Query String and Request Body

Brian Dickinson

University of Rochester

Fall 2019
Why not just URL Paths?

- In paths the order is important and every option must be routed or return 404.
- All parameters in the URL are necessarily visible to others even with HTTPS.
- While a web application can work with just these variables it is not scalable and is more difficult to design and maintain.
What is the Query String?

Query strings are used to submit query parameters to the server. They are commonly used in form submissions.

http://www.example.com/user?firstName=Brian&lastName=Dickinson

?firstName=Brian&lastName=Dickinson

- Query strings begin with a question mark ?
- Keys and values are separated by an equal sign =
- Key value pairs are separated by an ampersand &
GET /users?firstName=Brian&lastName=Dickinson HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: example.com
Content-Type: text/html; charset=utf-8
Accept-Language: en-us
What is the Request Body?

GET /users HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: example.com
Content-Type: text/html; charset=utf-8
Accept-Language: en-us

firstName=Brian&lastName=Dickinson
Query String vs Request Body

- With parameters in the query string, they may be bookmarked and shared (may or may not be desirable). They are also visible to others by requirement since they are part of the URL that is used to direct the request.

- With parameters in the request body, they can be protected by HTTPS. They cannot however be used to bookmark or share this request (may or may not be desired).

- The query string is limited in size. The limit varies by platform and it is safest to use the request body for large submissions.
Flask processes each request in a request context to make it easy to access both types of parameters. We list a few attributes below. A more complete table is in the docs and in Chapter 2 of your book.

- `request.query_string` - raw unparsed query string as a string
- `request.args` - dictionary of key,value pairs from the query string
- `request.get_data()` - string of raw request body
- `request.form` - dictionary of key,value pairs from the request body
#!/usr/bin/env python3

from flask import Flask, request
app = Flask(__name__)

@app.route('/users', methods=['GET'])
def hello():
    fname = request.args.get("firstName", "John")
    lname = request.args.get("lastName", "Doe")
    return f"Searching for user {fname} {lname}\n", 200

if __name__=="__main__":
    app.run()
#!/usr/bin/env python3

from flask import Flask, request
app = Flask(__name__)

@app.route('/users', methods=['POST'])
def hello():
    fname = request.form.get("firstName", "John")
    lname = request.form.get("lastName", "Doe")
    return f"Searching for user {fname} {lname}\n", 200

if __name__=="__main__":
    app.run()
**Bibliography I**

- **Fullsour.**
  When should you use path variable and query parameter?
  
  https://medium.com/@fullsour/when-should-you-use-path-variable-and-query-parameter-a34679e8a6d


- **Pallets Team.**
  The request context.
  
  http://flask.pocoo.org/docs/1.0/reqcontext/.