

The Nitty-Gritty of the Docker API

How to be friends with the Docker API

Tom Duffield
@tomduffield

Disclaimer

- The following is based on v1.16 of the API.

Origins of the Talk

I was working on a collection of Docker-related Chef projects that all used the CLI to communicate with the API. To avoid shelling out from Ruby and get more control I wanted to talk to the Docker API itself.

- <https://github.com/bflad/chef-docker>
- <https://github.com/chef/chef-provisioning-docker>
- <https://github.com/portertech/kitchen-docker>

Why am I giving this talk?

- The process of moving these projects from the CLI to the API was much more difficult than I anticipated.
- The Docker documentation doesn't cover how to transition from using the CLI to using the API.
- If your backend is talking to the API, you need to decide which way you want your users to speak with you: CLI format or API format.
- The result was a PR to swipely/docker-api^[1].

[1] <https://github.com/swipely/docker-api/pull/231>

TL;DL

I wrote a Ruby class^[1] that will convert CLI input into something you can pass directly to the API. It is in Ruby but the code is pretty well documented and all in one place.

[1] <https://github.com/swipely/docker-api/blob/lib/docker/container/config.rb>

Why would I use the API?

- Talk to Docker without installing Docker.
 - No Access or Permission
 - Shared hosts, highly restrictive environments
 - No Native Docker Support
 - iOS, Android, Windows, OS X
 - More “secure”
 - No shelling out means no shell vulnerabilities.

Tips for getting started with the API

- Keep on eye on the remote documentation^[1]
- Use existing libraries^[2]

[1] https://docs.docker.com/reference/api/docker_remote_api/

[2] https://docs.docker.com/reference/api/remote_api_client_libraries/

The Gotchas

- The CLI does a lot of validation and transformation^[1] on the data before it passes it to the API.
- If something isn't documented, the only real option is to look at the code. If you don't know how to read Go code, then you'll need to learn.

[1] <https://github.com/docker/docker/blob/master/api/client/commands.go>

Gotcha #1: No auto naming

Those funky names? They come from the CLI.
You need to set your own name when
submitting via the API.

```
REMOTE_API/containers/create?name=NAME
```

Gotchas #2 & 3: Cmd & Entrypoint

The API “needs” your Cmd and Entrypoint values to be shellword^[1] arrays.

```
'Cmd' : [ 'my_app',  
          '-p', 'param1',  
          '--long-param', 'param2' ]
```

[1] <http://ruby-doc.org/stdlib-2.0/libdoc/shellwords/rdoc/Shellwords.html>

Gotcha #4: Device Mapping

In the API, devices require a special mapping.

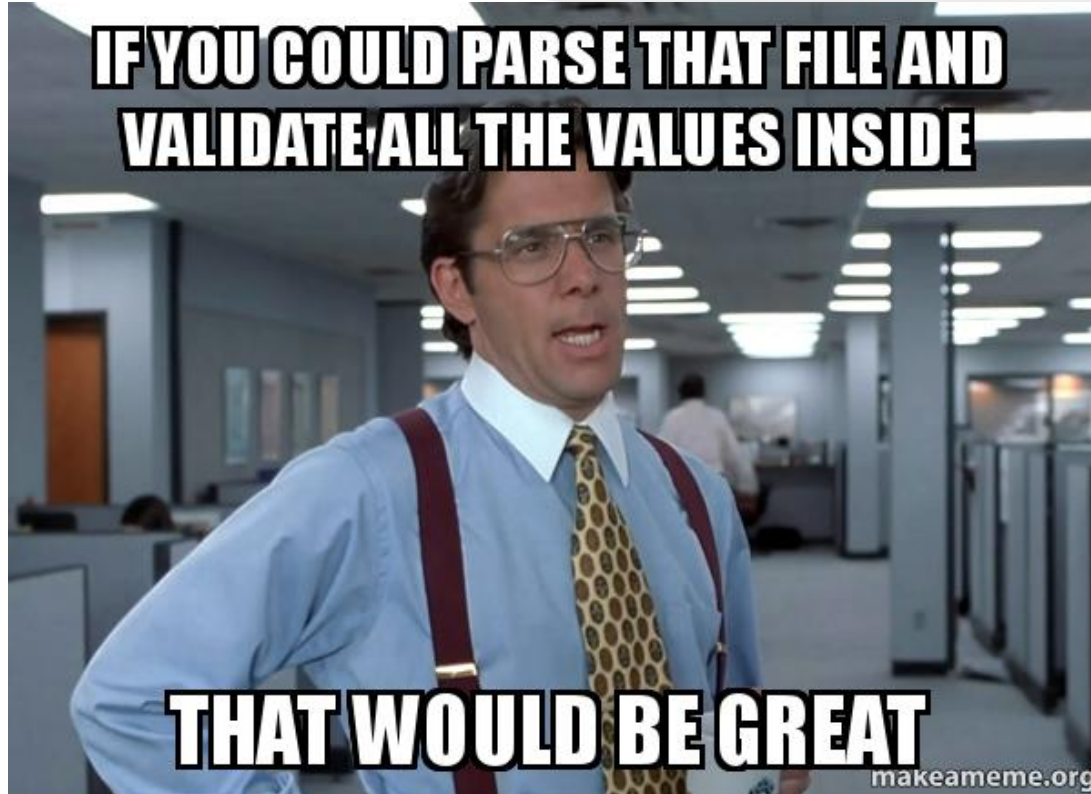
CLI:

```
--device=hostPath:containerPath:permissions
```

API:

```
'HostConfig': {'Devices': [{  
  'PathOnHost': hostPath,  
  'PathInContainer': containerPath,  
  'CgroupPermissions': permissions  
}]}
```

Gotcha #5: Environment Files



Gotcha #6: Exposing Ports

No ranges. You need to specify each port yourself. Hope you like loops! Also, protocols!

CLI:

```
--expose 8000-9000
```

API:

```
'ExposedPorts': {  
  'tcp/8000': {},  
  'tcp/8001': {},  
  ...
```

Gotcha #7: Memory

In the CLI you can specify memory size in KB, MB or GB. However, the API only accepts Bytes.

CLI:

```
--memory 2g
```

API:

```
'Memory' : 2000
```

Gotcha #8: Publish Ports

A single CLI input turns into two API values.

CLI:

```
--publish ip:hostPort:containerPort/proto
```

API:

```
'ExposedPorts': {'containerPort/proto': {}},  
'HostConfig': {'PortBindings': {'containerPort/proto': {  
  'HostPort': hostPort,  
  'HostIP': ip  
}}}
```

Gotchas #9: Volumes

A single CLI input turns into two API values.

CLI

```
--volume /host:/container
```

API

- 'Volumes'
- 'HostConfig': {'Binds'}

Volumes Example 1 (no host)

CLI:

```
--volume /container
```

API:

```
'Volumes': {'/container': {}}
```

Volumes Example 2 (with host)

CLI:

```
--volume /host:/container
```

API:

```
'Volumes': {'/container': {}},  
'HostConfig': {'Binds': ['/host:/container']}
```

Questions?

Tom Duffield
@tomduffield