



zen-gineer Update README.md

00b3b74 on Feb 7

[1 contributor](#)

Raw Blame History



71 lines (43 sloc) 2.68 KB

Mission to Mars

This is a web application that scrapes various websites for data related to the Mission to Mars and displays the information in a single HTML page. The following outlines the details.

🔗 Step 1 - Scraping

Initial scraping used Jupyter Notebook, BeautifulSoup, Pandas, and Requests/Splinter is recommended so you can visualize as you go. It can be moved to Python Flask later.

NASA Mars News

- Scraped the [NASA Mars News Site](#) and collected the latest News Title and Paragraph Text.

Example:

```
news_title = "NASA's Next Mars Mission to Investigate Interior of Red Planet"
```

```
news_p = "Preparation of NASA's next spacecraft to Mars, InSight, has ramped up this summer, on course f
```

JPL Mars Space Images - Featured Image

- Next visited the url for JPL's Featured Space Image [here](#).
- Splinter was used to navigate the site and find the image url for the current Featured Mars Image and assign the url string to the variable `featured_image_url`.

Example:

```
featured_image_url = 'https://www.jpl.nasa.gov/spaceimages/images/largesize/PIA16225_hires.jpg'
```

Mars Weather

- Next visited the Mars Weather twitter account [here](#) was scraped for the latest Mars weather tweet from the page. Saved the weather report in the variable `mars_weather`.

Example:

```
mars_weather = 'Sol 1801 (Aug 30, 2017), Sunny, high -21C/-5F, low -80C/-112F, pressure at 8.82 hPa, day
```

Mars Facts

- Next visited the Mars Facts webpage [here](#) Used Pandas to scrape the table containing facts about the planet including Diameter, Mass, etc and convert the data to an html table string.

Mars Hemispheres

- Next visited the USGS Astrogeology site [here](#) to obtain high resolution images for each of Mar's hemispheres.

```
# Example:
hemisphere_image_urls = [
    {"title": "Valles Marineris Hemisphere", "img_url": "..."},
    {"title": "Cerberus Hemisphere", "img_url": "..."},
    {"title": "Schiaparelli Hemisphere", "img_url": "..."},
    {"title": "Syrtis Major Hemisphere", "img_url": "..."},
]
```

Step 2 - MongoDB and Flask Application

Used MongoDB with Flask templating to create a new HTML page that displays all of the information that was scraped from the URLs above.

Requirements

- Splitter, BeautifulSoup, Pymongo, Bootstrap, Flask