

Recitation Week - 11

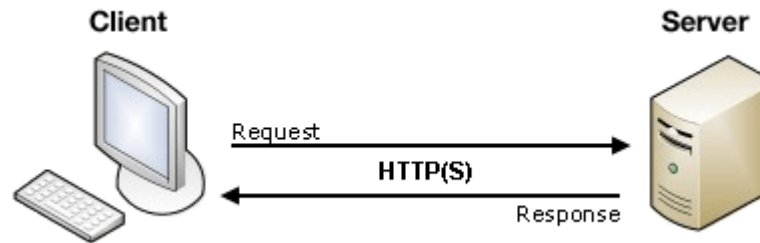
CS1520 TA: PRANUTJAIN@PITT.EDU

Plan for today

HTTP and client-server communication

AJAX

Client/server architecture



a client (usually a web browser) sends a request to a server (most of the time a web server like [Apache](#), [Nginx](#), [IIS](#), [Tomcat](#), etc.), using the [HTTP protocol](#).

The server answers the request using the same protocol.

An HTML form on a web page is nothing more than a convenient user-friendly way to configure an HTTP request to send data to a server. This enables the user to provide information to be delivered in the HTTP request.

How to send the data

The `<form>` element defines how the data will be sent.

All of its attributes are designed to let you configure the request to be sent when a user hits a submit button.

The two most important attributes are **action** and **method**.

Action Attribute

The action attribute defines where the data gets sent.

```
<form action="https://example.com">
```

the data is sent to an absolute URL — <https://example.com>:

```
<form action="/somewhere_else">
```

the data is sent to a different URL on the same origin

The method attribute

The method attribute defines how data is sent.

The HTTP protocol provides several ways to perform a request; HTML form data can be transmitted via a number of different methods, the most common being the **GET** method and the **POST** method.

An HTTP request consists of two parts: a header that contains a set of global metadata about the browser's capabilities, and a body that can contain information necessary for the server to process the specific request.

The GET method

The GET method is the method used by the browser to ask the server to send back a given resource: **"Hey server, I want to get this resource."**

The body for the GET request is empty.

if a form is sent using this method the data sent to the server is appended to the URL

The GET method

```
<form action="http://www.foo.com" method="GET">
  <div>
    <label for="say">What greeting do you want to say?</label>
    <input name="say" id="say" value="Hi">
  </div>
  <div>
    <label for="to">Who do you want to say it to?</label>
    <input name="to" id="to" value="Mom">
  </div>
  <div>
    <button>Send my greetings</button>
  </div>
</form>
```

URL `www.foo.com/?say=Hi&to=Mom`

The POST method

It's the method the browser uses to talk to the server when asking for a response that takes into account the data provided in the body of the HTTP request: **"Hey server, take a look at this data and send me back an appropriate result."**

If a form is sent using this method, the data is appended to the body of the HTTP request.

When the form is submitted using the POST method, you get no data appended to the URL

The POST method

```
<form action="http://www.foo.com" method="POST">
  <div>
    <label for="say">What greeting do you want to say?</label>
    <input name="say" id="say" value="Hi">
  </div>
  <div>
    <label for="to">Who do you want to say it to?</label>
    <input name="to" id="to" value="Mom">
  </div>
  <div>
    <button>Send my greetings</button>
  </div>
</form>
```

What does the request look like?

```
GET /?say=Hi&to=Mom HTTP/2.0  
Host: foo.com
```

```
POST / HTTP/2.0  
Host: foo.com  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 13  
  
say=Hi&to=Mom
```

Checking the request/response

Using developer tools in Chrome:

1. Open the developer tools.
2. Select "Network"
3. Select "All"
4. Select "foo.com" in the "Name" tab
5. Select "Headers"

Question

How should you send the form password to the server?

- A. HTTP GET request
- B. HTTP POST request
- C. None of the above

Server side: retrieving the data

The server receives a string that will be parsed in order to get the data as a list of key/value pairs.

```
from flask import Flask, render_template, request

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def form():
    return render_template('form.html')

@app.route('/hello', methods=['GET', 'POST'])
def hello():
    return render_template('greeting.html', say=request.form['say'], to=request.form['to'])

if __name__ == "__main__":
    app.run()
```