Exploring The Syntax Schema

In this lesson we will be exploring the syntax schema using jx CLI.

We'll cover the following

- The jx step syntax schema command
- Validating buildpacks

I bet you were wondering how I knew that there are instructions like `overrides` and `loop`? I could have consulted the documentation in jenkins-x.io, but it is not always up-to-date. I could have consulted the code. As a matter of fact, most of what I know about pipelines comes from reviewing the code of the project. I did that not only because I like reading code, but also because that was the only reliable way to find out all the instructions we can specify.

The `jx step syntax schema` command

Fortunately, things have gotten a bit simpler since then, and the community has added the means to consult the full syntax schema with a single command.

```
jx step syntax schema
```

The output is too long to be presented in a book, so I'll choose a single construct (`loop`) and show only the output related to it.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "$ref": "#/definitions/ProjectConfig",
  "definitions": {
    ...
    "Loop": {
      "properties": {
        "steps": {
          "items": {
            "$ref": "#/definitions/Step"
          },
          "type": "array"
        },
        "values": {
          "items": {
            "type": "string"
            "type": "string"
          },
          "type": "array"

        },
        "variable": {
          "type": "string"
        }
      },
      "additionalProperties": false,
      "type": "object"
    },
    ...
    "Step": {
      "properties": {
        ...
        "loop": {
          "$schema": "http://json-schema.org/draft-04/schema#",
          "$ref": "#/definitions/Loop"
        },
        ...
```

We can see that `Loop` is one of the definitions and that it can contain:

- `steps`
- `values`
- `variables`

The array of steps includes a reference to the `Step` definition. If we scroll down to that definition, we can see that, among others, it can have a loop inside. A step can contain a loop, and a loop can include steps inside it.

We won’t go into all the definitions we can use inside pipelines. That would be too much and could easily fill an entire book alone. We will probably explore a couple of other schema definitions, and it’s up to you to go through those we’ll skip. For now, I’ll use this opportunity to introduce you to a few other potentially useful commands.

We can add the `--buildpack` argument if we want to find out the schema that is used with build pack pipelines.

```
jx step syntax schema --buildpack
```

Application pipeline and build pack schemas are very similar, but there are still a few differences. Use one schema or the other depending on whether you’re planning to modify the pipeline of your application or of a build pack.

Validating buildpacks

Since we are talking about build packs, it’s worth noting that we can validate those already in use by our cluster with the command that follows.

```
jx step syntax validate buildpacks
```

If all the build packs are valid, we’ll see the `SUCCESS` status in the output of each.

Let’s wrap up the discussion related to extending Jenkins X pipelines in the next lesson and see how we can remove the used resources.