



Sandpiper (/spaces/127/sandpiper) ▸ Discussions (...)

Private Space · Technology Standards (/spaces/9/technology-standards/technologyhome)

(/spaces/127/sandpiper)

Activity Stream (/spaces/127/sandpiper/?act=1)

People (/spaces/127/sandpiper/people)

Content ▼

Posted in: API (/spaces/127/sandpiper/forums/5044/api)

The Sync API

✉ Unsubscribe from this discussion

📡 Subscribe to RSS (../..../..../spaces-cf/forums/rss-space-posts.ashx?

spaceID=127&topicID=5939&forumID=5044&key=rrer%2B1xDo%2BlxpC7jBRbM5w%3D%3D)



Doug Winsby (<https://autocare.communifire.com/people/dougwinsby>)



4/21/2020

The server includes these sync end points:

```
POST /sync/{url}
GET /sync
```

The POST end point is only valid on a "secondary" server and is used to initiate the sync with a "primary" server (address provided in {url}). No body request is required. The standard headers are used for JWT authentication.

The secondary server then acts like a "client" and sends a GET /sync request to the designated server url (after logging in and receiving a JWT).

When the primary server receives the sync GET request, it "upgrades" the http connection to a websocket connection and responds with a begin sync message.

The secondary server then sends a JSON sync request for a slice_id, and the primary responds with content information (hash and count) for that slice. The sync then proceeds as explained [here](https://autocare.communifire.com/spaces/127/sandpiper/forums/api/5146/the-sync-process) (<https://autocare.communifire.com/spaces/127/sandpiper/forums/api/5146/the-sync-process>).

The GET command is only available on a "primary" server. It listens for sync requests.

You might be asking, how does the initial request from the secondary server know to kick off? It can certainly be manual (because it is a simple API endpoint), but that would be tedious. So we are including a "sync" command in the `sandpiper` CLI. This can be run from a scheduler on a periodic basis. It will have an option to "sync all" companies.

```
sync_endpoint (/spaces/127/sandpiper/searchresults?keyword=%22sync_endpoint%22&searchtags=1)
```

```
sync_initiation (/spaces/127/sandpiper/searchresults?keyword=%22sync_initiation%22&searchtags=1)
```

👍 1 Like

Reply (/forums/post?tid=5939&ReplyPostID=5940&SpaceID=127)



Doug Winsby (<https://autocare.communifire.com/people/dougwinsby>)



4/23/2020

The **sandpiper sync** command currently looks like this:

```
sandpiper sync \  
--sub "acme-rotors" \ # optional subscription name or company_id  
--noupdate           # dry-run without changing anything
```

This would allow the following example usage:

```
sandpiper sync # sync all our "active" subscriptions  
sandpiper sync --sub "acme-rotors" # sync by subscription name  
sandpiper sync --sub 2bea8308-1840-4802-ad38-72b53e31594c # sync by company_id
```

Adding the `--noupdate` flag performs the actual sync with the primary servers, but only reports status messages (without updating the local database). We might want a way to indicate a test sync so the "sync activity" statistics on the primary server report actual syncs.

```
sandpiper-sync-cli (/spaces/127/sandpiper/searchresults?keyword=%22sandpiper-sync-cli%22&searchtags=1)
```

```
test-sync (/spaces/127/sandpiper/searchresults?keyword=%22test-sync%22&searchtags=1)
```

👍 Like

Reply (/forums/post?tid=5939&ReplyPostID=5947&SpaceID=127)

Answer



Doug Winsby (<https://autocare.communifire.com/people/dougwinsby>)



4/24/2020

Thinking through how a company will setup for the initial sync, it seems natural that the Primary would add the subscriptions (slices assigned to companies) and then the Secondary would ask for all subscriptions assigned to them.

So this means that the Secondary (who currently initiates the sync) would open a sync session by first asking for their subscriptions. It would add any new ones to their local database.

There is also an "active" subscription flag that could be changed (on either side). If disabled on the Primary, it will update the Secondary and log the activity. If enabled on the Primary, it will not change the Secondary. The active flag on the Secondary controls if it tries to sync that subscription, but changes are not propagated to the Primary. So, all of this means that the Primary controls what can be synced, but the Secondary can turn the sync off.

The sync process will also observe the "active" company flag (on both sides) and the "allow_sync" slice updating flag (on the Primary).

new-subscriptions (/spaces/127/sandpiper/searchresults?keyword=%22new-subscriptions%22&searchtags=1)

sync-flags (/spaces/127/sandpiper/searchresults?keyword=%22sync-flags%22&searchtags=1)

sync-process (/spaces/127/sandpiper/searchresults?keyword=%22sync-process%22&searchtags=1)

👍 Like

Reply (/forums/post?tid=5939&ReplyPostID=5949&SpaceID=127)

Answer



Doug Winsby (<https://autocare.communifire.com/people/dougwinsby>)



4/30/2020

On 4/24/2020 4:51 PM, Doug Winsby said:

So this means that the Secondary (who currently initiates the sync) would open a sync session by first asking for their subscriptions. It would add any new ones to their local database.

It's interesting to see how combining the server-roles (primary and secondary) into the same codebase and database schema, subscriptions become a bit tricky.

The way we designed it, subscriptions are relative to the server they're on. So, for example, a primary server for ABC Brakes would have subscriptions (company_id, slice_id) for each of its trading partners (e.g. Epicor, Advance Auto, WHI). But a secondary server for Epicor would have its subscriptions with each of its trading partners (e.g. ABC Brakes, XYZ Wipers).

If you look at that carefully, you'll notice that the companies are swapped for the same subscription. That is to say, they are relative to the server owner. This means that we must substitute the company_id when we apply any new subscriptions from a primary to a secondary.

For example, when Epicor asks for new subscriptions from ABC Brakes, it must change the company_id from "Epicor" (as found on the primary) to "ABC Brakes" (as required on the secondary).

There's probably a design that would include both primary and secondary company_ids in the subscription, but it makes everything more complicated when performing certain operations (because you'd need to know the server_role to join on the correct company_id).

Just a fun little twist we ran into when working on the sync process.

server-role (/spaces/127/sandpiper/searchresults?keyword=%22server-role%22&searchtags=1)

 Like

Reply (/forums/post?tid=5939&ReplyPostID=5953&SpaceID=127)

Answer

Page 1 of 1 (4 items)

Copyright © 2021 Axero Solutions LLC.
Auto Care Association powered by Communifire™ Version 8.0.7789.8960

© 2021 - AUTO CARE ASSOCIATION (<http://autocare.org>) | LEGAL & PRIVACY STATEMENT
(<https://www.autocare.org/privacy-statement/>)