Introduction

XINA provides an integrated system for running and tracking asynchronous tasks.

There are currently two task implementation options. They may either be a generic process registered with the XINA Run utility, or implemented in the Amazon Web Services (AWS) Lambda service. Task deployment must be explicitly configured for each XINA instance.

Tasks are executed with the `RUN` API action. Note that successful completion of a `RUN` action does not mean the task has executed successfully, just that the task has been queued for execution successfully. Depending on the task type and configuration it may take additional time to start, or may never start if something is misconfigured or disabled.

The `RUN` action used the following syntax:

```
{
 "action" : "run",
 "tasks"  : [ <task JSON object>, ... ]
}
```

Each task is defined with the following properties:

Property	Type	Value
`name`	`string`	task name
`conf`	`JSON object`	task configuration
`auto`	`boolean`	indicates if a task was started automatically (optional, default `false`)
`archive`	`boolean`	if `true` files will be saved locally by XINA Run (optional, default `false`)
`open`	`boolean`	if `true` not be concluded after completing (lambda only) (optional, default `false`)
`parent`	`integer`	parent task ID (lambda only) (optional)
`desc`	`string`	plain text task description (optional)
`thread`	`string`	task thread name (XINA Run only) (optional)
`ref_id`	`integer`	arbitrary reference ID for client use (optional)
`priority`	`integer`	task priority (not yet implemented)
`timeout`	`integer`	absolute task timeout (not yet implemented)

The task configuration object is the primary means by which information can be passed to the task. The format required for the object is defined by each task. Note that XINA does not validate the configuration format when receiving the as it does not know what a valid format would be. It only ensures that the configuration is a valid JSON object.

Revision #2
Created 9 June 2022 19:27:25 by Nick Dobson
Updated 9 June 2022 19:29:56 by Nick Dobson