Java Web

1. OSI Model and TCP/IP Model
   1. OSI model – open system interconnection model (old one)
      1. 7 layers

7.Application Layer – human computer interaction layer where application can access the network services

* HTTP (Hypertext Transfer Protocol)
* FTP (File Transfer Protocol)
  + - * + POP (Post Office Protocol)
        + SMTP ()
        + DBS

6.Presentation Layer – ensures that data is in usable format and is where data encryption occurs

5.Session Layer – maintains connection and is for controlling ports and sessions

4.Transport Layer – transmits data using transmission protocol: TCP, UDP

3.Network Layer – decides which physical path the data will take

2.Data Link Layer – defines the format of data one the network

1.Physical Layer – transmits tow bit stream

* 1. TCP/IP models
     1. 对比图

1. HTTP (Hypertext Transfer Protocol)
   1. HTTP requests
      1. 图
      2. HTTP version type
      3. A URL
      4. HTTP method
         1. CREATE, PUT, UPDATE…
      5. HTTP request headers
      6. HTTP body (optional)
   2. HTTP response
      1. 图
      2. HTTP status code (details in 3)
         1. 1xx informational
         2. 2xx success
         3. 3xx redirection
         4. 4xx client error
         5. 5xx server error
      3. HTTP response header
      4. HTTP body (optional) – usually render some page, so usually a HTML file
2. HTTP Advanced
   1. HTTP request method
      1. Safe – a HTTP method doesn’t alter the state of the server
      2. Idempotent – if an identical request is made once or several times, the server will be the same state
      3. Cacheable – can be cache
         1. Private browser cache
         2. Shared proxy cache (ISP)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| method | safe | idempotent | cacheable | description |
| GET | yes | yes | yes | Read/retrieve data from the server |
| POST | no | no | no | Create some resources |
| PUT | no | yes | no | Update the whole object |
| DELETE | no | yes | no | Remove |
| PATCH | no | no | no | Partial update |

HEAD, OPTIONS, TRACE, CONNECT are also method but are seldom used.

* 1. HTTP status code
     1. 1xx informational
     2. 2xx success
        1. 200 OK, GEO/PUT/POST method
        2. 201 success and new resources has been created, POST method
        3. 202 accepted, request has been received for processing, but process hasn’t been completed yet
        4. 204 no content, PUT method, usually update the resources without changing the current page displayed to the user
     3. 3xx redirection
        1. 307 temporary redirect
        2. 308 permanent redirect
     4. 4xx client error
        1. 400 bad requests, the server could not understand the request due to invalid syntax
        2. 401 unauthorized -> unauthenticated, the client is not authenticated
        3. 403 forbidden -> the client has no permission
        4. 404 not found, the server cannot find the requested resources
     5. 5xx server error
        1. 500 internal server error
        2. 501 not implemented, request method is not supported by the server
        3. 502 bad gateways, the error response means that the server, while working as a gateway to get a response needed to handle the request, got an invalid response

1. TCP (Transmission Control Protocol) / UDP
   1. TCP Functionality
      1. Break application data into packets that networks can deliver
      2. Send packets to and accept packets from network layer
      3. Handle retransmission of dropped or garbled packets -> provide error free data transmission
      4. Acknowledge all packets that arrive
   2. TCP usage
      1. Secure shell (ssh)
      2. FTP (File Transfer Protocol)
      3. SMTP (Simple Mail Transfer Protocol)
      4. POP, IMAP
      5. HTTP
   3. UDP also in transport layer
      1. Possibility of the data loss
      2. Less delay
      3. Allows small transaction
      4. Supports bandwidth intensive application that tolerate packets loss
   4. UDP usage – tolerate packets loss
      1. Voice Over IP (VoIP)
      2. Streaming Video
      3. Gaming
   5. TCP 3-way handshaking
      1. 图
   6. TCP 4-way handshaking
2. Java Web Application
   1. Web application server
      1. Tomcat
      2. JBoss
      3. Jetty
      4. Apache TomEE
      5. Oracle Weblogic
      6. WildFly
      7. Apache Geronimo
      8. GlassFish
   2. Web application server flow

图

1. Basic flow of control
   1. 图
   2. Bin: startup, shutdown, and other scripts and executables
   3. Common: common classes that Catalina and web app can use
   4. Config: xml files (config files)
   5. Logs: app logs
   6. Temp: temporary variables and files
   7. Webapps: contains the web application you deployed
   8. Work: temporary storage
   9. Install tomcat on EC2 instance
      1. Sudo -i
      2. Yum update
      3. Yum install tomcat
      4. Systemctl start tomcat
      5. Systemctl status tomcat
      6. Systemctl stop tomcat
2. Servlet Life Cycle
   1. 图
   2. The servlet is initialized by calling init() method
   3. The servlet calls service() method to process a client’s request
   4. The servlet is terminated by calling the destroy() method
   5. Finally, servlet is garbage collected by the garbage collector of JVM
   6. init() – called only once
   7. service() – main method to perform the actual task
      1. will trigger doGet(), doPost(), doPut(), doDelete()
   8. destroy() – called only once at the end of the life cycle of a servlet
      1. usage: close database connection, stop background threads, write cookie lists, and other cleanup activities
3. Three Tier Architecture
   1. 图
4. Session and Cookies
   1. Session
      1. an object which provides the implementation of HttpSession interface
      2. on the server side
   2. Cookie
      1. Cookie is a small piece of information sent by server on client memory
      2. Client side
      3. Two types
         1. Session cookie: do not expire, in browser memory. As soon as the browser is closed, the cookie gets destroyed
         2. Persistent cookie: have expired time, hard disk
   3. 图

Hw可以在local docker virtual machine