Google

Lesson 4

Live programmatic validation

Before you get started

This onboarding deck has interactive features and activities that enable a self-guided learning experience. To help you get started, here are two tips for viewing and navigating through the deck.

1 View this deck in presentation mode.

- To enter presentation mode, you can either:
  - Click the **Present** or **Slideshow** button in the top-right corner of this page.
  - Press **Ctrl+F5** (Windows), **Cmd+Enter** (macOS), or **Ctrl+Search+5** (Chrome OS) on your keyboard.
- To exit presentation mode, press the **Esc** key on your keyboard.

2 Navigate by clicking the buttons and links.

- Click the **Back** or **Next** buttons to go backward or forward in the deck. Moving forward, you'll find them in the bottom corners of every slide.
- Click blue text to go to another slide in this deck or open a new page in your browser.
- For the best learning experience, using your keyboard or mouse wheel to navigate is discouraged.

Ready to get started? Let's go!

Lesson 4

Live programmatic validation

What you'll learn about:

- Tools that validate UDMI data at small and large scales

By the end of this lesson, you'll be able to:

- Run the validator with a single device
- Check the captured messages and error files
- Run the validator with multiple devices and interpret the results

Back

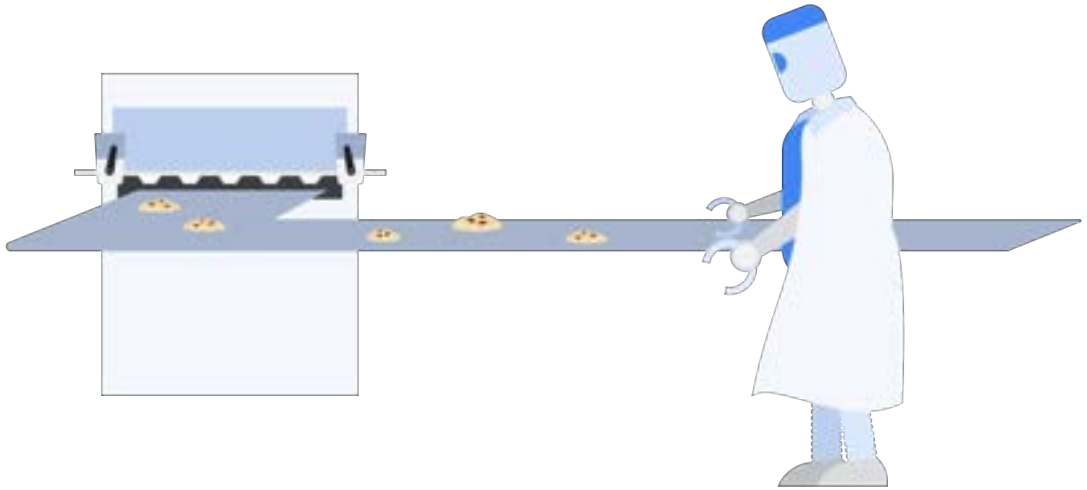
Next

Google

What is live programmatic validation?

To understand what live programmatic validation is, it helps to think about it with a slightly less technical example.

Imagine you work at a cookie factory. In that factory is a machine dedicated to making chocolate chip cookies. Periodically, a cookie inspection robot observes as the cookies come out of the machine. The robot checks that each cookie matches the factory's definition for a good chocolate chip cookie.



Back

Next

What is live programmatic validation? (continued)

The robot scans the cookies as they come off the line.

- Are they burnt?
- Do they have peppermints instead of chocolate chips?
- Has the machine started producing cupcakes instead of cookies for some reason?

If the output of the machine needs to be chocolate chip cookies, these would all be potential issues.

Back

Next

Google

What is live programmatic validation? (continued)

In this example, the cookie inspector robot is doing a form of live programmatic validation.

- It's live because the inspector is checking the machine's output as it's working.
- It's programmatic because no human needs to check the cookies manually.
- It's validation because the inspector robot is confirming if the output of the cookie machine is indeed chocolate chip cookies.

Back

Next

The validator tool

Validation for UDMI is handled by a special tool included in UDMI — the validator.

While it's running, the validator receives all messages sent to a particular Pub/Sub topic and then validates them. This ensures that it can be used to validate one, or multiple devices.

Keep in mind that the validation performed by the validator is a "read only" observation of individual messages. It doesn't alter the data contained within those messages.

Back

Next

How the validator works

The validator works by observing messages (telemetry, state) being sent from devices to the cloud.

Since the validator checks real messages as sent to the cloud, it's considered live validation.

The validator checks:

- If devices are communicating with the cloud
- That messages being sent from devices conform with the UDMI schema
- That messages include all required points, as defined by the site model

Since the validator checks messages without manual intervention, it's considered programmatic.

Validation criteria

The validator compares each message against the UDMI message schema definition to confirm whether or not it's structured correctly.

If it's not, an error is generated.

To learn more about the UDMI schema, check out the schema viewer.

The validator checks devices it receives messages from against the set of devices listed in the site model, and highlights if there are any unexpected or missing devices.

The validator also compares messages to the metadata in the site model to ensure that they include all required points. If any points are missing, or if the message contains additional points not outlined in the metadata, an error is generated.

Back

Next

Lesson 4

Knowledge check

Let’s take a moment to reflect on what you’ve learned so far.

- The next slide will have a question related to the concepts and actions covered in this lesson.
- Review the question and select the correct response.

You won’t be able to move forward until the correct answer is selected.

Back

*Click **Next** when you’re ready to begin.*

Next

Google

Knowledge check 1

Which of these actions is NOT performed by the validator when it examines a message?

Select the best answer from the options listed below.

- Checks if the device is communicating with the cloud
- Compares the message against the UDMI schema
- Confirms that the message contains all required points
- Adds any missing points to the message

Back

Next

Google

Knowledge check 1

Which of these actions is NOT performed by the validator when it examines a message?

Select the best answer from the options listed below.

- Checks if the device is communicating with the cloud
- Compares the message against the UDMI schema
- Confirms that the message contains all required points
- Adds any missing points to the message

Close... but not quite right! 🤔

Actually, that is an action the validator performs while examining a message. Remember, the validator makes “read-only” observations of messages. Which answer choice doesn’t seem to align with that fact?

Try again

Back

Next

Google

Knowledge check 1

Which of these actions is NOT performed by the validator when it examines a message?

Select the best answer from the options listed below.

- Checks if the device is communicating with the cloud
- Compares the message against the UDMI schema
- Confirms that the message contains all required points
- Adds any missing points to the message

Close... but not quite right! 🤔

Actually, that is an action the validator performs while examining a message. Remember, the validator makes “read-only” observations of messages. Which answer choice doesn’t seem to align with that fact?

Try again

Back

Next

Google

Knowledge check 1

Which of these actions is NOT performed by the validator when it examines a message?

Select the best answer from the options listed below.

- Checks if the device is communicating with the cloud
- Compares the message against the UDMI schema
- Confirms that the message contains all required points
- Adds any missing points to the message

Close... but not quite right! 🤔

Actually, that is an action the validator performs while examining a message. Remember, the validator makes “read-only” observations of messages. Which answer choice doesn’t seem to align with that fact?

Try again

Back

Next

Google

Knowledge check 1

Which of these actions is NOT performed by the validator when it examines a message?

Select the best answer from the options listed below.

Checks if the device is communicating with the cloud

Compares the message against the UDMI schema

Confirms that the message contains all required points

Adds any missing points to the message

That's right! 🎉

The validator makes "read-only" observations of messages. It doesn't add, delete, or alter any data within a message.

Back

Next

Google

Lesson 4

Practice

Let’s take a moment to apply what you’ve learned so far.

- The next slides will walk through the steps for running the validator for a single device.
- After this practice activity, there will be a brief knowledge check.
- Later, you’ll move on to validating multiple devices.

Back

*Click **Next** when you’re ready to begin.*

Next

Google

Running the validator

For this exercise, we'll be using pubber to simulate a device sending messages to the cloud.

1. The first thing you need to do is start up pubber.

Starting pubber

You can use the following command to start an instance of pubber. You'll just need to replace the project ID shown below (in green) with your own. In this instance, we're using the extra_field option to have pubber send non-UDMI-compliant messages so that we can see the full validation report with errors.

```
bin/pubber sites/udmi_site_model udmi-learning AHU-1 6236 extra_field
```

For more on pubber and how to use it, refer to Lesson 3: UDMI site model and device-to-cloud basics.

Back

Next

Running the validator (continued)

2 Next, we need to start up the validator tool.

Starting the validator

The template for running the validator is:

```
bin/validator SITE_PATH PROJECT_ID SUBSCRIPTION_ID
```

- SITE_PATH is the path to your site model directory
- PROJECT_ID is your Google Cloud Platform (GCP) project ID
- SUBSCRIPTION_ID should match the GCP subscription created

Here's an example of the command:

```
bin/validator sites/udmi_site_model udmi-learning udmi_target_subscription
```

Back

Next

Running the validator (continued)

The validator tool may take a few moments to load up.

It will immediately start to validate messages as they're received. A running output log will be printed to the terminal window, similar to the example to the right.

Validator will continue to run indefinitely until you end it by using **CTRL + C**.

Example

```
Processing device #1/4: AHU-1/event_pointset
Warning: Nashorn engine is planned to be removed from a future JDK release
Error validating schema: While converting to json node: 1 schema violations found
Error validating contents: Unrecognized field "extraField" (class udmi.schema.PointsetEvent), not marked as ignorable (4
 known properties: "version", "points", "timestamp", "partial_update"])
 at [Source: UNKNOWN; line: -1, column: -1] (through reference chain: udmi.schema.PointsetEvent["extraField"])
Writing errors to /Users/username/udmi_site_model/out/devices/AHU-1/event_pointset.out
Processing device #1/4: AHU-1/event_system
Processing device #1/4: AHU-1/state
```

Back

Note: When running the validator, allow enough time for all devices to be seen.
To do that, you'll need to consider the time at which they publish data.

Next

Validation results

Validation results are stored in the “out” directory of the site model.

If you’re following earlier instructions this is: **`sites/udmi_site_model/out`.** Within that directory, an overall validation report titled **`validation_summary.json`** will be generated.

The following validation results may appear:

- **`missing_devices`** – Devices which were expected, but weren’t seen by the validator (didn’t send any telemetry or states whilst it was running)
- **`extra_devices`** – Devices which were seen but weren’t in the site model
- **`correct_devices`** – Devices which had no validation errors
- **`error_devices`** – Devices which had validation errors
- **`devices`** – A collection of devices which were seen during validation. A device’s entry may comprise the following:
  - **`status`** – Present if the device had an error, the status block gives summary of the device’s errors
  - **`last_seen`** – Timestamp of the last message seen during validation

Lesson 4

Knowledge check

Let’s take a moment to reflect on what you’ve learned so far.

- In the upcoming slides, you’ll see examples of overall validation reports.
- Use the examples and what you’ve learned about validation results to answer the questions that follow.

For the multiple choice questions, you won’t be able to move forward until the correct answer is selected.

Back

*Click **Next** when you’re ready to begin.*

Next

Google

Knowledge check 2

In the following report, what is the error with the message from device AHU-1?

Select the best answer from the options listed below.

It's missing the point "filter_alarm_pressure_status".

There is a schema error within the message.

It's missing the point "temperature_set_point".

It contains an extra point.


```
{
 "timestamp" : "2022-08-03T11:43:54Z" ,
 "version" : "1.3.14",
 "errors" : [ ],
 "summary" : {
   "correct_devices " : [ ],
   "extra_devices " : [ ],
   "missing_devices " : [ "AHU-22", "GAT-123", "SNS-4" ],
   "error_devices" : [ "AHU-1" ]
 },
 "devices" : {
   "AHU-1" : {
     "last_seen": "2022-07-16T18:28:35Z" ,
     "status" : {
       "message" : "Metadata validation failed: Missing
points: filter_alarm_pressure_status" ,
       "category" : "validation.error.simple" ,
       "level" : 500
     }
   }
 }
}
```


Back

Next

Google

Knowledge check 2

In the following report, what is the error with the message from device AHU-1?

Select the best answer from the options listed below.

- It's missing the point "filter_alarm_pressure_status". ✓
- There is a schema error within the message.
- It's missing the point "temperature_set_point".
- It contains an extra point.

That's right! 🎉

According to the site model, messages from AHU-1 should contain the point "filter_alarm_pressure_status" but the validator is not seeing that point.

```
"message" : "Metadata validation failed:  Missing points:
filter_alarm_pressure_status ",
```

Back

Next

Knowledge check 2

In the following report, what is the error with the message from device AHU-1?

Select the best answer from the options listed below.

It's missing the point `"filter_alarm_pressure_status"`.

There is a schema error within the message.

It's missing the point `"temperature_set_point"`.

It contains an extra point.

Close... but not quite right! 🤔

Actually, the validator didn't find any schema issues with the message received from AHU-1. Take a closer look at the error the report lists for AHU-1 and try again.

```
    "AHU-1" : {
      "last_seen": "2022-07-16T18:28:35Z" ,
      "status" : {
        "message" : "Metadata validation failed: Missing
points: filter_alarm_pressure_status" ,
        "category" : "validation.error.simple" ,
        "level" : 500
      }
```

Try again

Back

Next

Knowledge check 2

In the following report, what is the error with the message from device AHU-1?

Select the best answer from the options listed below.

It's missing the point "filter_alarm_pressure_status".

There is a schema error within the message.

It's missing the point "temperature_set_point".

It contains an extra point.

Close... but not quite right! 🤔

While the message received from AHU-1 is missing a point, "temperature_set_point" is not it. Take a closer look at the error the report lists for AHU-1 and try again.


```
    "AHU-1" : {
      "last_seen": "2022-07-16T18:28:35Z" ,
      "status" : {
        "message" : "Metadata validation failed: Missing
points: filter_alarm_pressure_status" ,
        "category" : "validation.error.simple" ,
        "level" : 500
      }
```


Try again

Back

Next

Knowledge check 2

In the following report, what is the error with the message from device AHU-1?

Select the best answer from the options listed below.

- It's missing the point "filter_alarm_pressure_status".
- There is a schema error within the message.
- It's missing the point "temperature_set_point".
- It contains an extra point.

Close... but not quite right! 🤔

Actually, the message does not mention that extra points were received from AHU-1. Take another look at the report and try again.


```
    "AHU-1" : {
      "last_seen": "2022-07-16T18:28:35Z" ,
      "status" : {
        "message" : "Metadata validation failed: Missing
points: filter_alarm_pressure_status" ,
        "category" : "validation.error.simple" ,
        "level" : 500
      }
```


Try again

Back

Next

Knowledge check 3

Which of the following errors did the validator detect?

Select the best answer from the options listed below.

An extra device was seen.

A message from AHU-1 contained extra points.

A message from AHU-1 contained an unrecognized field.

A message from AHU-1 was missing a required field.


```
{
   "timestamp" : "2022-08-03T11:43:54Z" ,
   "version" : "1.3.14",
   "errors" : [ ],
   "summary" : {
     "correct_devices " : [ ],
     "extra_devices " : [ ],
     "missing_devices " : [ "AHU-22", "GAT-123", "SNS-4" ],
     "error_devices" : [ "AHU-1" ]
   },
   "devices" : {
     "AHU-1" : {
       "status" : {
       "last_seen": "2022-07-16T18:28:35Z" ,
         "message" : "Unrecognized field 'extraField' (class
udmi.schema.PointsetEvent), not marked as ignorable (4 known
properties: 'version', 'points', 'timestamp',
'partial_update']) at [Source: UNKNOWN; line: -1, column:
-1] (through reference chain:
udmi.schema.PointsetEvent['extraField'])" ,
         "category" : "validation.error.simple" ,
         "level" : 500
       }
     }
   }
 }
```


Back

Next

Google

Knowledge check 3

Which of the following errors did the validator detect?

Select the best answer from the options listed below.

- An extra device was seen.
- A message from AHU-1 contained extra points.
- A message from AHU-1 contained an unrecognized field.
- A message from AHU-1 was missing a required field.

Close... but not quite right! 🤔

Actually, there are no extra devices listed, so that's not the error that the validator detected. Take another look at the report and try again.

```
  "summary" : {
    "correct_devices " : [ ],
    "extra_devices" : [ ],
    "missing_devices" : [ "AHU-22", "GAT-123", "SNS-4" ],
    "error_devices" : [ "AHU-1" ]
  },
```

Try again

Back

Next

Google

Knowledge check 3

Which of the following errors did the validator detect?

Select the best answer from the options listed below.

An extra device was seen.

A message from AHU-1 contained extra points.

A message from AHU-1 contained an unrecognized field.

A message from AHU-1 was missing a required field.

Close... but not quite right! 🤔

Actually, the report shows that no extra points were received from AHU-1. Take another look at the report and try again.


```
      "AHU-1" : {
        "status" : {
        "last_seen": "2022-07-16T18:28:35Z" ,
          "message" : "Unrecognized field 'extraField' (class
udmi.schema.PointsetEvent), not marked as ignorable (4 known
properties: 'version', 'points', 'timestamp',
'partial_update']) at [Source: UNKNOWN; line: -1, column: -1]
(through reference chain:
udmi.schema.PointsetEvent['extraField'])" ,
          "category" : "validation.error.simple" ,
          "level" : 500
        }
      }
```


Try again

Back

Next

Knowledge check 3

Which of the following errors did the validator detect?

Select the best answer from the options listed below.

An extra device was seen.

A message from AHU-1 contained extra points.

A message from AHU-1 contained an unrecognized field.

A message from AHU-1 was missing a required field.

That's right! 🎉

The validator detected an instance in a message from AHU-1 where the message didn't conform to the UDMI schema due to an unrecognized field. Nice work!


```
      "AHU-1" : {
        "last_seen" : "2022-07-16T18:28:35Z" ,
        "status" : {
          "message" : "Unrecognized field 'extraField' (class
udmi.schema.PointsetEvent), not marked as ignorable (4 known
properties: 'version', 'points', 'timestamp',
'partial_update']) at [Source: UNKNOWN; line: -1, column: -1]
(through reference chain:
udmi.schema.PointsetEvent['extraField'])" ,
          "category" : "validation.error.simple" ,
          "level" : 500
        }
      }
```


Back

Next

Knowledge check 3

Which of the following errors did the validator detect?

Select the best answer from the options listed below.

An extra device was seen.

A message from AHU-1 contained extra points.

A message from AHU-1 contained an unrecognized field.

A message from AHU-1 was missing a required field.

Close... but not quite right! 🤔

Actually, the message from AHU-1 isn't missing a required field. Take a closer look at the error message and try again.


```
      "AHU-1" : {
        "last_seen": "2022-07-16T18:28:35Z" ,
        "status" : {
          "message" : "Unrecognized field 'extraField' (class
udmi.schema.PointsetEvent), not marked as ignorable (4 known
properties: 'version', 'points', 'timestamp',

'partial_update']) at [Source: UNKNOWN; line: -1, column: -1]
(through reference chain:
udmi.schema.PointsetEvent['extraField']) ",
          "category" : "validation.error.simple" ,
          "level" : 500
        }
      }
```


Try again

Back

Next

Knowledge check 4

Take a look at the following validation report, then list all the errors that the validator discovered.

If it helps, use a separate document to note your findings. When you're ready to check your answers, advance to the next slide.


```
{
   "timestamp" : "2022-07-16T18:28:00Z",
   "version" : "1.3.14",
   "errors" : [ ],
   "summary" : {
     "correct_devices" : [ "FCU-3" ],
     "extra_devices" : [ "FCU-901" ],
     "missing_devices" : [ "FCU-4" ],
     "error_devices" : [ "FCU-1" ]
   },
   "devices" : {
     "last_seen": "2022-07-16T18:27:19Z",
     "FCU-1" : {
       "status" : {
         "message" : "Metadata validation failed: Extra points:
temperature",
         "category" : "validation.error.simple",
         "level" : 500
       }
     },
     "FCU-2" : {
       "last_seen": "2022-07-16T18:28:35Z",
       "status" : {
         "message" : "Unrecognized field 'extraField', not marked as ignorable (4
known properties: 'version', 'points', 'timestamp', 'partial_update'])"
         "category" : "validation.error.simple",
         "level" : 500
       }
     },
     "FCU-3" : {
       "last_seen": "2022-07-16T18:27:11Z"
     },
     "FCU-5" : {
       "last_seen": "2022-07-16T18:27:19Z",
       "status" : {
         "message" : "Metadata validation failed: Missing points:
filter_alarm_pressure_status",
         "category" : "validation.error.simple",
         "level" : 500
       }
     },
     "FCU-6" : {
       "last_seen": "2022-07-16T18:28:01Z",
       "status" : {
         "message" : "While converting to json node: 1 schema violations found.
object has missing required properties (['hardware'])"
         "category" : "validation.error.simple",
         "level" : 500
       }
     }
   }
 }
```


Back

Next

Knowledge check 4

Check your answers.

Here's the list of the errors that the validator found. Does this list match the one you created?

1. Issue with FCU-1: The validator found an extra point—temperature.
2. Issue with FCU-2: The validator found a violation of the schema due to an unrecognized field.
3. Issue with FCU-3: No issues were found.
4. Issue with FCU-4: This device was not seen by the validator, indicating that it could be offline.
5. Issue with FCU-5: The message from this device was missing a required point.
6. Issue with FCU-6: The validator found a violation of the schema due to a missing required field.
7. Issue with FCU-901: The validator noted that this device is not listed in the site model.

Back

Next

Lesson 4

Validation results

The validator also places results for each device in their respective device's directory within the "out" directory, like so: `SITE_MODEL/out/DEVICE_ID`.

For each message type (e.g. state, event_pointset, event_system), the validator creates:

- A record of the message (payload) which was received and validated, named `TYPE.json`, eg. `event_pointset.json`.
- A report named `TYPE.out` e.g. `state.out` for any validation errors that were found.
  `While converting to json node: 1 schema violations found`

In the next few slides, we'll take a look at some examples of these types of files.

Back

Next

Google

Validation results (continued)

This is an example of a record of an event pointset message.

The file is titled **`event_pointset.json`**.

Example


```
{
 "extraField" : "extrafield",
 "points" : {
   "filter_alarm_pressure_status"  : {
     "present_value" : false
   },
   "filter_differential_pressure_setpoint"  : {
     "present_value" : 98
   },
   "filter_differential_pressure_sensor"  : {
     "present_value" : 8
   }
 },
 "timestamp" : "2022-05-10T10:22:55Z" ,
 "version" : "1.3.14"
}
```

Validation results (continued)

This is an example of a report detailing an error discovered by the validator.

The file is titled **`event_pointset.out`**, and it details a schema violation.

Example


```
{
 "timestamp" : "2022-08-03T11:59:12Z" ,
 "version" : "1.3.14",
 "sub_folder" : "pointset",
 "status" : {
   "message" : "Metadata validation failed: Extra points:
extra_point",
   "category" : "validation.error.simple" ,
   "detail": "1 schema violations found. object instance has
properties which are not allowed by the schema: ['extraField']
Unrecognized field 'extraField' (class
udmi.schema.PointsetEvent), not marked as ignorable (4 known
properties: 'version', 'points', 'timestamp',
'partial_update']) at [Source: UNKNOWN; line: -1, column: -1]
(through reference chain:
udmi.schema.PointsetEvent['extraField'])'" ,
   "level" : 500
 },
 "pointset" : {
 }
}
```


Back

Next

Lesson 4

Practice

Running validator with multiple devices

Earlier, we took a look at how the validator works with one device. Now, let's turn our attention towards how to use it with multiple devices.

Back

*Click **Next** when you're ready to begin.*

Next

Google

Running validator with multiple devices

1. Keeping the existing pubber running, open a new terminal window and type the following to launch a new instance of pubber. In this example, we're using the device "FCU-1" which was created in a previous tutorial, but you can use any device in your site model.

   ```
   bin/pubber sites/udmi_site_model PROJECT_ID FCU-1 A1B2
   ```

   Note: A device can only have one active connection to IoT Core; therefore, you need to use a different device ID to simulate another device.

2. Start the validator again. For this example, use the following command:

   ```
   bin/validator sites/udmi_site_model PROJECT_ID udmi_target_subscription.
   ```

3. Wait for the messages to complete validating.

4. Observe the output. How does it differ from the output generated when you ran the validator earlier?

Back

Next

Lesson 4 summary

Let's review what you learned about:

- Tools that validate UDMI data at small and large scales

Now you should be able to:

- Run the validator with a single device.
- Check the captured messages and error files.
- Run the validator with multiple devices and interpret the results.

Back

Next

Google

You completed Lesson 4!

Now's a great time to take a quick break before starting Lesson 5.

Ready for Lesson 5?

Let's go!

Back

*Press the **Esc** key on your keyboard to exit presentation mode.*

Google

Helpful resources

Bookmark these resources for future reference.

- UDMI Project GitHub
  Contains specification for management and operation of IoT systems.
- Git Documentation
  Contains various sources of information about Git contributed by the Git community.