

Admin REST API

Excluded Features

I intentionally excluded the following management portal features:

- The entire Analytics, Interoperability, and System Explorer sections
- Everything regarding shadow servers
 - Mirroring should be used instead
- System Operation System Logs
 - Available through OTel tracing
- System Administration Configuration Zen Reports
 - No longer going to be relevant
- System Administration Configuration [Additional Settings](#)
 - Can/should all be configured through CPF
- System Administration Configuration Web Gateway Management
 - Redirects to a separate page
- System Administration Configuration Device Settings [Magnetic Tape Devices](#)
 - Old and should not be used anymore
- System Administration Security Public Key Infrastructure
 - Deprecated
- Creating and viewing globals of a specific local database
 - Not really a configuration issue
- An endpoint to activate a new license key
 - There are other preferable ways to do this
- Deleting all locks held by a process
 - Possibly a security issue, going to hold off on this for now
- Anything that would require restarting IRIS to take full effect
 - For instance, many of the settings (like global buffers size) in System Administration Configuration System Configuration Memory and Startup
- Security Advisor
 - Going to either skip this or reimagine it somehow
- Exporting audit records to a file, importing a list of tasks from a file, importing a list of resources from a file, exporting a list of resources to a file.
 - These should be doable through other apis
- Everything related to backups (configuring backup tasks, running backup tasks, viewing a history of backups, viewing/editing the list of databases to be backed up)
 - The current SMP features focus on the old online backup model, and we are working on a new approach

Async Endpoints

Several of the actions available in the management portal could take a long time to run. I've provided a list below. For most of them, we will use an async work queue. The initial request will return a 202 status code and a header with the id of the task in the queue. There are also endpoints at /v1/work to check on the status of one such action, or cancel an action. This work queue would be similar to the concept of "Background Tasks" in the current management portal. I've marked each endpoint below with whether or not I think it should be implemented with the async queue. The main dilemma is whether or not to take the async approach with actions that involve network requests but aren't likely to be CPU-intensive, such as joining a mirror or testing an SSL configuration.

1. Running an integrity check on a journal file (async)
2. Truncating a local database (async)
3. Compacting a local database (async)
4. Defragmenting a database (async)
5. Running an integrity check on a database (async)
6. Mounting a database (not async)
7. Dismounting a database (not async?)
8. Expanding the size of a database / expanding into a new volume (async)
9. Catching up a mirrored database (async)
10. Copying audit records to another namespace (async)
11. Purging audit records (async)
12. Listing audit records (async)
13. Testing an SSL configuration (async?)
14. Testing an LDAP configuration (async?)
15. Joining a mirror (async?)
16. ~~Running a backup task (async)~~
17. Listing journal records (async)
18. Enabling interop in a namespace or copying mappings to it (async)

API Security

Each of these APIs are associated with an IRIS resource that the caller must hold. I've specified these in the OpenAPI spec with the "x-resource" field. I've also added them to the "summary" field so that they show up in the Confluence OpenAPI viewer

Open Questions

- Should we rename the following sets of endpoints?
 - `/v1/external-language-server`
 - Should these be spelled with "els" instead of "external-language-server"?
 - `/v1/iml`
 - Should we get rid of these altogether?
 - `/v1/web`
 - `/v1/device`
 - `/v1/work`
- How should we reimagine the `/v1/security/service` and `/v1/security/system` endpoints? And what about the security advisor?
- Should cancelling an async task (i.e. `DELETE /work/id/{id}`) kill the process running the task? You probably wouldn't want this to happen for tasks like defragmenting a database
- What CPF fields should go in the `/v1/config` endpoints?
- Do we need the `/security/phone` endpoints?
- Are the `/security/privileged-routine` endpoints safe?

Other Notes

- All response bodies have the following fields
 - "status": An object representing errors which occurred
 - "console" : An array of strings captured from the write command - usually is empty
 - "result": An object containing info specific to the endpoint.
- Dates are represented in YYYY-MM-DD format. Times are in HH:MM:SS format
- Several of us had previously agreed to combine the GET, PUT, and DELETE `/v1/database/local/id` and `/v1/database/remote/id` endpoints into just `/v1/database/id` endpoints. However, I found it cleaner to split them back into separate endpoints for local and remote databases, as the request/response schemas were quite different in the two cases and I didn't want a bunch of fields which were only relevant for one case or the other.
- I've made an effort to keep PUT request body schemas identical to the response body schemas from GET and PUT on the same route. In a few cases, I've created extra endpoints like `/v1/database/local/id/{name}/metrics` to fetch read-only info that does not belong in a PUT request body.
- A few terms like "Id", "Namespace", and "Uri" are spelled with different capitalization ("Id" vs "ID") in various places in the schemas. The names in the schemas are intended to intentionally match the name of underlying class properties.