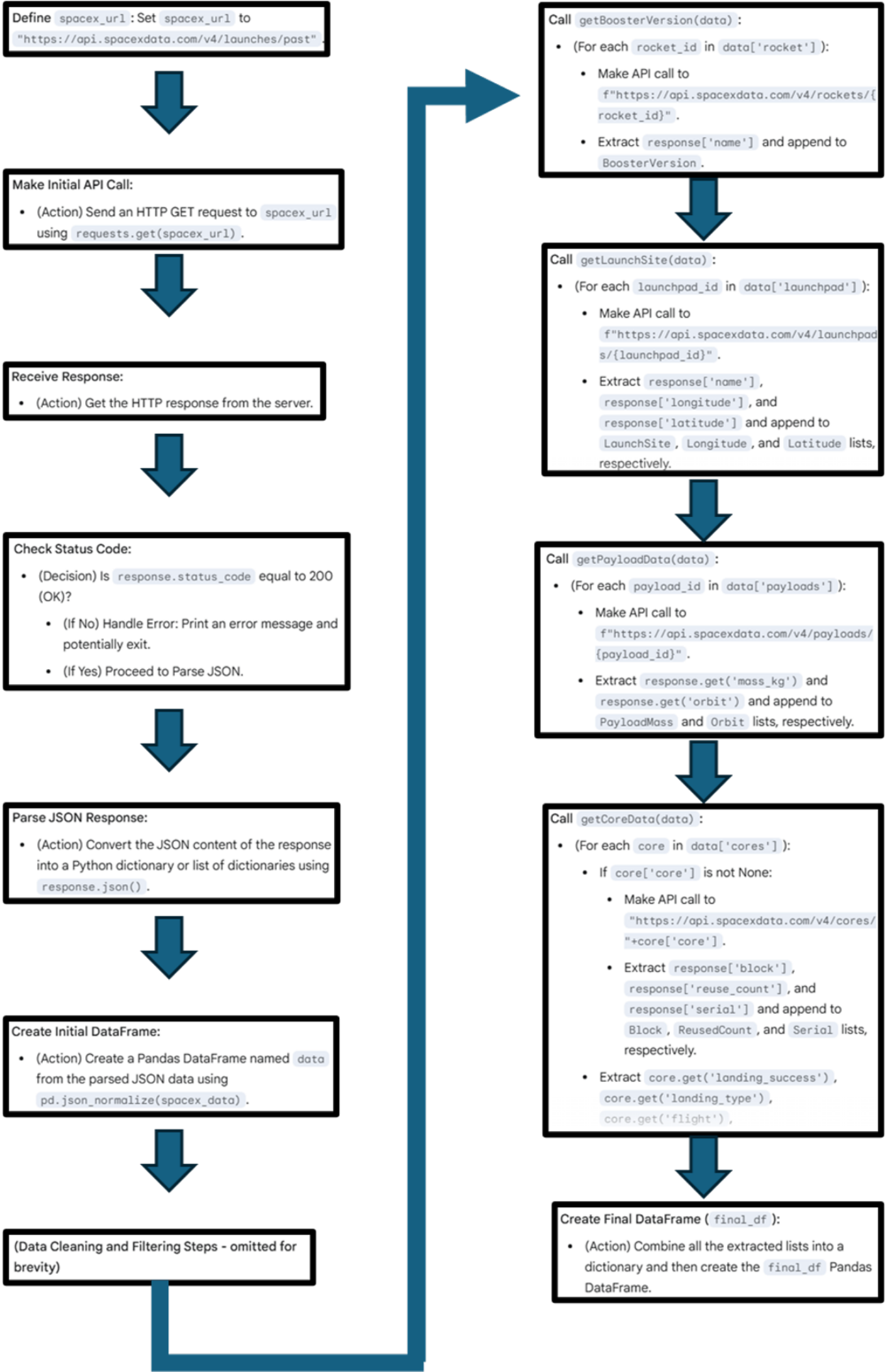
Define spacex_url : Set spacex_url to "https://api.spacexdata.com/v4/launches/past" .
Make Initial API Call:
(Action) Send an HTTP GET request to spacex_url using requests.get(spacex_url) .
Receive Response:
(Action) Get the HTTP response from the server.
Check Status Code:
(Decision) Is response.status_code equal to 200 (OK)?
(If No) Handle Error: Print an error message and potentially exit.
(If Yes) Proceed to Parse JSON.
Parse JSON Response:
(Action) Convert the JSON content of the response into a Python dictionary or list of dictionaries using response.json() .
Create Initial DataFrame:
(Action) Create a Pandas DataFrame named data from the parsed JSON data using pd.json_normalize(spacex_data) .
(Data Cleaning and Filtering Steps - omitted for brevity)
Call getBoosterVersion(data) :
(For each rocket_id in data['rocket']):
Make API call to f"https://api.spacexdata.com/v4/rockets/{rocket_id}" .
Extract response['name'] and append to BoosterVersion .
Call getLaunchSite(data) :
(For each launchpad_id in data['launchpad']):
Make API call to f"https://api.spacexdata.com/v4/launchpads/{launchpad_id}" .
Extract response['name'] , response['longitude'] , and response['latitude'] and append to LaunchSite , Longitude , and Latitude lists, respectively.
Call getPayloadData(data) :
(For each payload_id in data['payloads']):
Make API call to f"https://api.spacexdata.com/v4/payloads/{payload_id}" .
Extract response.get('mass_kg') and response.get('orbit') and append to PayloadMass and Orbit lists, respectively.
Call getCoreData(data) :
(For each core in data['cores']):
If core['core'] is not None:
Make API call to "https://api.spacexdata.com/v4/cores/"+core['core'] .
Extract response['block'] , response['reuse_count'] , and response['serial'] and append to Block , ReusedCount , and Serial lists, respectively.
Extract core.get('landing_success') , core.get('landing_type') , core.get('flight') ,
Create Final DataFrame (final_df):
(Action) Combine all the extracted lists into a dictionary and then create the final_df Pandas DataFrame.