

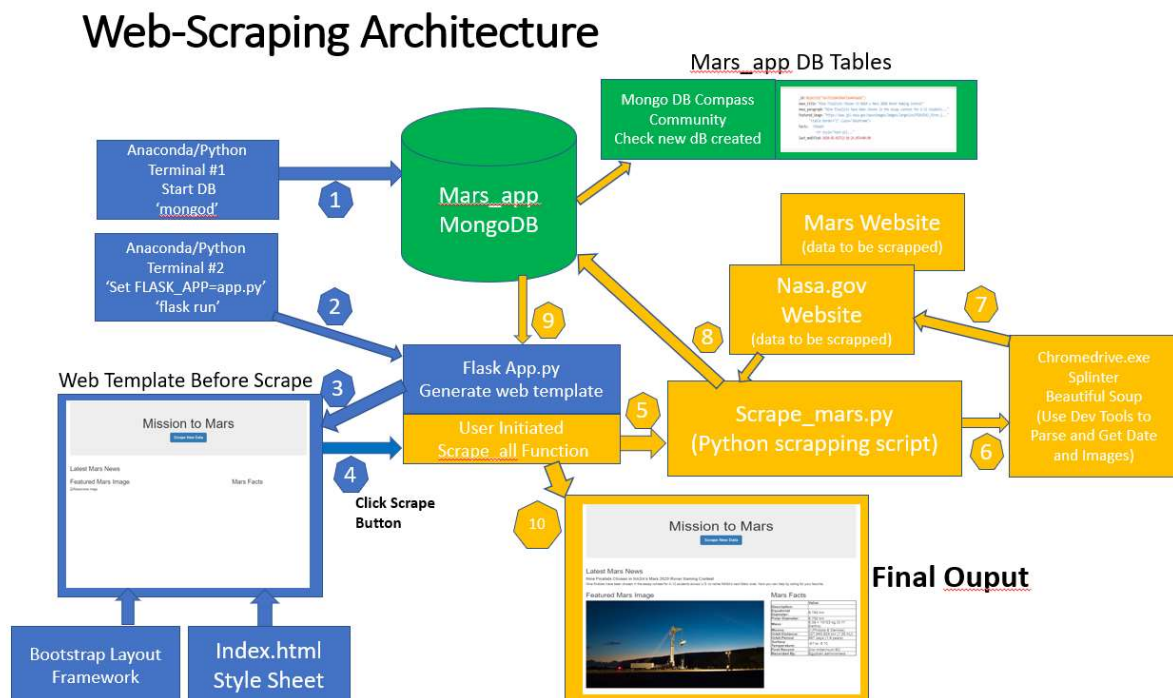
Module 10 - Web Scraping Architecture and Data Flow

This repository hold several files to scrape key information and images from several Mars related websites, store them into a mongo DB and retrieve them for display on a custom webpage. The images and information can be updated with new information by the user with the use of a "web Scrapping" button - which calls the scrape_all object and returns updated information to the Mongo DB for rendering on the web page.

Resources

- Python 3.6 environment
- Import Pandas
- Import datetime
- Import BeautifulSoup
- Splinter Import Browser
- Flask
- Pymongo
- MongoDB
- Bootstrap

Web-Scraping Architecture Diagram



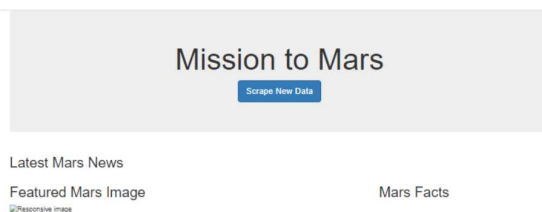
Seudo Code on Data Flow

- 1) Open Python terminal and issue the 'mongod' command to prepare the database for startup.
- 2) Open a second Python terminal and issue the 'Set FLASK_APP=app.py' and "flask run" commands to run the Flask App.
- 3) Using Bootstrap CDN and the local Index.html file, render the template web page.
 - a. Home page is found at root directory <http://127.0.0.1:5000>
- 4) User then initiates the "Scrape" cycle by clicking the "New Scrape" button on the home page.
- 5) Flask calls the "scrape_mars.py" file (imported into FLASK) and runs the 3 subroutines.
- 6) Using Splinter and BeautifulSoup, a separate scraping browser is started up, directed to the web pages by splinter and then using the Chromedriver Developer Tools, Beautiful soup parses the HTML looking for specific "tag" or "text" = matches.
- 7) Websites are jpl.nasa.gov and mars.nasa.gov.
- 8) Text data and images are stored in defined dictionaries and then stored in the Mongo DB entitled mars_app DB. A message is issued "Scrape Successful" and the web browser lands on <http://127.0.0.1:5000/scrape>.
- 9) Flask then pulls the data and sends it back to the web server and renders within the framework defined by Bootstrap and Index.html.
- 10) The user needs to go back to the home page at <http://127.0.0.1:5000> and will see the fully rendered web page with data and images as:
 - a. Latest News Article
 - b. Featured space image
 - c. Table of Mars Facts

Key Files

- Flask file entitled "app.py"
- Index.html
- Scrap_mars.py

Web Output Prior to Scrape



Web Output After Scrape

Mission to Mars


Scrape New Data

Latest Mars News

Nine Finalists Chosen in NASA's Mars 2020 Rover Naming Contest

Nine finalists have been chosen in the essay contest for K-12 students across U.S. to name NASA's next Mars rover. Now you can help by voting for your favorite.

Featured Mars Image



Mars Facts

Description	Value
Equatorial Diameter:	6,792 km
Polar Diameter:	6,752 km
Mass:	6.39×10^{23} kg (0.11 Earth's)
Moons:	2 (Phobos & Deimos)
Orbit Distance:	227,943,824 km (1.38 AU)
Orbit Period:	687 days (1.9 years)
Surface Temperature:	-87 to -5 °C
First Record:	2nd millennium BC
Recorded By:	Egyptian astronomers

Challenge 10

The Challenge 10 assignment was to:

- Add 4 more images of the Mars Hemisphere from the JPL website.
- This required modifying the scraping-mars file using several unfamiliar languages known as Splinter, BeautifulSoup as well as HTML.
- The Flask App then needed to be modified to ensure these additional items were scrapped and variables defined and results returned for storage in the Mongo DB.
- The Index.html and Bootstrap templates then needed to be modified to accommodate this additional information.
- Work-In-Progress was uploaded to the repository as
 - App_c.py
 - Index_c.html
 - Scrape_mars_c.py

These files are not yet complete and will be updated as additional learning can be accomplished on the Splinter, BeautifulSoup and HTML languages.