Servlet

Servlet: java classes that can handle HTTP Requests and Responses from web clients.

Web server is a program that uses HTTP to serve the files from Web pages to users, in response to their requests.

Application servers is a software framework that provides facilities to create web applications and a server environment to run them.

Client-server interaction is a distributed application structure that partitions workloads between the servers, and clients.

Web container -- Tomcat:

* + Tomcat is “Thread-per-request”
  + Checks whether any thread is available to cater to an incoming request
  + Server can process many concurrent requests.
  + Tomcat’s thread pool is represented by the executor.

Servlet hierarchy

* + Servlet Interface ---> Generic Servlet ---> HttpServlet ---> MyServlet
  + The servlet interface is the root of the hierarchy.

Servlet lifecycle: init(), service(), destroy().

Inti called by the servlet container to indicate to a servlet that the servlet is being placed into service

Destroy is called by the servlet container to indicate to a servlet is being taken out of service.

How is a request processed with servlets:

* Client sends http request
* Server receives request.
* Servlet container consults web.xml – Deployment descriptor – to map the request to appropriate servlet.
* container instantiates the servlet.
* Container calls the init() method of the servlet – the first step in the servlet life cycle.
* Container calls the servlet’s service () method.
* HTTP response is returned by servlet to the server then to the client.
* Container calls the servlet’s destroy () method, releasing the instance to save memory. After the active period for the servlet has elapsed, container shut down.

Web.xml, deployment descriptor is the file with which we configure our web apps. We indicate which url-patterns will map requests to respective servlets.

Important tags: <servlet> and <servlet mapping>

ServletConfig: is an interface and is used to initialize a single servlet in a web app by the servlet container.

ServletContext: allows objects to be shared to any servlet in the app.

PrintWriter allows you to write responses

Load on startup is defined within the .js file. The method that is called within window.onload is the first HTML that the app will show when the server is initialized.

Class MyServlet extends HttpServlet

doGet: used to make an HTTP GET request

doPost: used to make an HTTP POST request.

doPut: for HTTP PUT request, allows a client to place a file on the server

doDelete: for HTTP Delete Requests, allows a client to remove a document from the server

.

ObjectMapper provides functionality for reading and writing JSON. ObjectMappers are thread-safe.

HttpServletRequest receives data sent by client.

HttpServletResponse sends data back to the client.

req.getRequestDispatcher(“login.html”).forward(req,resp);

resp.sendRedirect(“index.html”);

RequestDispatcher dines object that receives requests from the client and sends them to any resource.

Methods include: Forward: forwards to another source, Include: includes content in response

Forward:

* The request will be further processed on the server side.
* URL in a browser stays the same.
* Req & Resp objs will remain the same objects after forwarding.

Redirect:

* Request is redirected to a different resource
* Client will see URL change after redirect
* A new request is created
* Used within POST/REDIRECT/GET

In general, a forward should be used if the operation can be safely repeated upon a browser reload of the resulting web page, otherwise, redirect must be used. Typically, if the operation performs an edit on the datastore, then a redirect is required.

Session management

* Cookies: small piece of information persisted between multiple client request. Have names, values, optional attributes. STORED IN THE BROWSER CACHE.
* PROS: simplest form of session mgmt.
* CONS: text-only and cookies can be disabled

URL rewriting: appendID to url.

* PROS: works regardless of cookies being disabled, doesn’t require extra form submission.
* CONS: text-only and only works with links

The front controller design pattern means that all requests will be handled by a single handler and then dispatched to the appropriate handler for that type of request.

**Typescript**

TYPESCRIPT is a superset of JS, meaning that any valid JS code is also valid TS code. TS code must be TRANSPILED into JS code.

TS Data Types: strong typing

Number, boolean, string, void, null, undefined, never, any.

Decorators: Special kind of declaration that can be attached to a class declaration, method, accessor, property, or parameter.

Access modifiers: Public and Private, by default the members are public

Classes:

Similar to classes in most OOP languages

Properties are made public by default but can be made private

In classes, functions are called methods

Abstract Classes, interfaces same as java.

**ANGULAR**

* Npm: package manager for javascript
* Angular: is a TypeScript-based open-source front-end web application platform.
* CLI: Command-line interface, is a command line tool for creating angular apps.

Components: Ui building block of an angular app

Modules refers to a place where you can group the components, directives, pipes, and services, which are related to the application. To define module, we use the NgModule.

* Templates: HTML
* Data binding: Allows you to define communication between a component and the DOM.
* Directives: changes the appearance or behavior of a DOM element

1. Components: directives with a template

2. Structural directives: ngIf, ngFor, ngSwitch

3. Attribute directives: ngClass, ngStyle.

* Services: great way to share information among classes that don’t know each other.
* Dependency Injection
  + Increase efficiency and modularity
  + Dependency injection (DI) is a coding pattern in which a class asks for dependencies from external sources rather than creating them itself.
* Pipes: takes in data as input and transforms it to a desired output
* HttpClient offers a simplified client HTTP API for angular applications that rests on the XMLHTTPRequest interface. Additional benefits of HttpClient include testability features, typed request and response object, request and response interception, observable apis, and streamlined error handling.
* RxJS. Reactive Extensions for JavaScript is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code.

Routing enables navigation from one view to the next as users perform application tasks.

Design Patterns

() = “”: event binding

[] = “” property binding

[(ngModel)]: two way binding

Guard: strict on route

Cassandra

A NoSQL database is a database that provides a mechanism to store and retrieve data other than the tabular relations used in relational databases. These databases are schema-free, support easy replication, have simple API, eventually consistent, and can handle huge amounts of data.

Apache Cassandra is an open source, distributed and decentralized/distributed database, for managing very large amounts of structured data. It provides highly available service with no single point of failure.