Vizier DB - WebUser Interface Documentation

Release 1.0

New York University

May 30, 2018

Contents

1 Contents **3**

1.1 Install and Run 3

1.2 Getting Started 5

2 Links **11**

3 Indices and tables **13**

Vizier is a new powerful tool to streamline the data curation process. Data curation (also known as data preparation, wrangling, or cleaning) is a critical stage in data science in which raw data is structured, validated, and repaired. Data validation and repair establish trust in analytical results, while appropriate structuring streamlines analytics.

Vizier makes it easier and faster to explore and analyze raw data by combining a simple notebook interface with spreadsheet views of your data. Powerful back-end tools that track changes, edits, and the effects of automation. These forms of provenance capture both parts of the exploratory curation process - how the cleaning workflows evolve, and how the data changes over time.

Vizier is a collaboration between the **University at Buffalo**, **New York University**, and the **Illinois Institute of Technology**.

CHAPTER 1

Contents

1.1 Install and Run

Before installing Vizier DB Web UI, you should install VizierDB - Web API. The Web API is the backend that provides the API that is used by the Vizier DB Web UI.

1.1.1 Install VizierDB - Web API

Installation is still a bit labor intensive. The following steps seem to work for now (requires [Anaconda](https://conda.io/docs/user-guide/install/index.html)). If you want to use Mimir modules within your curation workflows a local installation of Mimir v0.2 is required. Refer to this [guide for Mimir installation details](https://github.com/VizierDB/Vistrails/tree/MimirPackage/vistrails/packages/mimir).

Python Environment

To setup the Python environment clone the repository and run the following commands:

```
>>> git clone https://github.com/VizierDB/web-api.git
>>> cd web-api
>>> conda env create -f environment.yml
>>> source activate vizier
>>> pip install git+https://github.com/VizierDB/Vistrails.git
>>> pip install -e .
```

As an alternative the following sequence of steps might also work (e.g., for MacOS):

```
>>> git clone https://github.com/VizierDB/web-api.git
>>> cd web-api
>>> conda create --name vizier pip
>>> source activate vizier
>>> pip install -r requirements.txt
```

(continues on next page)

(continued from previous page)

```
>>> pip install -e .
>>> conda install pyqt=4.11.4=py27_4
```

Configuration

The web server is configured using a configuration file. There are two example configuration files in the (config directory)[https://github.com/VizierDB/web-api/tree/master/config] (depending on whether including Mimir `` `config-mimir.yaml` `` or not `` `config-default.yaml` ``). The configuration paramaters are:

api - *server_url*: Url of the server (e.g., http://localhost) - *server_port*: Server port (e.g., 5000) - *app_path*: Application path for Web API (e.g., /vizier-db/api/v1) - *app_base_url*: Concatenation of server_url, server_port and app_path - *doc_url*: Url to API documentation

fileserver - *directory*: Path to base directory for file server - *max_file_size*: Maximum size for file uploads

engines - *identifier*: Engine type (i.e., DEFAULT or MIMIR) - *name*: Engine printable name - *description*: Descriptive text for engine - *datastore*:

- directory: Base directory for data store

viztrails

- *directory*: Base directory for storing viztrail information and meta data

name: Web Service name

debug: Flag indicating whether server is started in debug mode

logs: Path to log directory

When the Web server starts it first looks for the configuration file that is reference in the environment variable `` `VIZIERSERVER_CONFIG` ``. If the variable is not set the server looks for a file `` `config.yaml` `` in the current working directory.

Note that there is a `` `config.yaml` `` file in the working directory of the server that can be used for development mode.

Run Server

After adjusting the server configuration the server is run using the following command:

```
>>> cd vizier
>>> python server.py
```

Make sure that the conda environment has been activated using `` `source activate vizier` ``.

If using Mimir the gateway server sould be started before running the web server.

API Documentation

For development it can be helpful to have a local copy of the API documentation. The [repository README](https://github.com/VizierDB/webapi-swagger-ui) contains information on how to install the UI locally.

1.1.2 Install VizierDB - Web UI

Start by cloning the repository and switching to the app directory.

```
>>> git clone https://github.com/VizierDB/web-ui.git
>>> cd web-ui
```

Inside the app directory, you can run several commands:

Install build dependencies

```
>>> yarn install
```

Start the development server

```
>>> yarn start
```

Bundles the app into static files for production

```
>>> yarn build
```

Additional Commands

Starts the test runner.

```
>>> yarn test
```

Remove this tool and copies build dependencies, configuration files and scripts into the app directory. If you do this, you can't go back!

```
>>> yarn eject
```

Configuration

The UI app connects to the Web API server. The Url for the server is currently hard-coded in the file `` `public/env.js` ``. Before running `` `yarn start` `` adjust the Url to point to a running Web API server. By default a local server running on port 5000 is used.

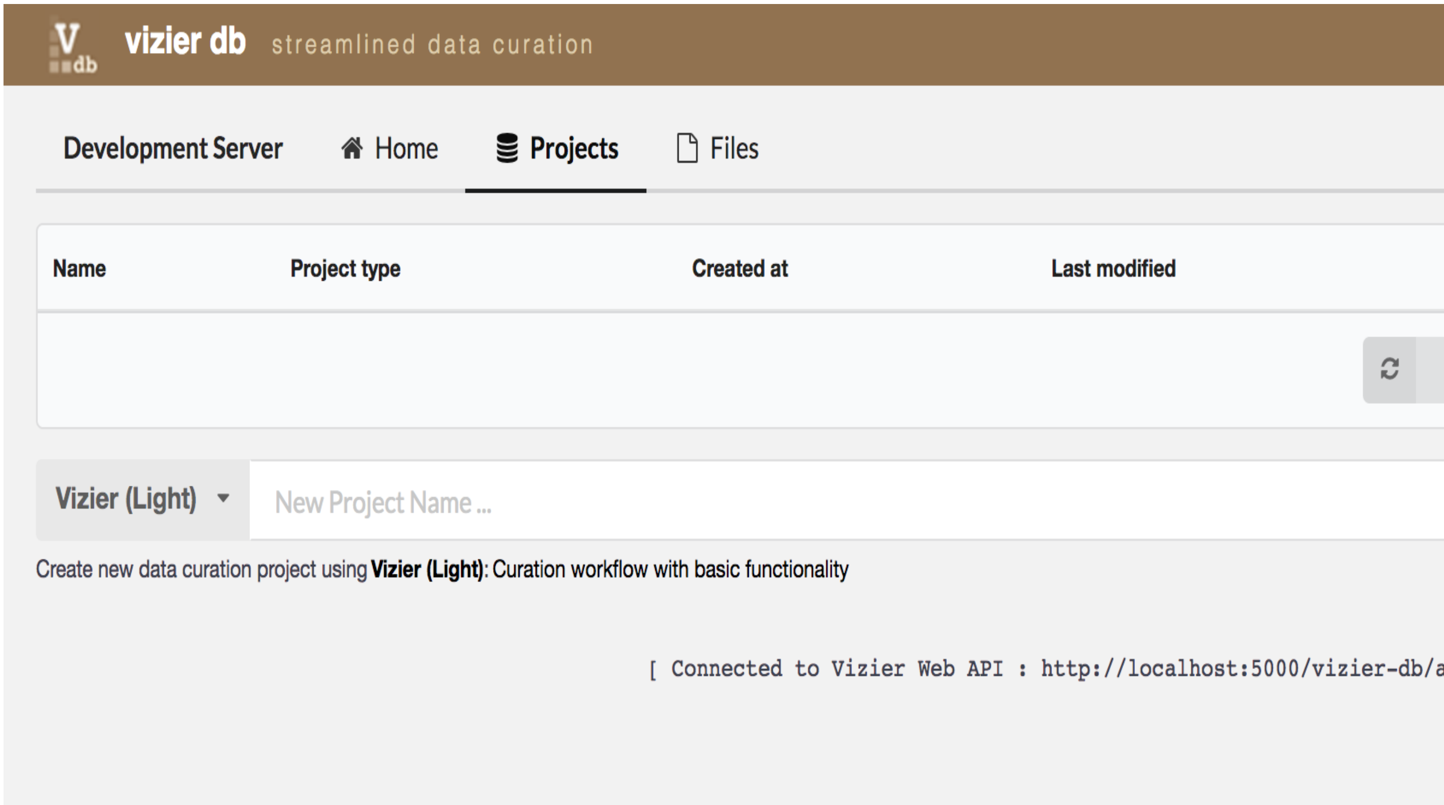
1.2 Getting Started

Vizier organizes data curation workflows into projects.

- Start by selecting or creating a new project under the Projects Tab.
- If the data that you want to clean is currently stored in CSV files, these files have to be uploaded to the file server. You can upload your data files under the Files Tab.

1.2.1 Step 1

Create Project



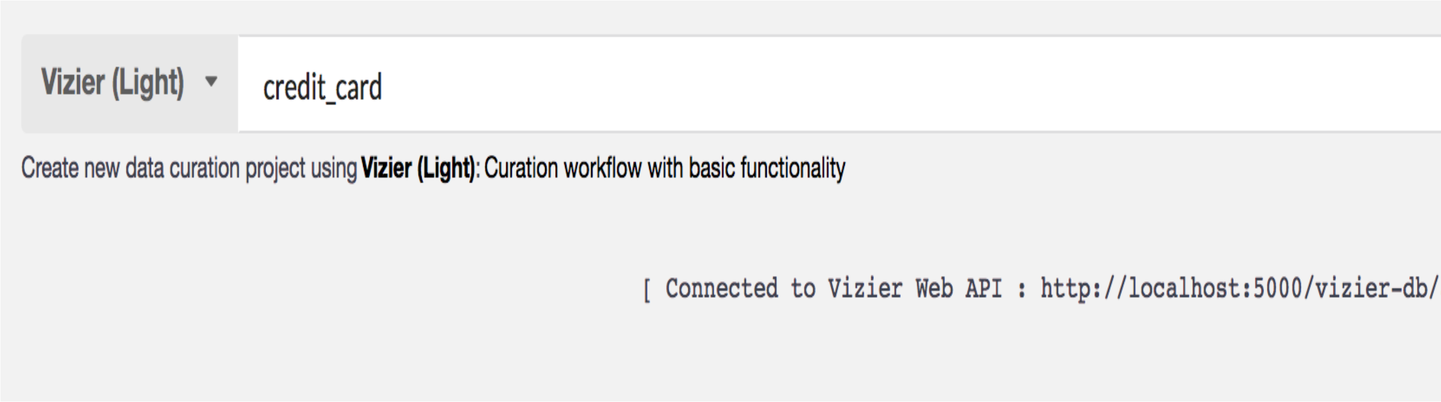
Name	Project type	Created at	Last modified

Vizier (Light) ▾ New Project Name ...

Create new data curation project using **Vizier (Light)**: Curation workflow with basic functionality

```
[ Connected to Vizier Web API : http://localhost:5000/vizier-db/a
```

Begin by adding a project on the Vizier page (initial page), shown in the figure above, by clicking on the **Projects** tab button.

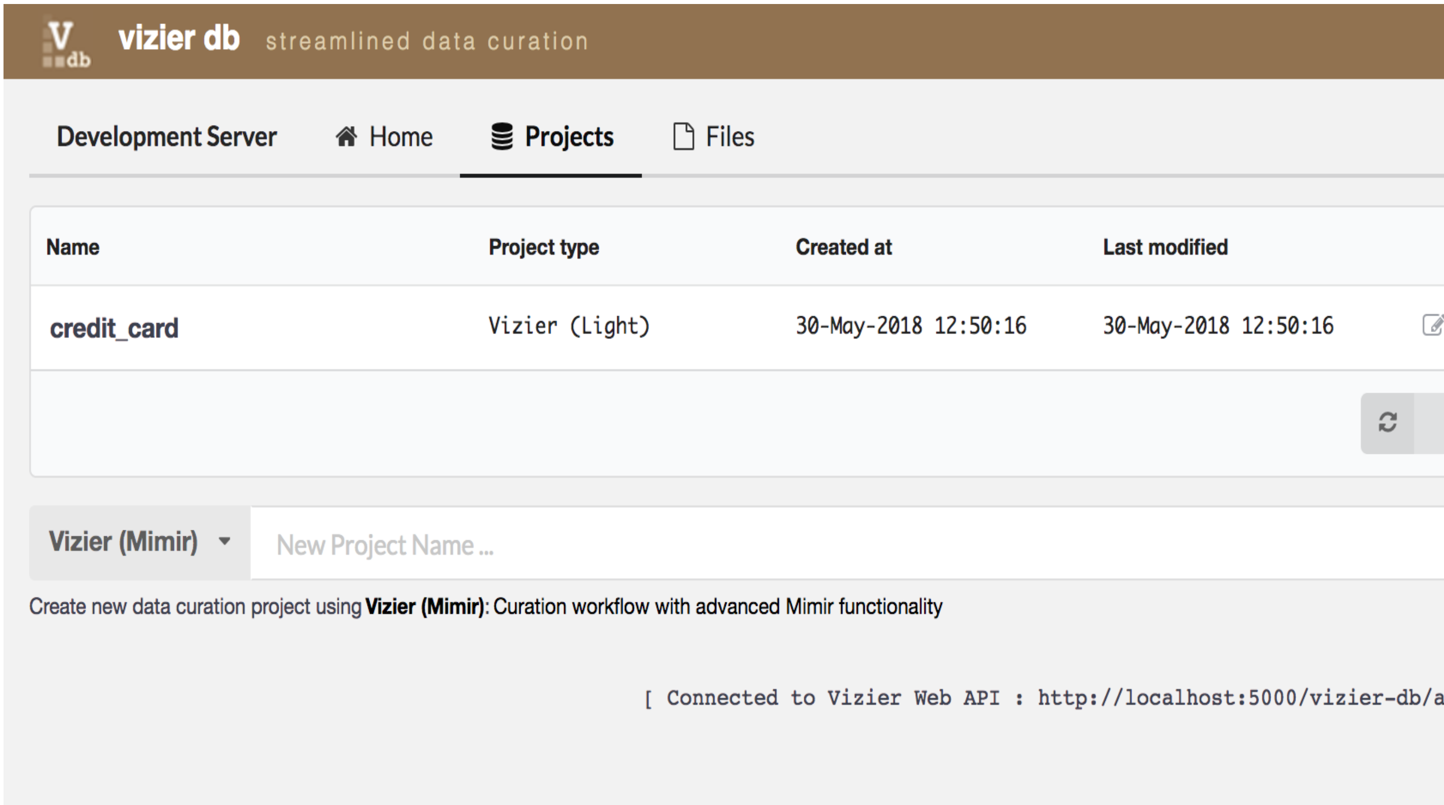


Vizier (Light) ▾ credit_card

Create new data curation project using **Vizier (Light)**: Curation workflow with basic functionality

```
[ Connected to Vizier Web API : http://localhost:5000/vizier-db/
```

On the **New Project Name ...** textbox shown in figure above, enter the name of the project you would like to create, for example **credit_card**, and click on **+** button. You should now see the new project you added in the list of projects as shown below.



Name	Project type	Created at	Last modified
credit_card	Vizier (Light)	30-May-2018 12:50:16	30-May-2018 12:50:16

Create new data curation project using **Vizier (Mimir)**: Curation workflow with advanced Mimir functionality

```
[ Connected to Vizier Web API : http://localhost:5000/vizier-db/a
```

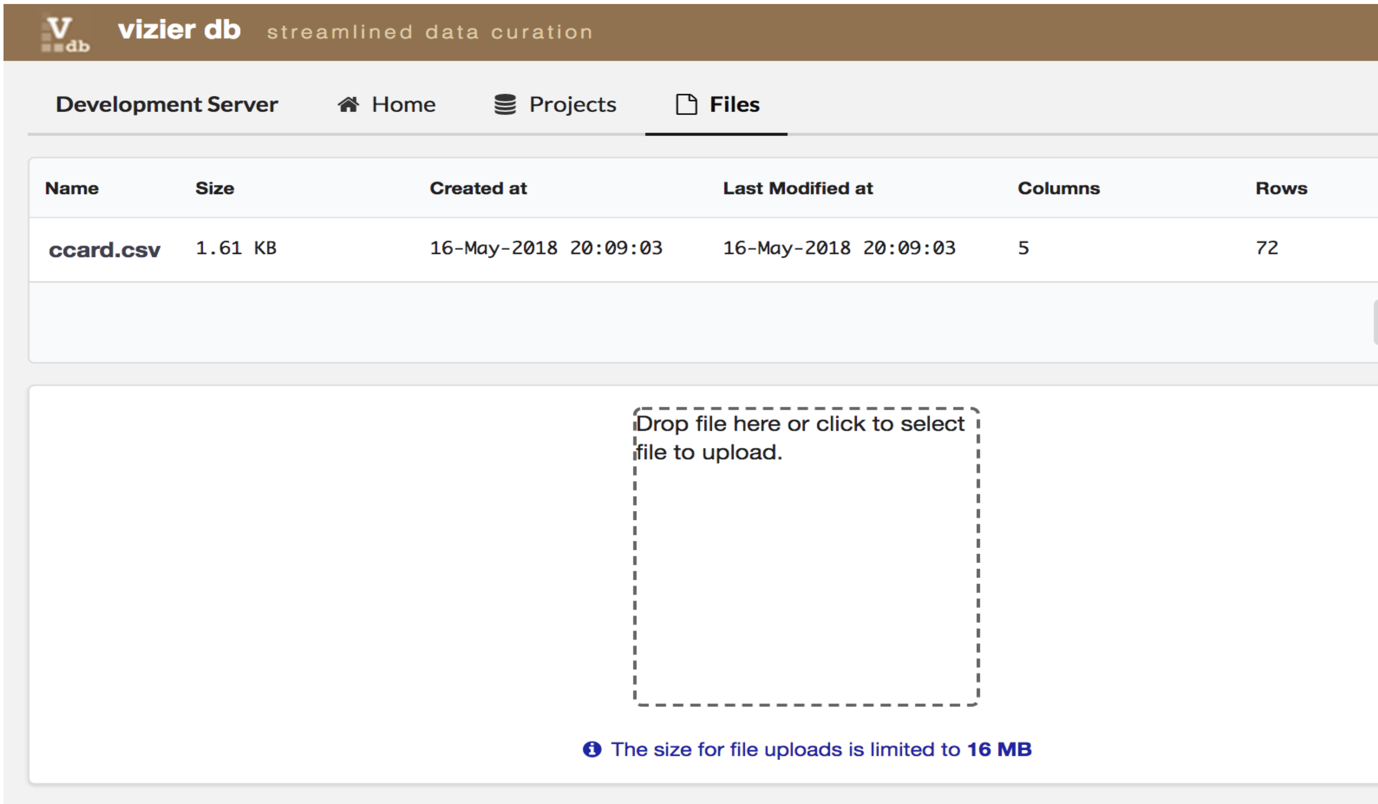
Once project is added click on project name in the list of projects to data curation.

1.2.2 Step 2

Load Dataset

Continuing with our example of the credit_card project, we show here the methods of uploading data.

First at all come back to the initial page of Vizier. If the data that you want to clean is currently stored in CSV files, these files have to be uploaded to the file server. If you want to upload your own data file, then go under the **Files** Tab.



Name	Size	Created at	Last Modified at	Columns	Rows
ccard.csv	1.61 KB	16-May-2018 20:09:03	16-May-2018 20:09:03	5	72

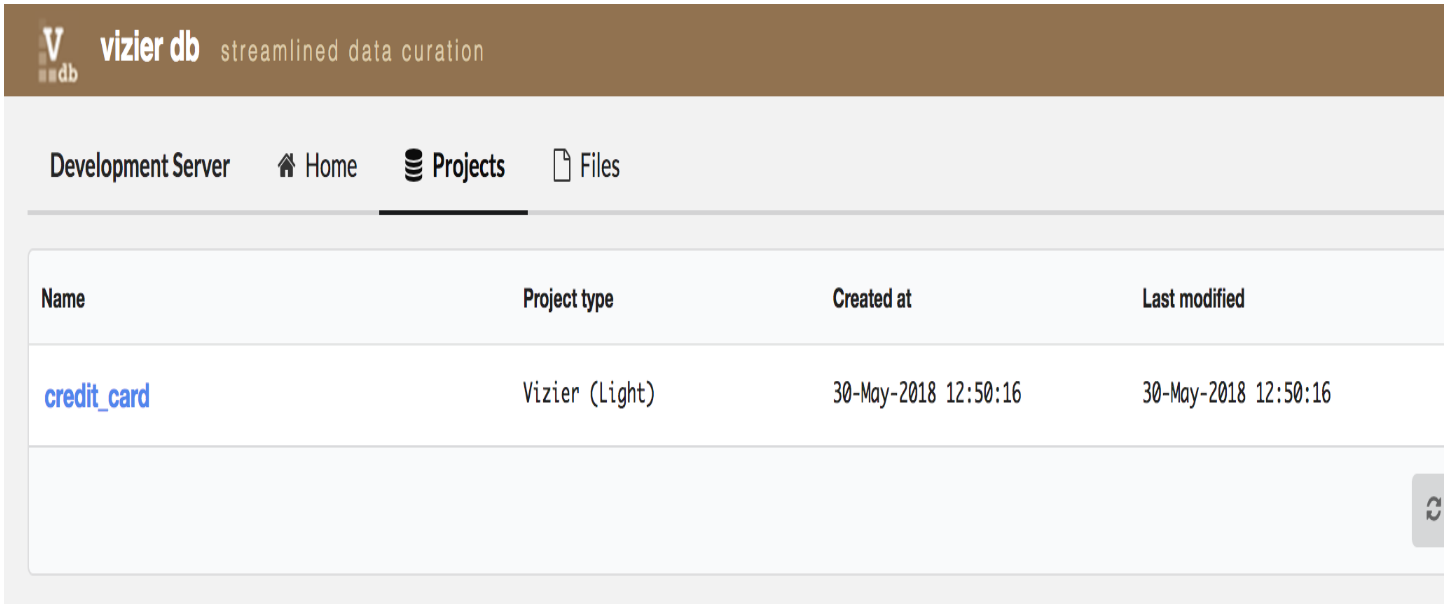
Drop file here or click to select file to upload.

The size for file uploads is limited to **16 MB**

1.2.3 Step 3

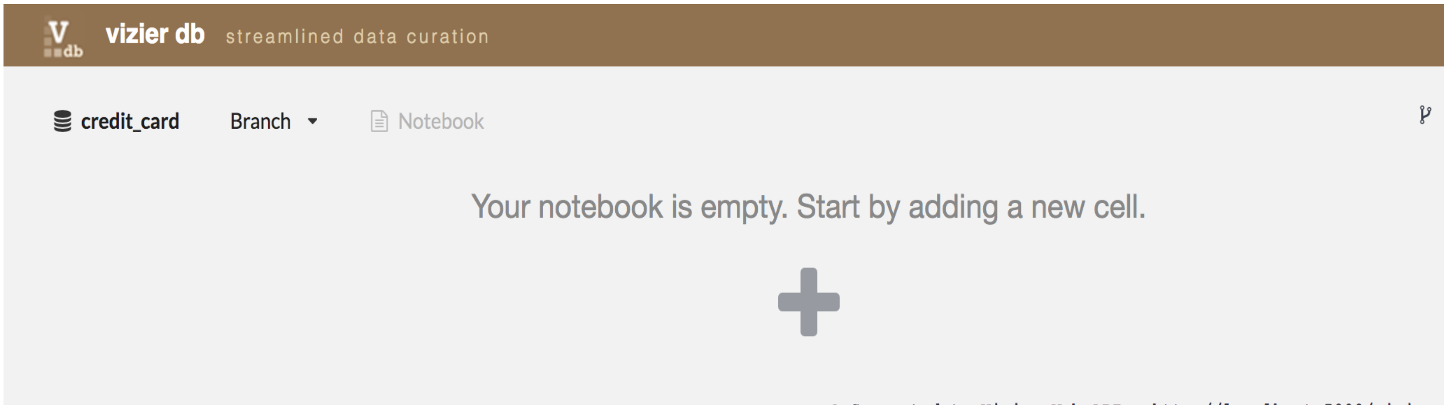
Loading Dataset in Project

First, go to the **Project** tab. There, you will be able to see the list of projects. Select one, for example, **credict_card** project by clicking on the name project.



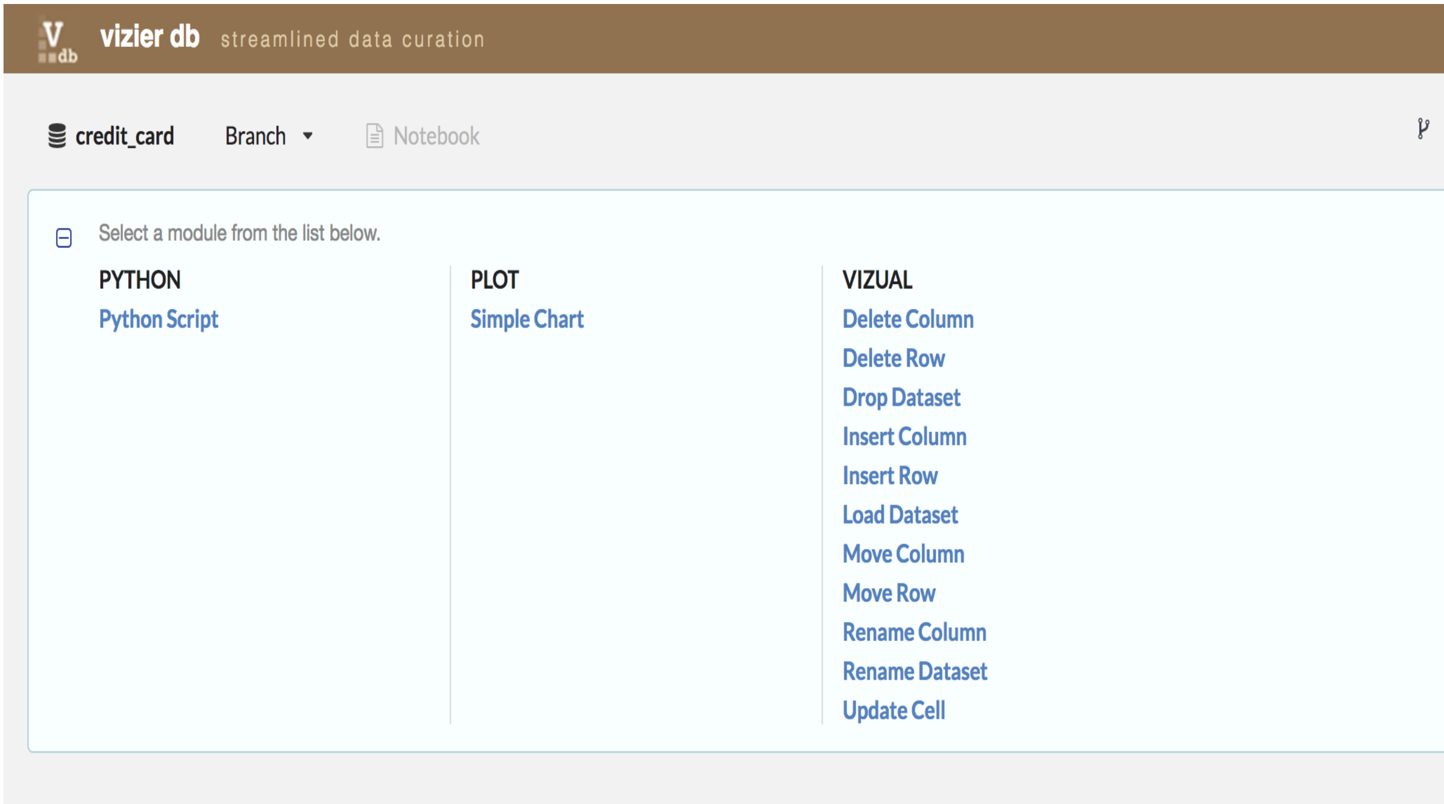
Name	Project type	Created at	Last modified
credit_card	Vizier (Light)	30-May-2018 12:50:16	30-May-2018 12:50:16

Once you are inside the project, load the data by clicking in the sign **+**.



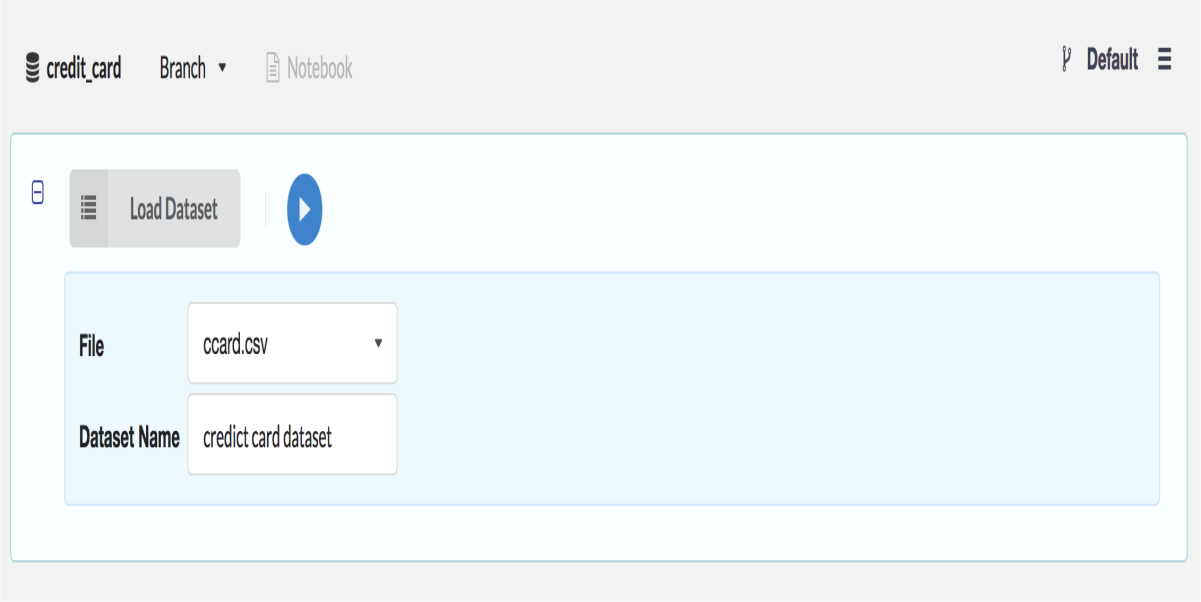
Your notebook is empty. Start by adding a new cell.

Then, go to the column **VIZUAL**, and click on **Load Dataset**



Select a module from the list below.

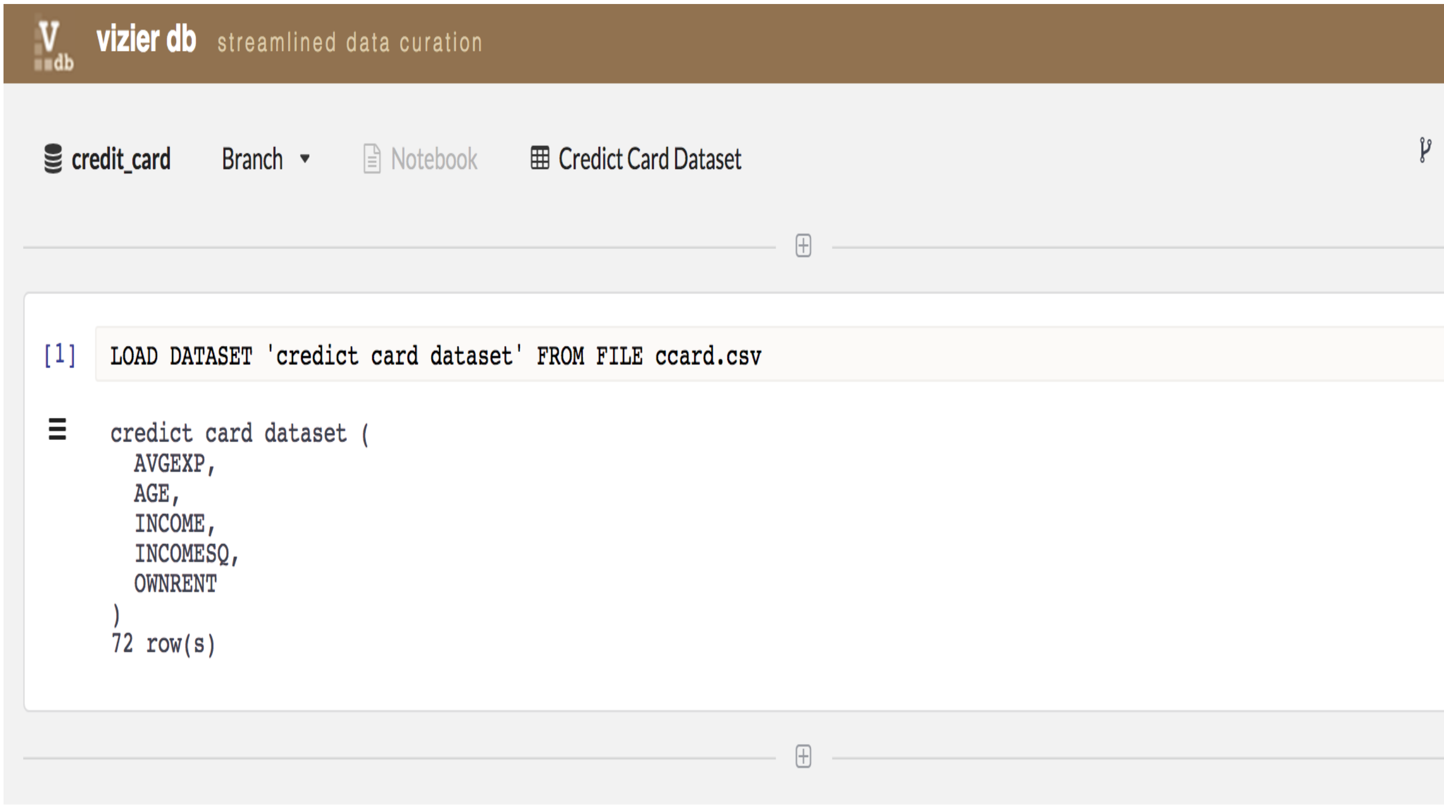
Then, select a dataset listed in **File** ComboBox. For example, we selected ccard.csv dataset and entered **credict card dataset** as the name of the dataset for that project, then, click on the blue **play** icon.



File: ccard.csv

Dataset Name: credict card dataset

After loading the **credict card dataset**, we can start to explore and curate our data.



```
[1] LOAD DATASET 'credict card dataset' FROM FILE ccard.csv
```

```
credict card dataset (
  AVGEXP,
  AGE,
  INCOME,
  INCOMESQ,
  OWNRENT
)
72 row(s)
```

CHAPTER 2

Links

- GitHub repository

CHAPTER 3

Indices and tables

- genindex
- modindex
- search