verifone

Installing an Application

Applications are installed by constructing an object with a `services` key which holds an array of objects with docker containers to be installed. The objects in v2 of the API contain three fields.

- `image` This is the full url to the repo holding your docker image and tag without the protocol (http, https etc)
- `username` Username to your docker repo
- `password` Password to your docker repo

In our API we have environmental variables to allow for dynamic variable references inside your applications. These can be any environmental variable you wish. Below is an example of an application install body.

```
{
"services": [
{
"image": "index.docker.io/oaklabs/app-lights:latest",
"environment": {
"LIGHTS_HOST": "localhost:9100",
"PLATFORM_HOST": "localhost:443",
"NODE_ENV": "production"
}
},
{
"image": "index.docker.io/oaklabs/component-oak-lights:0.0.1",
"environment": {
"PORT": "9100"
}
}
]
}
```

As you can see here the oak-lights container will run on port `9100` and the UI application has the LIGHTS_HOST environmental variable also set to port `9100` so that communication from the UI can send messages to the right container.

Getting Information About Your Application

Applications are installed in both a `Live` and `Idle` area of the OakOS Platform. New installs always go into the `Idle` area and can be swapped with the `Live` installed application. By default and through using the 'Factory Reset' call, the default application is installed in both the Live and Idle application staging areas. The following `GET` requests manipulate these staging areas.

View Live and Idle

These calls will return the docker-compose YAML file that is used while running this application.

GET Request

```
http://{{dashboardHost}}/api/{{dashboardVersion}}/machine/{{dashboardMachine}}/application/viewLive
```

GET Request

```
http://{{dashboardHost}}/api/{{dashboardVersion}}/machine/{{dashboardMachine}}/application/viewIdle
```

Stop and Start

Stops and Starts the `Live` application.

GET Request

```
http://{{dashboardHost}}/api/{{dashboardVersion}}/machine/{{dashboardMachine}}/application/stop
```

GET Request

```
http://{{dashboardHost}}/api/{{dashboardVersion}}/machine/{{dashboardMachine}}/application/start
```

Swap Idle and Live

Swapping the `Live` and `Idle` application allows you to test or stage applications and is handy to see a `what if` scenario.

GET Request

```
http://{{dashboardHost}}/api/{{dashboardVersion}}/machine/{{dashboardMachine}}/application/SwapIdleAndLive
```

View Status

Allows you to check the status of the `Live` application to see if it is running.

GET Request

```
http://{{dashboardHost}}/api/{{dashboardVersion}}/machine/{{dashboardMachine}}/application/status
```