#JOINdata
@lookerdata

Hands-on Lab: Amplify, Looker, and the Central Source of Truth

Samuel Crane
Amplify

Leigha Jarett
Looker

#JOINdata
@lookerdata

looker.com/hol

Select the **Amplify, Looker, and the Central Source of Truth** lab in the drop-down

© 2019 Looker. All rights reserved. Confidential.

Samuel Crane, PhD

Director of Data Science, Amplify

@samuelcrane

Leigha Jarett

Solutions Engineer, Looker

© 2019 Looker. All rights reserved. Confidential.

looker

Agenda

Why create a data dictionary?

The Looker API

Creating a `field_metadata` table

Making a data dictionary dashboard

Taking things to the next level

© 2019 Looker. All rights reserved. Confidential.

Why create a data dictionary?

© 2019 Looker. All rights reserved. Confidential.

“Using the Looker API, we've been able to build data documentation that answers the hard questions we get from both our business and technical users.”

© 2019 Looker. All rights reserved. Confidential.

looker

Welcome to the Looker API

© 2019 Looker. All rights reserved. Confidential.

What is the Looker API?

Looker API is a secure, “RESTful” application programming interface for managing your Looker instance and fetching data through the Looker data platform.

With the Looker API we can:

- Manage users
- Create and run content
- Pull data for data science workflows
- Automate schedules
- Pre-cache reports
- And much more!

© 2019 Looker. All rights reserved. Confidential.

How do we make API calls to Looker?

RESTful API calls are made through HTTP requests, which can be sent in a variety of different ways.

- **HTTP Request** is a packet of Information that one computer sends to another computer to communicate something.

- Today, we are going to send requests using a Python script in an iPython notebook environment in Google Drive.

https://colab.research.google.com/drive/1asQncVaZoBM7Ic3zh2-s9AYhJ1UFU1Yo

looker

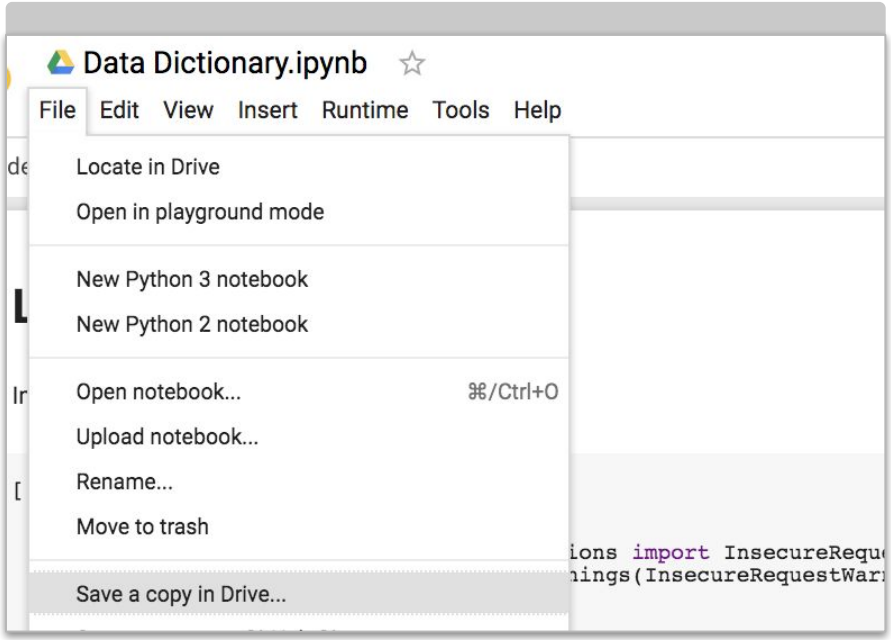
Creating a `field_metadata` Table

This table will contain the metadata on each field (dimensions, measures) in the model and will be written back to the database used within Looker.

© 2019 Looker. All rights reserved. Confidential.

Getting started in the notebook

You should have view-only access to this file, so the first thing you'll need to do is make a copy and save it to your drive so you can run and edit it.



Now you can go ahead and make changes to the file and run Python code directly from Google Drive!

© 2019 Looker. All rights reserved. Confidential.

Making the API calls from Python

In Python, we'll send API calls to our Looker instance. First, let's run the calls that import the packages we need from Python and then create some functions to make sending API calls easier.

Data Dictionary.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Looker Data Dictionary

Import packages

```
import pandas as pd
import requests

from requests.packages.urllib3.exceptions import InsecureRequestWarning
requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
```

Define some functions to help us make API calls to Looker

```
[2] class LookerApi(object):

        def __init__(self, token, secret, host):

            self.token = token
            self.secret = secret
            self.host = host
            self.session = requests.Session()
            self.session.verify = False
            self.auth()

        def auth(self):
            url = '{}{}'.format(self.host,'login')
            params = {'client_id':self.token,
                      'client_secret':self.secret}
            r = self.session.post(url,params=params)
            access_token = r.json().get('access_token')
            self.session.headers.update({'Authorization': 'token {}'.format(access_token)})
```

So instead of making HTTP requests each time we send an API call, we can use our Python functions.

looker

© 2019 Looker. All rights reserved. Confidential.

Creating the data table

Next, we'll connect to Looker and tell Python our model name. The script will grab all the information about every dimension and measure within each Explore in that model.

Connect to your Looker Instance

```
my_host = 'https://myname.looker.com:19999/api/3.1/'
my_secret = 'my_secret'
my_token = 'my_client_id'
```

```
[4]  looker = LookerApi(host=my_host,
         token=my_token,
         secret = my_secret)
```

We have already added the API keys for the JOIN Looker user — but feel free to update with your own after the lab!

Enter the model name you want to create the data dictionary on

```
[5]  model_name = 'thelook_leigha'
     model = looker.get_model(model_name)
```

https://saleseng.dev.looker.com:19999/api/3.1/lookml_models/thelook_leigha

Iterate through all the explores for that model and create a dataframe with the metadata for each field

```
[6]  field_dict = []
     for explore_n in model['explores']:
         explore = looker.get_explore(explore_name=explore_n["name"],model_name=model_name)
         dimensions = explore['fields']['dimensions']
         for d in dimensions:
             d['explore'] = explore['label']
             d['explore_name'] = explore['name']
             d['model_name'] = model_name
             d['explore_description'] = explore['description']
         measures = explore['fields']['measures']
         for m in measures:
             m['explore'] = explore['label']
             m['explore_name'] = explore['name']
             m['model_name'] = model_name
             m['explore_description'] = explore['description']
         field_dict = field_dict+dimensions+measures
```

© 2019 Looker. All rights reserved. Confidential.

Writing the data back to the database

Next, we need to take this data frame and write it back to our database used in Looker. Here is example code you can modify with your own scratch schema and table name when you're trying this out on your own. We're not going to run it today because we have already done it.

Write metadata results back to database

```
[213] def change_data_type(dtype_name):
        if dtype_name == 'object':
          return 'varchar'
        elif 'int' in dtype_name:
          return 'int'
        else:
          return dtype_name
```

```
[214] cols_and_dtypes = field_df.dtypes.apply(lambda x: change_data_type(x.name)).to_dict()
```

```
[220] create_table_query = 'create table hackathon3_scratch.join_field_metadata ( '
      for col_name in cols_and_dtypes.keys():
        create_columns_string = '%s %s,' % (col_name,cols_and_dtypes[col_name])
        create_table_query += create_columns_string +' '
      create_table_query = create_table_query[0:-2]+');'
```

```
[221] create_table_q = looker.create_sql_runner_query('redshift_ecommerce',create_table_query)
      looker.run_sql_runner_query(create_table_q['slug'])
```

```
<Response [200]>
```

```
[222] for i in range(len(field_df)):
        query = 'insert into category_stage values ('
        row = field_df.iloc[i,:]
        insert_values = ''
        for col_name in cols_and_dtypes.keys():
          val = str(row[col_name])
          if cols_and_dtypes[col_name] ==  'varchar':
            val ='"' + val + '"'
          if val == 'None' or val == '"None"':
            val = 'NULL'
          insert_values += val + ','
        query += insert_values +'), '
        query = query[0:-2]+';'
        query_r = looker.create_sql_runner_query('redshift_ecommerce',query)
        r = looker.run_sql_runner_query(query_r['slug'])
        if r.status_code == 200:
          print('Added row')
        time.sleep(2)
```

© 2019 Looker. All rights reserved. Confidential.

Making the data dictionary dashboard

Here we will model out the table we just added to the database so that we can create a dashboard within Looker that acts as a data dictionary.

looker

Creating the Explore

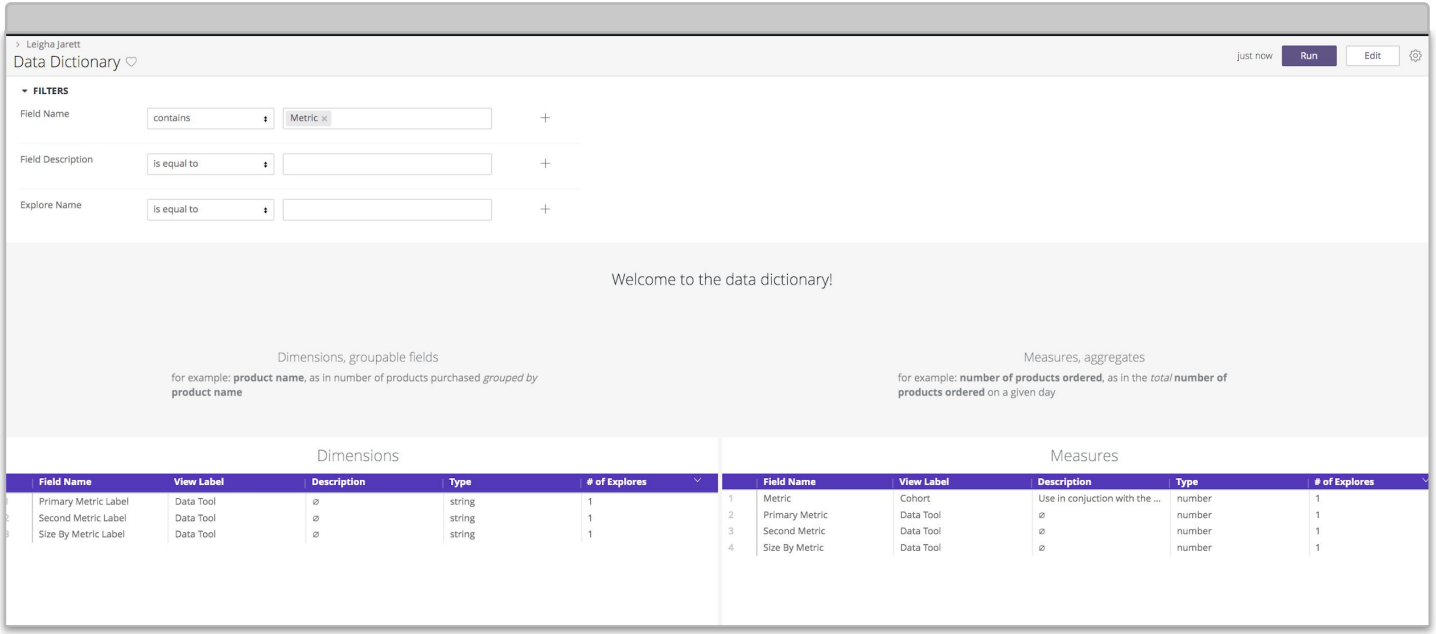
Now that the data exists as a table in our database, we have to create a view and an Explore — just like we would in Looker for any other dataset.

1. Go ahead and log in to Looker, then navigate to the **ecommerce** project.
2. Choose **Create view from table** and select the **join_field_metadata** table from the **hackathon3_**scratch schema to create a view file from that table.
3. Now create an Explore with just the **field_metadata** view by adding an `explore` object to the model file you created.

© 2019 Looker. All rights reserved. Confidential.

Creating the dashboard

With a model or Explore in place, we can create a dashboard that makes data definitions easily accessible and operational for our end users.



© 2019 Looker. All rights reserved. Confidential.

Customize with data actions & links

Consider how your end users are going to be using the data dictionary. You can add links to view a field in the Explore, or an action to email you with questions.

LOOKER DATA DICTIONARY Looker Data Dictionary Category **is "dimension"** Looker Data Dictionary Description **is null** Looker Data Dictionary Field Name **is "ID"** Looker Data Dictionary Hidden (Yes / No) **is No**

	Explore	Explore Description
1	(1) Orders, Items and Users	Use this explore to see all the data surrounding orders that have been placed, including information about the order itself, the product, and the User
2	(2) Web Event Data	Use this explore to see the data about web event sessions, including the users and the products viewed
3	(3) Web Session Data	Use this explore to see web session data, including information of the landing page
4	(5) Share of Wallet Analysis	Use this explore to do analysis on share of wallet
5	(6) Customer Journey Mapping	Use this explore to undertstand the customer journey, including repeat purchase information
6	Cohort Data Tool	∅
7	Web Analytics Data Tool	∅

LINKS

Go to explore

© 2019 Looker. All rights reserved. Confidential.

Taking things to the next level

An inside look into how Amplify has created a website to serve as business users' one-stop shop for all things related to data.

looker

Beyond the data dictionary

- **Looker:** Write good descriptions in the LookML.
- **Metadata:** Write down business logic not captured in Looker.
- **Data Hub:** Combine all this information in a central place.
- Use the Looker API to keep the documentation up to date.

© 2019 Looker. All rights reserved. Confidential.

Amplify Data Hub :: Amplify Da

Amplify.

Need help? Find us in the Slack channel: #support-looker

Amplify Data Hub

DATAHUB

This website describes the data that is available for analysis and reporting, covering all products and most data sources. The data described here is available through our business intelligence tool, Looker, or directly from the data warehouse. There are four sections to the site. Chapter 1 briefly reviews Amplify's data policies, and explains the access policy for the data warehouse. Chapter 2 describes the apps, services, and other sources of data. Chapter 3 defines all of the terms and metrics used across Amplify. Chapter 4 highlights common data analysis use cases for different teams.

The site is maintained by the Data Science and Data Engineering teams. All team members are encouraged to contribute. In making this service available, we had three goals in mind:

The Data Hub should be…

1. **Comprehensive.** The site should cover all terms used across the business.
2. **Helpful.** The site should be clear, informative, and accurate.
3. **Relevant.** The site should be kept up-to-date, growing and changing with the business.

Please have a look around and if you have any questions or comments, find us in Slack at #support-looker.

Amplify.

Need help? Find us in the Slack channel: #support-looker

Explore: Student Work (Curriculum)

Why use this Explore?

Use this Explore to analyze student work in the Curriculum App for Science and ELA. This Explore is based on the submissions to the Curriculum App, and joins in additional information about the content of ELA and Science, the users, and licenses.

Examples

- How many activities were handed in last week for each product?
  This query use a count of submissions, a date filter, and pivots on product.
- Which Units have the most submissions?
  This query uses a date range filter and groups the count of submissions by Unit, across ELA and Science. How would you add a count of users?

Caveats

- Including dimensions from the 'Class' view may lead to duplication of student-level submission data.
- PII is hidden by default, unless you have approrpritae permissions.

Dimensions & Measures:

View	Dimension	Description	Comment
Asset	Business Key	GUID used to identify the asset for end users.	NA
Asset	Is Unit Revision School Year?	Is the unit revision a school year (e.g. 2018-2019)?	NA
Asset	Metadata	Additional information or keywords describing an asset.	NA

Explore: Student Work (Curric

Search Google or type a URL

Amplify.

Need help? Find us in the Slack channel: #support-looker

Explore: Student Work (Curriculum)

Why use this Explore?

Use this Explore to analyze student work in the Curriculum App for Science and ELA. This Explore is based on the submissions to the Curriculum App, and joins in additional information about the content of ELA and Science, the users, and licenses.

Examples

- How many activities were handed in last week for each product?
  This query use a count of submissions, a date filter, and pivots on product.
- Which Units have the most submissions?
  This query uses a date range filter and groups the count of submissions by Unit, across ELA and Science. How would you add a count of users?

Caveats

- Inc[illegible] Cla[illegible] may lead to duplication of student-level submission data.
- PII is hidden by default, unless you have approrpritae permissions.

Looker API

Dimensions & Measures:

View	Dimension	Description	Comment
Asset	Business Key	GUID used to identify the asset for end users.	NA
Asset	Is Unit Revision School Year?	Is the unit revision a school year (e.g. 2018-2019)?	NA
Asset	Metadata	Additional information or keywords describing an asset.	NA

Metadata store

Amplify.

1. Data Policy
2. Data Sources
3. Data Glossary
   - About the Explores
   - Explore: Student Work (Curriculum)
   - Events (Curriculum App)
   - Explore: Evidence App
   - Explore: Core Curriculum Class Summary
4. Use Cases

Need help? Find us in the Slack channel: #support-looker

Edit this page

Explore: Student Work (Curriculum)

Why use this Explore?

Use this Explore to analyze student work in the Curriculum App for Science and ELA. This Explore is based on the submissions to the Curriculum App, and joins in additional information about the content of ELA and Science, the users, and licenses.

Examples

- How many activities were handed in last week for each product?
  This query use a count of submissions, a date filter, and pivots on product.
- Which Units have the most submissions?
  This query uses a date range filter and groups the count of submissions by Unit, across ELA and Science. How would you add a count of users?

Caveats

- Including dimensions from the 'Class' view may lead to duplication of student-level submission data.
- PII is hidden by default, unless you have approrpritae permissions.

Dimensions & Measures:

View	Dimension	Description	Comment
Asset	Business Key	GUID used to identify the asset for end users.	NA
Asset	Is Unit Revision School Year?	Is the unit revision a school year (e.g. 2018-2019)?	NA
Asset	Metadata	Additional information or keywords describing an asset.	NA

Metadata Store

- TOML files stored in a GitHub repo
- Allow arbitrary but structured metadata

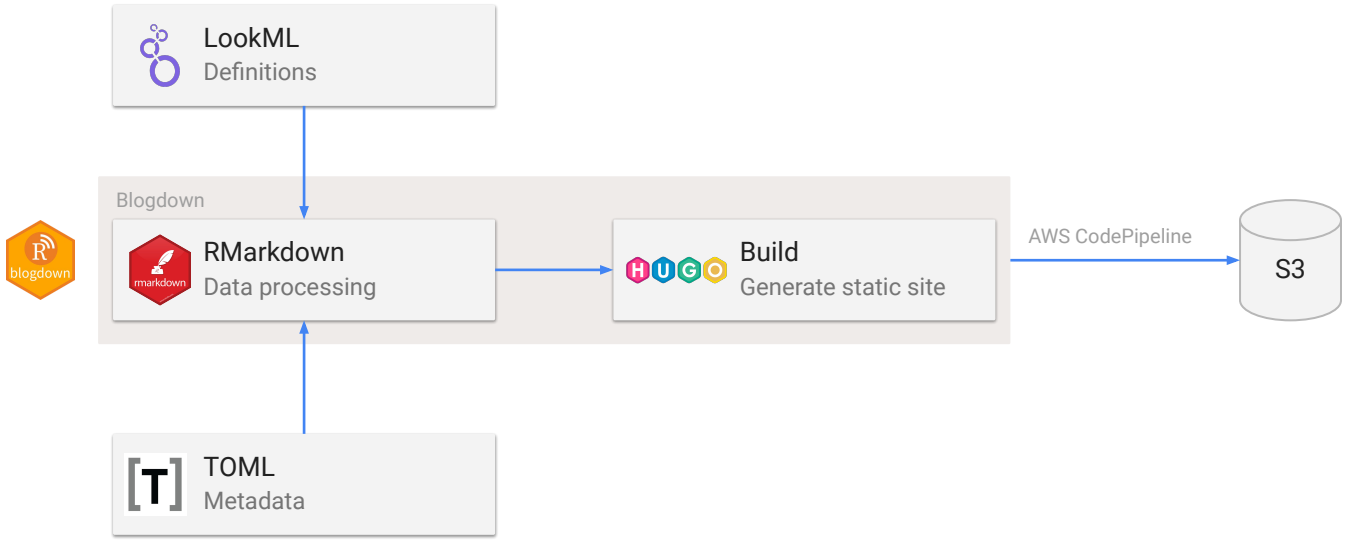
```
title = "ELA Evidence App Events"
type  = "view"

[lookml]
file = "ela_evidence_app_events.view.lkml"
view = "ela_evidence_app_event"
url  = "https://looker-
data.prod.learning.amplify.com/projects/new_curriculum/files/ela_evidence_app_events.view.lkml"
derived_table = true

[explore]
product_family = "ELA"
product = "Middle School ELA"
use = "This view contains the xAPI event data from the ELA Evidence App in tabular form. The Evidence
App was released for the 2019-2020 school year and is used by middle school ELA students. Teachers do
not use the app, unless they are doing a demo."

[dimensions]
"dim_group.group_business_key" = "May lead to duplication of results."
"dim_group.name" = "May lead to duplication of results."
"retail_dim_user_license_asset.asset_name" = "See information about the [Retail Service](https://apps-
studios.demo.learning.amplify.com/reports/dataportal/data-sources/retail-service/)"
```

Data Hub: a blogdown site



Questions?

© 2019 Looker. All rights reserved. Confidential.

looker

Thank you

Rate this session in
the JOIN mobile app

#JOINdata
@lookerdata