Unit-4 Web Application Security

- Jigna Solanky

Introduction

- Security is an essential part of web applications and should be taken into consideration from the first stage of the development process.
- Security is all about protecting your assets from unauthorized actions.

Cont..

- Mechanisms that can be developed to secure your web application.
  - Identifying users
  - Granting or denying access to sensitive resources
  - Protect your data that stored on your server and transmitted over the wire

Cont..

- Creating a secure architecture and design requires that you have an in-depth understanding of your application's environment.
- You can't create secure application if you don't know who has access to your application and where possible points of attack might be.
- Most important factors for creating secure application are
  - Good understanding of environmental factors such as,
    - Users
    - Entry points
    - Potential possible threats with points of attack

Threat Modeling

- Threat Modeling is structured way of analysing your application's environment for possible threats, ranking those threats, and then deciding about mitigation techniques based on those threats.

Secure coding guidelines

- **Never trust user input**
- Strongly validate your user input
- Write your validation code in a way that it verifies input against only allowed values and not invalid values.

Cont..

- **Never use string concatenation for creating SQL statements**
- Always use parameterized statements so that your application is not SQL injectable.

Cont..

- **Never output data entered by a user directly on your web page before validating and encoding it:**
- The user might enter some HTML code fragments (for example, scripts) that lead to cross-site scripting vulnerabilities.
- Therefore, always use HttpUtility.HtmlEncode() for escaping special characters such as < or > before outputting them on the page, or use a web control that performs this encoding automatically.

Cont..

- **Never store sensitive data, business-critical data, or data that affects internal business rule decisions made by your application in hidden fields on your web page:**
  - Hidden fields can be changed easily by just viewing the source of the web page, modifying it, and saving it to a file.
  - Then an attacker simply needs to submit the locally saved, modified web page to the server.
  - Browser plugins are available to make this approach as easy as writing an e-mail with Microsoft Outlook.

Cont..

- **Never store sensitive data or business-critical data in view state:**
  - View state is just another hidden field on the page, and it can be decoded and viewed easily.
  - View state encryption helps to protect information that's only valuable for a limited interval of time, but keep in mind that even encrypted data can eventually be cracked if an attacker has enough time, resources, and motivation.

Cont..

- **Enable SSL when using Basic authentication or ASP.NET forms authentication:**
- **Protect your cookies:**
  - Always protect your authentication cookies when using forms authentication, and set timeouts as short as possible and only as long as necessary.

Cont..

- **Use SSL:**
  - In general, if your web application processes sensitive data, secure your whole website using SSL.
  - Don’t forget to protect even image directories or directories with other files not managed by the application directly through SSL.

If you forget about just one of these guidelines, all the other security features are more or less useless.

Levels of security

- **Authentication:**
  - First, Authenticate your users
  - Authentication is the process of discovering a user's identity and ensuring the authenticity of this identity.
  - Authentication is implemented through several ways
    - Windows authentication
    - Forms authentication
    - A custom authentication process

Cont..

- In all the ways of authentication, the user provides credentials when logging in.
- User's identity is tracked in different ways depending on the type of authentication.
- Ex, Windows authentication uses 96-bit number called SID (Security identifier) to identify each logged-in user.
- Forms authentication uses a ticket

Cont..

- **Authorization:**
  - Once the user is logged in, application has to decide which operations the user may execute and which resources the user may access.
  - Authorization is the process of determining the rights and restrictions assigned to an authenticated user.

Cont..

- **Confidentiality:**
  - While the user is working with the application, you have to ensure that nobody else is able to view sensitive data processed by the user.
  - Encrypt the channel between the client's browser and the web server.
  - Even, you have to encrypt the data stored on the backend.
  - **Confidentiality** means ensuring that data cannot be viewed by unauthorized users while being transmitted over a network or stored in a data store such as database.

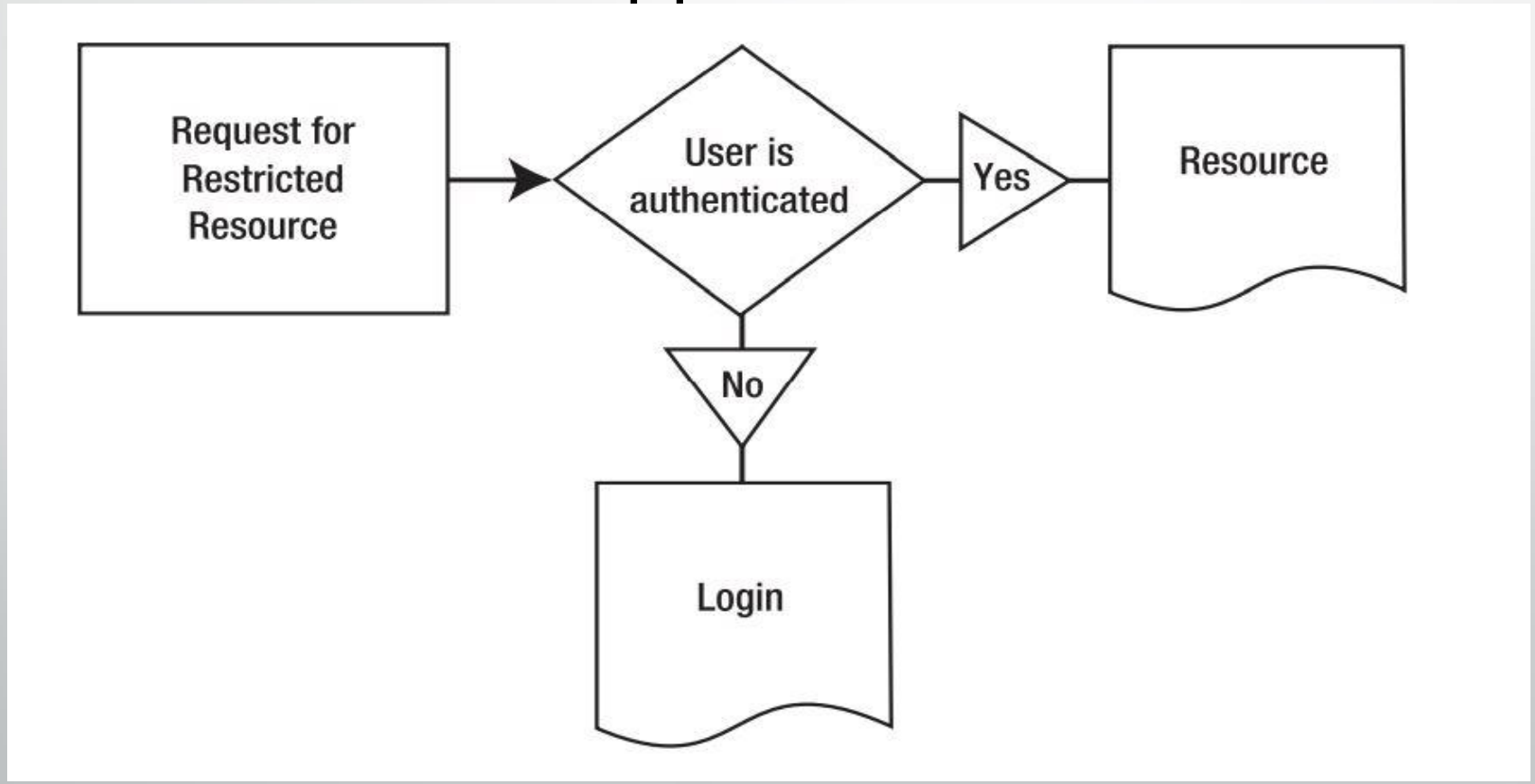
Cont..

- **Integrity:**
  - Make sure that data transmitted between the client's browser and the server is not changed by the unauthorized actors.
  - Digital signatured can be used to mitigate such requirements.
  - **Integrity** is all about ensuring that nobody can change the data while it is transmitted over a network or stored in a data store.

Cont..

- Both confidentiality and integrity is based on the encryption.
- **Encryption** is the process of scrambling data so that it's unreadable by other users.
- you might want to use encryption in a web application for two reasons:
  - To protect communication (data over the wire)
  - To protect permanent information (data in a database or in a file)

How do authentication, authorization, and impersonation all work together in a web application?



Cont..

- When users first come to your website, they are anonymous.
- By default, anonymous users can access any web page. But when a user requests a web page that doesn’t permit anonymous access, several steps take place:

Cont..

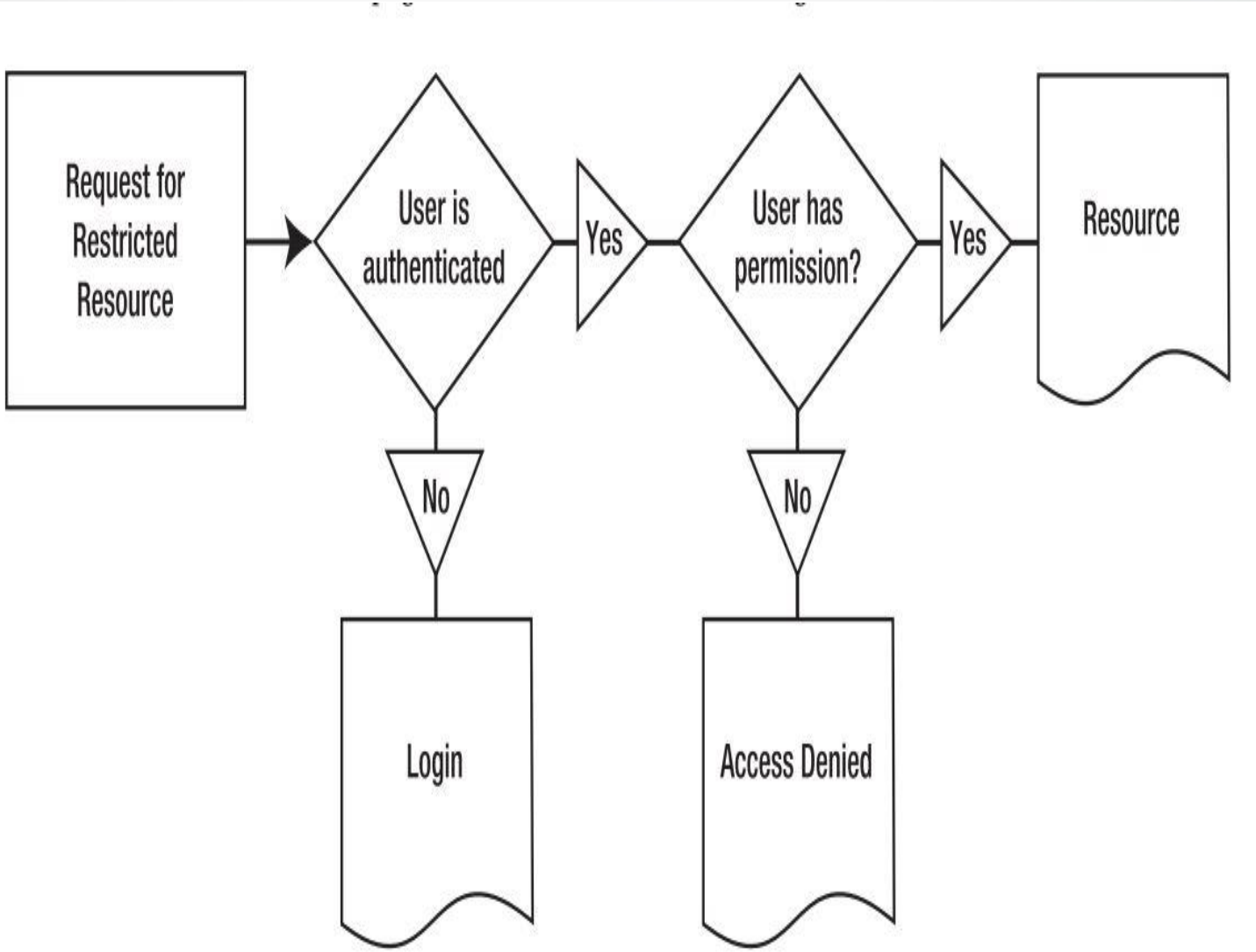
- The request is sent to the web server. Since the user identity is not known at this time, the user is asked to log in (using a custom web page or a browserbased login dialog box). The specific details of the login process depend on the type of authentication you're using.

- The user provides his or her credentials, which are then verified, either by your application (in the case of forms authentication) or automatically by IIS (in the case of Windows authentication)

- If the user credentials are legitimate, the user is granted access to the web page. If his or her credentials are not legitimate, then the user is prompted to log in again, or the user is redirected to a web page with an "access denied" message.

Cont..

When a user requests a secure web page that allows only specific users or users in specific roles,



Cont..

- The request is sent to the web server. Since the user identity is not known at this time, the user is asked to log in (using a custom web page or a browser based login dialog box). The specific details of the login process depend on the type of authentication you're using.
- The user provides his or her credentials, which are verified with the application. This is the authentication stage.
- The authenticated user's credentials or roles are compared to the list of allowed users or roles. If the user is in the list, then the user is granted access to the resource; otherwise, access is denied.
- Users who have access denied are either prompted to log in again, or they are redirected to a web page with an "access denied" message.

Secure Socket Layer (SSL)

- **Advantages of using HTTPS:**
- HTTP stands for Hyper Text Transfer Protocol where as HTTPS stands for Hyper Text Transfer Protocol Secure.
- HTTPS is more secure than HTTP.
- When the web server and client communicate, using HTTP protocol, the message that are exchanged over the internet are not encrypted. Any one can secretly listen and see the messages that are exchanged between the client and the web server.
- So, any sensitive information like passwords, financial transactions should never be done over HTTP protocol.

Cont..

- Most of the banking applications use HTTPS protocol.
- Message exchanged between the client and web server, using the HTTPS protocol are encrypted and are very secure.
- HTTP use port 80 and HTTPS use port 443.

Configure HTTPS instead of HTTP

- IIS is the web server for asp.net web applications.
- So, the configuration to use HTTPs, is usually done in IIS.
- The encryption and decryption of messages exchanged between the client and server is done by server certificates.
- These server certificates needs to be installed on the IIS Server.

How SSL is different from HTTPS

- HTTPS = HTTP + SSL
- SSL is a standard security technology for establishing an encrypted link between a web server and a web browser, so that the data sent over the internet can’t be read by others.
- SSL uses server certificates for encryption and decryption.
- Server certificated are issued by certificate authority (CA).

Cont.

- When a user requests a secure web page, the server generates an encryption key for the user's session and then encrypts the page's data before sending a response.
- On the client side, the browser uses the same encryption key to decrypt the requested web page and to encrypt new requests sent from that page.

Cont..

- An SSL certificate contains a public key and certificate issuer.
- Not only clients use the certificate to communicate with a server, clients can verify that the certificate was cryptographically signed by an official Certificate Authority.
- For ex, if my browser trusts the Verisign certificate authority, and verisign signs my SSL certificate, my browser will trust my SSL certificate.
- There are several trusted certificate authorities like
  - Verisign
  - Thawte
  - GeoTrust
  - Comodo
  - GoDaddy

Cont..

- The certificate authority acts as a clearing house to verify the server's identity over the internet.
- When a browser requests a page over https, the browser also requests the server certificate and checks it against a list of trusted sites provided by the certificate authority.
- If the server certificate dose not match one of the sites already authorized by the user, or if the server certificate does not match the web address for which it was registered, or if there are any other problems with the server certificate, a warning message is displayed.

Cont..

- Certificate Authority not only provides encryption and decryption for secure data transmission but also provides assurance to users that a website is authentic.
- It is also possible to generate our own server certificates, using a tool called makecert.exe. This tool comes with visual studio and can be used from visual studio command prompt.
- Certificates that are generated using the tool, can only be used for testing purpose.

Cont..

- The industry-standard certificate type, known as x.509v3, contains the following basic information.
  - The holder's name, organization, and address
  - The holder's public key, which will be used to negotiate an SSL session key for encrypting communication
  - The certificate's validation dates
  - The certificate's serial number

Performance when using HTTPS over HTTP

- Extra processing time is required for HTTPS, for key negotiation.
- Key negotiation is also termed as SSL handshake.
- The handshake allows the server to authenticate itself to the client

Windows Authentication

- Anonymous Authentication is fine for web sites that contain public information, that everyone can see.
- If the website contains private information or performs tasks such as ticket booking, placing orders etc, then the users need to be authenticated and authorised.
- Windows Authentication, identifies and authorizes users based on the server's user list.
- Access to resources on the server is then granted or denied based on the user account's privileges.

Cont..

- Windows Authentication is best suited for Intranet web applications.
- The advantages of windows authentication is that, web application can use the exact same security scheme that applies to your corporate network. User names, passwords, and permissions are the same for network resources and web applications.
- Security for an ASP.NET application can be configured at two places:
  - In IIS
  - In Application itself

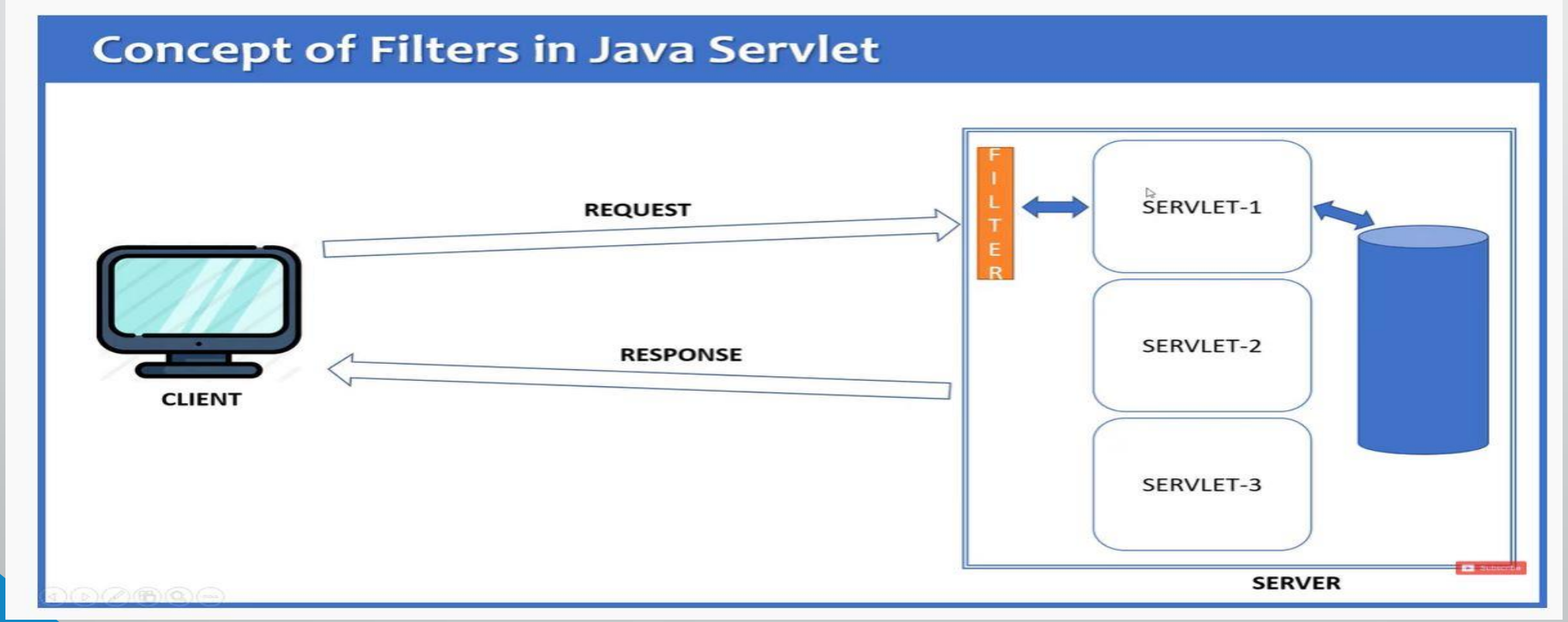
Cont..

- Windows authentication is used for intranet web applications, where the users are part of a windows domain-based network.

When to use Forms Authentication?

- Forms Authentication is used for internet web applications.
- The advantage of forms authentication is that users do not have to be a member of a domain-based network to have access to your application.
- Many internet web sites like gmail.com, amazon.com and facebook.com etc uses forms authentication.
- To access these applications we do not have to be member of their domain-based network.

RequestFilters in Java



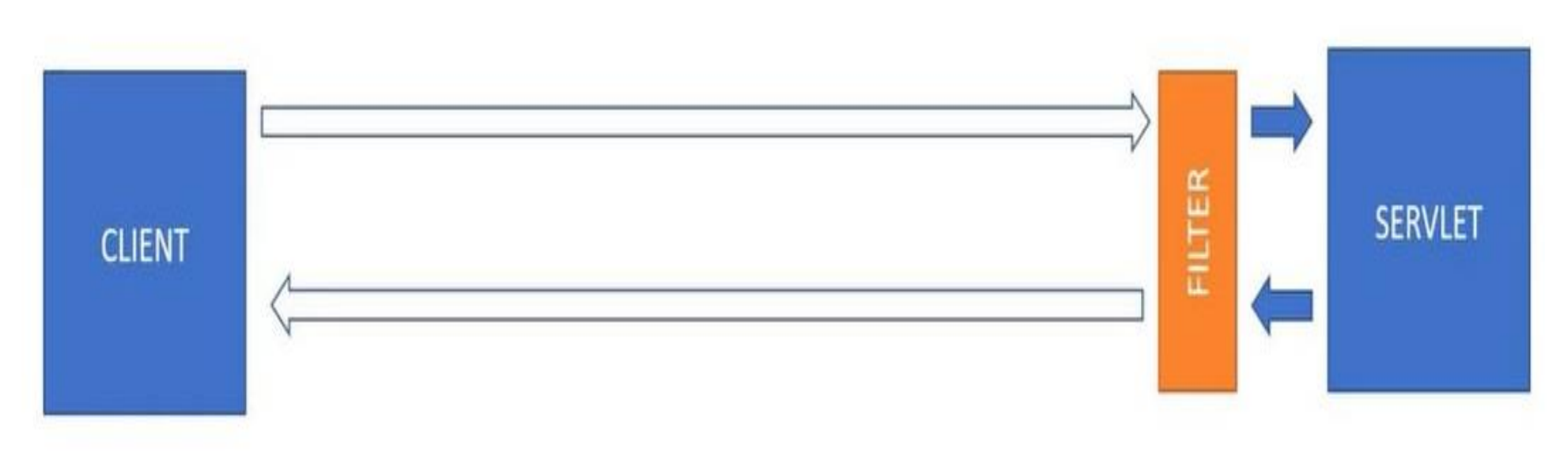
Filters

- A **filter** is an object that is invoked at the pre-processing and postprocessing of a request.
- **Before** and **After** servlet execution for filter data.
- The **servlet filter is pluggable**, i.e. its entry is defined in web.xml file, if we remove the entry of filter from the web.xml file, filter will be removed automatically and we don't need to change the servlet.

Why do we use filters?

- To validate the data coming from client because through the server the data will be stored in the database.
- So, only valid data should enter the database.

Usage of Filter



Cont..

- Authentication and authorization of request for resources.
- Formatting of request body or header before sending it to servlet.
- Compressing the response data sent to the client.
- Alter response by adding some cookies, header information etc.
- Input validations etc.

Filter API (javax.servlet)

- Filter (Interface)
  - Init()
  - doFilter()
  - Destroy()
- FilterChain (Interface)
  - Forward current request to some another resource
- FilterConfig
  - Can get or send the value from/to the web.xml while initializing

Introduction to AJAX

- Postback Operation: Action triggered by an end user or by code in a web page that sends data back to a web server for processing.
- Partial-page updates: Data in a section of a page is updated rather than the entire page

Cont..

- AJAX – Asynchronous JavaScript and XML
- XML – Extensible Markup Language
- JSON – JavaScript Object Notation
- DOM – Document Object Model
- REST – Representational State Transfer

What is AJAX?

- Ajax – Asynchronous JavaScript and XML
  - A technology for making asynchronous (Parallel) calls from a web page
  - Relies upon XMLHttpRequest object, JavaScript, CSS and JSON
  - Message sent back and forth between web pages and server using XMLHttpRequest
  - Results in minimal (or zero) page refreshes
  - Can integrate with web services or REST APIs
  - Works with all mainstream browsers

Cont..

- Ajax is cross platform technology which speeds up response time.
- The Ajax server controls add script to the page which is executed and processed by the browser.
- Like ASP.NET server control, AJAX server control can have methods and event handlers associated with them, which are processed on the server side.

ASP.NET AJAX Controls

- ScriptManager
- UpdatePanel
- Timer
- UpdateProgress
- ScriptManagerProxy

ScriptManager Control

- The ScriptManager control is the most important control and must be present on the page for other controls to work.
- The ScriptManager control takes care of the client-side script for all the server side controls.

Update Panel Control

- The UpdatePanel control is container control and derives from the control class.
- It acts as a container for the child controls within it and does not have its own interface.
- When a control inside it triggers a post back, the UpdatePanel intervenes to initiate the post asynchronously and update just that portion of the page.

Cont..

- For example, if a button control is inside the update panel and it is clicked, only the controls within the update panel will be affected.
- The controls on the other parts of the page will not be affected.
- This is called the partial post back or the asynchronous post back.

Timer Control

- The timer control is used to initiate the post back automatically. This could be done in two ways:
  - Setting the triggers property of the UpdatePanel Control
  - Placing a timer control directly inside the UpdatePanel to act as a child control trigger.
    - A single timer can be the trigger for multiple UpdatePanels

Use of timer control

- Periodically update the contents of one or more Update panel controls without refreshing the whole web page.
- Run code on the server every time that a timer control causes a post back.
- Synchronously post the whole web page to the web server at defined intervals.

AdRotator Control

- AdRotator control is used to display random Ads.
- The Ads information can be stored in an xml file or in a database table.

Cont..

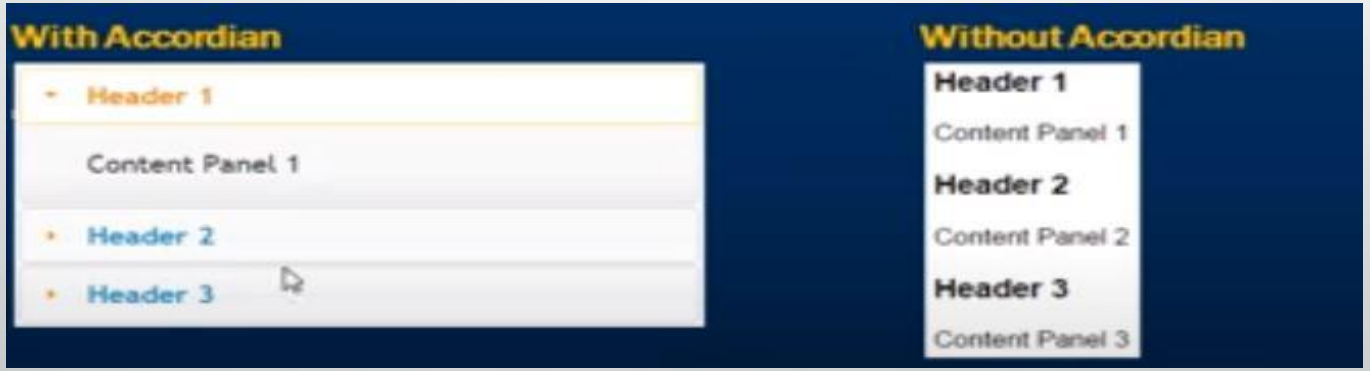
- XML file attributes
- ImageUrl – The URL of the image to display
- NavigateUrl – The URL to navigate to, when the ad is clicked
- AlternateText – The text to use if the image is missing
- Keyword – Used by the adrotator control to filter ads
- Impressions – A numeric value (a weighting number) that indicates the likelihood of how often the ad is displayed.

Cont..

- To open the target web page in a separate browser window, set Target="_blank"
- Use keyword attribute to filter ads
- The KeywordFilter and AdvertisementFile properties can be changed at runtime also.
- Changing the KeywordFilter at runtime could be very useful. For example, when the AdRotator control is on a master page, and if you want to change the keywordFilter on each content page.

jQuery Accordion

- Accordion is great for displaying collapsible content panels for presenting information in a limited amount of space.



Web Services

- A web service is a web-based functionality accessed using the protocols of the web to be used by the web applications.
- Web service is a piece of software that satisfies 3 key requirements:
  - Design in such a way that, **other applications can interact with it**. (Some form of data exchange happening between web service and that application)
  - **Interoperable** (Talk to applications across different platforms)
  - The **communication** between this web service and other applications should be facilitated **over the network**.

Cont..

- The message from an application to a web service is called a **Request**, and the message back to the application is called a **Response**.
- To ensure the webservice is interoperable, the request and response messages have to be specified in a universal format.
  - XML (Extensible Markup Language)
  - JSON (JavaScript Object Notation)

Web service types

- Two types of web services:
  - SOAP (Simple Object Access Protocol)
  - REST (Representational State Transfer)

SOAP

- In SOAP, the request and response exchange format is XML.
- The service definition in SOAP is known as WSDL (Web service Definition language).
- WSDL defines the end points, all operations are allowed through it, as well as the request and response structure.
- The WSDL acts as a contract between the application and the web service.

REST

- RESTful web service considers any data that gets exposed to an external application as a **resource**.
- Each resource is assigned a URI (Uniform Resource Locator) that can be used to obtain data back by an application.
- A RESTFul web services conforms to the HTTP methods such as GET, POST, PUT and DELETE and uses the HTTP status code.

Deploy ASP.NET Website to IIS

- Step – 1 : Open IIS (Internet Information System)
- Step -2 : Right click on sites directory in ISS and Click on Add Website
- Step – 3: Give the sitename
- Step – 4 : Browse for the physical path (Create one directory Sites at anywhere and inside sites directory create one more directory with name sitename)
  - Newly created directory must be selected as physical path

Cont..

- Step – 5 : Change the port number (Not Mandatory)
- Step – 6 : Click on “OK”
- Step – 7 : Right click on Sitename, select manage web site and click on Browse (It will not open any page)
- Step – 8 : Go to the location of physical path directory and create one html file.
- Step – 9 : Repeat the step -7.

Cont..

- Step – 10 : Open the visual studio in administrator mode.
- Step – 11 : Right click on project and click on publish.
- Step – 12 : Publish new with option folder and click on next
- Step – 13 : Browse for the folder location that you have created in IIS. Click on Finish
- Step – 14 : In publishing window, click on edit button of “Delete Existing files”, so it will open one window.

Cont..

- Step – 15 : Explore the file publish options and check “Delete all existing files prior to publish” and click on Save
- Step – 16: Click on publish button
- Step – 17: Open ISS – right click on sitename – Manage Website – Browse
- Step – 18: Database connectivity -> Open Application pool -> Select Application pool (SiteName) -> Right Click -> Advance Setting
- Step – 19: Change the Identity Property to “LocalSystem”

Cont..

- Step – 20 : Open SQL Server manager -> click on login
- Step – 21: Right click on “NT AUTHORITY\SYSTEM” and it will open one window
- Step – 22: Select the user mapping and choose database and select role as “db owner” and click on OK.

REST Web Services (JAX-RS)

- JAX-RS is an API for RESTful web services.
- JAX-RS contains interfaces, therefore, to build an App we need actual implementation of them.
- There are many implementation libraries of the API like..
  - Rest Jersey
  - Restlet
  - RESTEasy

Jersey RESTful web service framework

- It is open source, production quality, framework for developing RESTful web services in java that provides support for JAX-RS APIs.
- Jersey provides it's own API that extend the JAX-RS toolkit with additional features and utilities to further simplify RESTful service and client development.
- Jersey is mainly distributed via Maven.
- Currently latest version is 3.0.2 (Version 3.x requires Tomcat v10)
- For Tomcat V9, downgrade jersey version to 2.x (eg. 2.34)

Cont..

- Modify web.xml file to map the servlet that is provided by jersey package with corresponding URL pattern.
- <servlet-mapping>

  <servlet-name>Jersey Web Application </servlet-name>

  <url-pattern>/webapi/* </url-pattern>

  </servlet-mapping>

cont..

- **@Path** – annotation's value is a relative URI path
- **@GET, @PUT, @POST, @DELETE, @HEAD** – resource method designator annotations defined by JAX-RS and which corresponds to the HTTP method.
- **@Produces** – annotation is used to specify the MIME media types of representations a resource can produce and send back to the client (text/plain, application/xml, application/json)

Cont..

```
@Path("/hello")
public class PublicationResource {
        @GET
       @Produces("text/plain")
    // @Produces({"application/xml", "application/json"}) //
@Produces(value={MediaType.APPLICATION_JSON, MediaType.TEXT_XML})
public String HelloWorld(){
        return “Hello World…";
}
}
```

Cont..

- /publications/{publicationId}/
  - To get nested URI (nested path) use the same **@Path** annotation for a class method.
  - Since we can not hardcode the path of a resource accessed by Id, we use **{variablename}** instead of concrete part of the URL.
  - To get an access to the variable use **@PathParam("variable name")** annotation for method's argument.

Cont..

- Example

```
@Path("/publications")
public class PublicationResource {
      @GET @Path("/{publicationId}")
      @Produces(MediaType.APPLICATION_JSON)
      public Publication getPublication(@PathParam(“publicationId") long id)
      {
            Publication publication = publicationService.getPublication(id);
             return publication;

      }
}
```

Cont..

- the variable in @Path annotation may be customized by specifying a different regular expression after the variable name (default regular expression is [^/]+?).
- For example, if a user name must begin with one uppercase or lowercase letter and zero or more alphanumeric characters and the underscore character. If a user name does not match that template, a 404 (Not Found) response will be sent to the client
  - **@Path("users/{username: [a-zA-Z][a-zA-Z_0-9]*}"**

Cont..

- Use **@POST** annotation to identify method that consumes HTTP POST request and creates new resource based on resource sent with the request body.
- **@Consumes** annotation is used to specify the MIME media types of representations a resource can consume from the client (text/plain, application/xml, application/json)

Cont..

- **Example**

```
@POST
@Consumes(MediaType.APPLICATION_JSON)
public Publication addPublication (Publication publication){
    return publicationService.addPublication(publication);
}
```

Cont..

- Use **@PUT** annotation to identify method that consumes HTTP PUT request and updates specified resource with resource sent within the request body.
- Use **@DELETE** annotation to identify method that consumes HTTP DELETE request and deletes specified resource.

Cont..

- **Example**

```
@PUT
@Path("/{publicationId}")
@Consumes(MediaType.APPLICATION_JSON)
public Publication updatePublication(@PathParam("publicationId") long id,
Publication publication)
{
    publication.setId(id);
    return publicationService.updatePublication(publication);
}
```

Cont..

```
@DELETE
@Path("/{publicationId}")
public void deletePublication (@PathParam("publicationId") long id)
{
      publicationService.removePublication (id);
}
```

Cont..

- **Filtering and Pagination** require use of **query parameters** of the request.
  - /publications?year=2018
  - /publications?start=5&size=10
- Use **@QueryParam("Parameter name") annotation** for method's argument to get a value of the parameter.

Cont..

- Example

```
@GET
public List getPublications(@QueryParam(“year”) int year,
@QueryParam(“start”) int start, @QueryParam(“size”) int size)
{
}
```

Param Annotations

- **@MatrixParam("parameter name")** annotation is similar to @QueryParam and used for the cases when parameters are separated by (;) in the request.
  - /publications;year=2016;size=5

Cont..

- **@HeaderParam("parameter name")** annotation is used to access an extra metadata in a form of custom header values of the request.
- **@CookieParam("parameter name")** annotation to access values of the cookie's names.
- **@FormParam("parameter name")** annotation to access "key:value" pairs in HTML Form submissions.

Cont..

- In the mentioned cases you suppose to know **parameter names** in advance. If you do not have such opportunity, you are able to get them and other useful metadata from the **Context** of the request using **@Context** annotation and corresponding components:
  - **UriInfo** provides both static and dynamic, per-request information, about the components of a request URI (e.g. absolute path .getAbsolutePath(), base URI .getBaseUri(), query parameters .getQueryParameters(), etc.).
  - **HttpHeaders** provides access to request header information either in map form or via strongly typed convenience methods. (e.g. names of all the headers .getRequestHeaders(), cookies .getCookies(), date .getDate(), accepted Media Types .getAcceptedMediaTypes(), etc.).

Cont..

```
public String getParamsUsingContext(@Context UriInfo uriInfo, @Context
HttpHeaders headers)
{
        String path = uriInfo.getAbsolutePath().toString();
        String cookies = headers.getCookies().toString();
        return "Path: "+path+"; Cookies - "+cookies;
 }
```

Hints

- REST follows one-to-one mapping between create, read, update, and delete (CRUD) operations and HTTP methods.
  - To create a resource on the server, use POST
  - To retrieve a resource, use GET
  - To change the state of a resource or to update it, use PUT
  - To remove or delete a resource, use DELETE

Steps to create restful web service

- Create dynamic project
- Convert project to maven project (Right click on project, configure -> Convert to maven project)
- Open pom.xml
- Add following dependency
  - `org.glassfish.jersey.containers (2.35)`
  - `org.glassfish.jersey.inject(2.35)`
  - `org.glassfish.jersey.media (2.35)`

Cont..

- Open web.xml
-

```
<servlet>
    <servlet-name>Jersey REST Service</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
        <param-name>jersey.config.server.provider.packages</param-name>
        <param-value>com.rest.web.ws</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>Jersey REST Service</servlet-name>
    <url-pattern>/rest/*</url-pattern>
</servlet-mapping>
```