Lập trình mạng (Network Programming)

Chương 2. HTTP and WWW

Python packet

- Hai packet thường sử dụng
  - urllib
  - requests
- Request and Response
  - Client: khởi tạo HTTP section – mở kết nối TCP đến HTTP server và gửi request
  - Server: gửi phản hồi

Packet urllib

- Tạo request
  - from urllib.request import urlopen
  - response = urlopen(«http://www.example.com»)
- Một số thuộc tính và phương thức của đối tượng response
  - read() readline() url status
  - getheaders()
- Xử lý lỗi
  - import urllib.error
  - try ... except urrllib.error.HTTPError as e

Tùy biến request

- Thêm các header vào request trước khi gửi đi
- Tạo đối tượng request và gửi bằng urlopen()
  - Tạo đối tượng request
  - Thêm các header vào đối tượng request
  - Gửi các đối tượng request bằng urlopen
- Ví dụ
  - from urllib.request import Request
  - req = Request(«http://www.example»)
  - Thêm header: req.add_header('Accept-Language','sv')
  - Gửi: response = urlopen(req)

Tùy biến request

- Nén/mã hóa
- Header Accept-Encoding
  - Client gửi request yêu cầu nén/mã hóa trong header: Accept-Encoding
  - Server chọn phương pháp nén mà nó hỗ trợ
  - Server nén/mã nội dung message và gửi về cho client
- Ví dụ
  - Tạo request: req = Request(«www.example.com»)
  - Thêm header: req.add_header('Accept-Encoding','gzip')
  - Gửi: response = urlopen(req)
  - Kiểm tra header: response.getheader('Content-Encoding')

Tùy biến request

- Ví dụ giải nén dữ liệu bằng module gzip
- import gzip
- content = gzip.decompress(response.read())
- content.splitlines()[:5]
- Các chuẩn nén đăng ký với IANA
  - gzip, compress, deflate và identity

Các kiểu dữ liệu

- HTTP hỗ trợ nhiều kiểu dữ liệu
  - Header Content-Type trong phản hồi sẽ báo cho client biết về kiểu dữ liệu server gửi
  - r = urlopen('http://www.example.com')
  - r.getheader('Content-Type')

Media type	Description
text/html	HTML document
text/plain	Plain text document
image/jpeg	JPG image
application/pdf	PDF document
application/json	JSON data
application/xhtml+xml	XHTML document

User agent

- Là chuỗi nhận dạng của trình duyệt web khi gửi yêu cầu đến máy chủ web
  - Nội dung user agent tùy thuộc vào trình duyệt
  - Máy chủ sẽ biết user dùng trình duyệt gì
- req = Request('http://www.python.org')
- urlopen(req)

- req.get_header('User-agent')

Cookie

- Cookie là một đoạn văn bản mà một Web server có thể lưu trên ổ cứng của người dùng
- Nằm trong header Set-Cookie khi server gửi response
- Cookie cho phép một website lưu các thông tin trên máy tính của người dùng và sau đó lấy lại nó
- Thông tin: tên – giá trị (name-value)

Cookie

- Cấu trúc cookie: 4KB
  - Name Value
  - Expires Path
  - Domain Secure
  - HttpOnly
- Tạo không gian lưu cookie mà server gửi đến
  - from http.cookiejar import CookieJar
  - cookie_jar = CookieJar()

Cookie

- tạo urllib builder, nó sẽ tự động lấy ra cookie từ response của server và lưu trong cookie_jar:
  - from urllib.request import build_opener, HTTPCookieProcessor
  - opener = build_opener(HTTPCookieProcessor(cookie_jar))
- sử dụng opener để tạo HTTP request
  - opener.open('http://www.github.com')
- Tạo danh sách để lấy từng cookie
  - cookies = list(cookie_jar)

Url

- Một số thành phần
  - URL scheme, thường là Tên giao thức
  - Tên miền
  - Chỉ định thêm cổng (có thể không cần)
  - Đường dẫn tuyệt đối trên máy phục vụ của tài nguyên
  - Các truy vấn (có thể không cần)
  - Chỉ định mục con (có thể không cần)
- Module urllib.parse – chia các thành phần url
  - from urllib.parse import urlparse
  - result = urlparse('http://www.python.org/dev/peps')

Url

- Query string
  - là tập hợp các dữ liệu ở dạng key=value mà ta đưa vào đằng sau URL của website
    - urlparse('https://docs.python.org/3/search.html?q=urlparse&area=default%20')
- Nén/mã hóa url
  - from urllib.parse import quote
  - quote('A duck?')

Các phương thức HTTP

- là cách client yêu cầu server phải làm gì với request của mình
  - Phương thức GET (đã làm)
  - POST
  - HEAD

- Packet requests