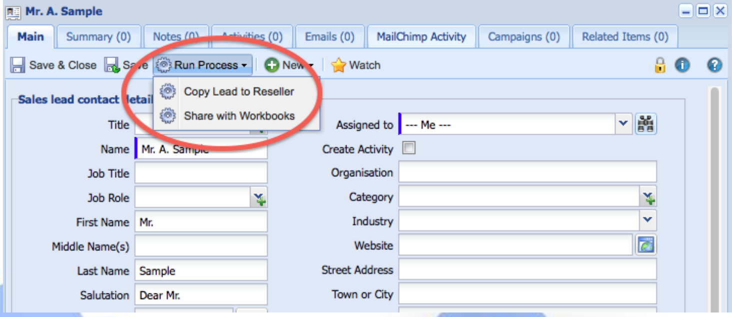
Position: First

Show Confirmation Message ☑

Script Details

Processes can be attached to this Form and will be accessible through a button.

If the selected script takes parameters their values should be specified using the "Parameter Values" tab on this form.



Mr. A. Sample

Sales lead contact details

Title:

Name: Mr. A. Sample

Job Title:

Job Role:

First Name: Mr.

Middle Name(s):

Last Name: Sample

Salutation: Dear Mr.

Assigned to: --- Me ---

Create Activity □

Organisation:

Category:

Industry:

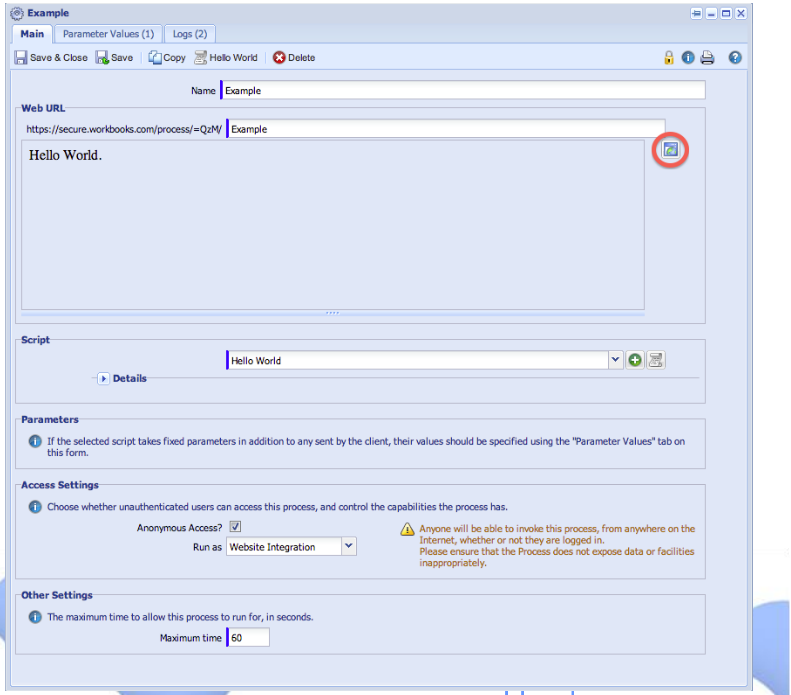
Website:

Street Address:

Town or City:

More on Web Processes

- Access Settings
  - Can be used to permit access
- Process output shown in preview ‘iframe’ on form
  - Or click on button circled to open in another window and capture URL
- Headers
  - Set headers prior to any other output
  - Use $workbooks->header()
  - e.g. to set cookies
- Output streamed to client as it is sent by the script
- Output is UTF-8, be sure to escape it: use ***htmlentities()***



Name: Example

Web URL

https://secure.workbooks.com/process/=QzM/ Example

Hello World.

Script

Hello World

Details

Parameters

If the selected script takes fixed parameters in addition to any sent by the client, their values should be specified using the "Parameter Values" tab on this form.

Access Settings

Choose whether unauthenticated users can access this process, and control the capabilities the process has.

Anonymous Access? ☑

Run as: Website Integration

Anyone will be able to invoke this process, from anywhere on the Internet, whether or not they are logged in.
Please ensure that the Process does not expose data or facilities inappropriately.

Other Settings

The maximum time to allow this process to run for, in seconds.

Maximum time: 60

Other Process Types

- Recent releases of Workbooks have added various options
  - You can attach a process button to a report.
  - Run a process on a list of items (like bulk update).

How the Jail runs processes

- Processes are run within a sandbox, separate from the main Workbooks service.
- Passed to the process each time it's run:
  - Scripts, included scripts, parameters, HTTP parameters, environment variables
  - One special script: workbooks_api.php
  - Each included script is run in turn, with the main script run last.
- Processes only have write access to their TMPDIR.
- Processes authenticate automatically back to Workbooks using credentials passed when they are invoked.

Jail Resource Constraints

- An alarm timer limits the process to its allocated maximum time.
- If ‘Requires External Access’ is set, firewall ruleset is more open:
  - ICMP
  - DNS
  - HTTP, HTTPS
  - IMAP, IMAPS, POP3, POP3S
  - Database (MySQL, MSSQL default ports)
- Specifically not SMTP: use Workbooks’ email API instead.
- Memory, disk usage, process limits all enforced
  - Receive a SIGTERM if memory limit exceeded.
- Workbooks recommends that processes do their work in small batches and checkpoint if required.

Support

- The API changes from time to time
  - Features are added, e.g. the proportion of Workbooks which is accessible via the API increases.
  - All changes are backwards-compatible.
    - Any exceptions would be widely announced before reaching.
  - All published examples are autotested
- Contact us via **support (at) workbooks.com**
  - Please include your code, the intention of the script, and as much information about the problem.
  - Make sure you’ve read your logs carefully first.
  - We are happy to write scripts for our customers if you purchase Admin Credits from us: contact **sales (at) workbooks.com**