Template Esame

▼ Invio di parametri in formato XML da una servlet a una JSP, che li recupera con Js

```
//nella JSP

function displayDataFromXML(xmlString) {
    // Effettua il parsing della stringa XML
    const parser = new DOMParser();
    const xmlDoc = parser.parseFromString(xmlString, "text/xm

    // Recupera i dati dalla stringa XML
    const string1 = xmlDoc.getElementsByTagName("string1")[0]
    const string2 = xmlDoc.getElementsByTagName("string2")[0]

    // Assegna i valori ai tuoi elementi HTML
    document.getElementById(".").value = string1;
    document.getElementById(".").value = string2;
}

// Recupera il parametro XML dalla query string dell'URL
const urlParams = new URLSearchParams(window.location.search)
const xmlString = urlParams.get('XML');

// Chiama la funzione passando la stringa XML
displayDataFromXML(xmlString);
```

```
// nella servelt:

response.setContentType("text/xml;charset=UTF-8");
String XML = "<response><string1>" + string2 + "</string1><st
response.sendRedirect("page.jsp?XML="+XML);
```

▼ Controllare se una stringa è alfanumerica / numerica e, alla pressione di un determinato tasto inviarla in automatico a una servelt/jsp

```
function checkChar(string) {
    // Verifica se il carattere  alfanumerico usando una re
    var chaRegex = /^[a-zA-Z ]+$/;     ///^[a-zA-Z0-9 ]+$/ se
    return charRegex.test(string);
}

function handleKeyPress(e) {
    var key = e.keyCode;
 //   console.log("key +", key);         //cosi da console ve
    if (key === "") {

        var text = document.getElementById("testo").value;

        if( checkChar(text)){
            document.getElementById("form").submit();
            document.getElementById("testo").value = "";


        }else {
            alert("Errore carattere NON alfabetico");
        document.getElementById("testo").value = "";
        }
            e.preventDefault();              //evita la propa

    }
}
```

▼ Admin che tiene il conto delle sessioni degli utenti attive negli ultimi x giorni, e il numero di operazioni effettuate: (Es1 11-01-2022)

```
//nella servlet che gestisce le richieste del client
```

```
private Map<HttpSession, LocalDateTime> state;
private Map<HttpSession, Integer> sess;
    //in init
      state = new HashMap<>();
        sess = new HashMap<>();
        this.getServletContext().setAttribute("state", state
        this.getServletContext().setAttribute("sess", sess);

    //nel servizio
    Map<HttpSession, LocalDateTime> state = (Map<HttpSession,
        Map<HttpSession, Integer> sess = (Map<HttpSession, In
        boolean b = state.containsKey(request.getSession());
        if(b == true) {
            state.replace(request.getSession(), LocalDateTime
            int x = sess.get(request.getSession());
            sess.replace(request.getSession(), ++x);
        }
        else  {
            state.put(request.getSession(), LocalDateTime.now
            sess.put(request.getSession(), 0);
        }


        this.getServletContext().setAttribute("sess", sess);
        this.getServletContext().setAttribute("state", state
```

```
//nella jsp dell' admin

<% Integer num = (Integer)this.getServletContext().getAttribu
System.out.println("en "+ num);         //check da terminale

if (num == 1) {
    try{
    List<HttpSession> activeSessions;
    LocalDateTime now = LocalDateTime.now();
```

```
    Map<HttpSession, LocalDateTime> state = (Map<HttpSession,
    activeSessions = state.entrySet().stream().filter(e -> Du
    Map<HttpSession, Integer> sess = (Map<HttpSession, Intege
    Map<HttpSession, Integer> newMap = sess.entrySet().strean

    for (Map.Entry<HttpSession, Integer> entry : newMap.entry
%>
      <div>  <%= entry.getKey().getId() %>: <%= entry.getValu
<%
    }
   }catch(Exception e){
        System.out.println("Exception");
        e.printStackTrace();
    }
%>

<%
}else
%> <div> Errore enable</div>
```

```
//nella servlet di login dell'admin

private int enable = 0;

        //in init
        this.getServletContext().setAttribute("enable", enabl

        //in service
    String admin = "admin.jsp";
        String login = "indexLoginUser.jsp";


        if(userName.equals("admin") && pwd.equals("admin")) {
            int e = 1;
             this.getServletContext().setAttribute("enable",
```

```
            login = admin;              //così facendo discri
        }

        response.sendRedirect(login);
```

▼ Invio e ricezione dati JSON da una JSP a una servlet

```
//nella JSP
Gson gson = new Gson();
    String testo = null;

    if ((testo = (String)request.getParameter("testo")) != nu
            String testoJson = (String)gson.toJson(testo);
            testo = null;
            response.sendRedirect("nomeServlet?testoJson="+te

        }

//nella servlet
Gson gson = new Gson();
String testoJson = (String)request.getParameter("testoJson");
String testo = gson.fromJson(testoJson);
```

▼ Abilitazione bottone dopo la scrittura di tot caratteri, altrimenti alert e pulizia casella di input

```
function buttonOn(e) {

        var testo = document.getElementById("testo").value;
        var dim = testo.length;
        if (dim >= 10) {
            if(checkChar(testo)){
                document.getElementById("submit").disabled =
            } else {
```

```
                alert("carattere NON alfa-numerico");
                document.getElementById("testo").value = ""
            }
        }
}
```

▼ Carattere speciale altra opzione

```
function carattereSpeciale(event) {
    var key = event.key;
    if (key === "$"){

    //fai qualosa
    };
}
```

▼ Invio richiesta AJAX

```
//parte client
function inviaRichiesta("nomeServlet", "datoInput", "IDdatoOu
    var xhr = new XMLHttpRequest();
    xhr.open('POST', "nomeServlet", true);
    xhr.setRequestHeader('Content-Type', 'application/x-www-f

    xhr.onreadystatechange = function () {
        if (xhr.readyState == 4 && xhr.status == 200) {
            document.getElementById("IDdatoOutput").innerHTMl
        }
    };

    xhr.send('stringa=' + encodeURIComponent("datoInput"));
}


//nella servlet
```

```
PrintWriter out = response.getWriter();         //per andare
        for(String s : " ") {
            out.println(s);
        }
        out.close();
*/
```

▼ Concedere/ togliere i permessi di scrittura su una textArea tramite una bottone

```
function noPermission() {
    // Disabilita la textarea
    document.getElementById("IDtextarea").disabled = true;

    // Puoi anche nascondere il pulsante se necessario
    document.getElementById("noPermissionButton").style.displ
}
```

▼ Estrazione numeri casuali in React

```
const [estratti, setEstratti] = useState([]);
const estraiNumeri = () => {
    // Genera 5 numeri casuali ammissibili
    const numeriEstratti = [];
    while (numeriEstratti.length < 5) {
        const numeroCasuale = Math.floor(Math.random() * 10)
        if (!numeriEstratti.includes(numeroCasuale)) {
            numeriEstratti.push(numeroCasuale);
        }
    }
    setEstratti(numeriEstratti);
};
```

▼ Creare N matrici jsp

```
<div>
<% for (int m = 0; m < nMatrici; m++){ %>    //creazione
    <table>
        <% for (int row = 0; row < 3; row++){ %>
            <tr>
            <% for (int col = 0; col < 3; col++){ %>
                <td>
                    <input style="height:30pt; width:30pt" ty
                </td>
            <%} %>
            </tr>
    <% }%>
    </table>
    <br></br>
    <br></br>
<% }%>
<button onclick="invia(<%= nMatrici %>)">Invia!</button>
<script> //controllo con JS
    for (let m = 0; m < <%=nMatrici%>; m++){
        for(let row = 0; row < 3; row++){
            for (let col =0; col < 3; col++){
                document.getElementById("cell-"+m+"-"+row+"-'

                    if (isNaN(parseFloat(this.value))){
                        alert("Inserire un valore numerico n
                    }
                })
            }
        }
    }
</script>

</div>
```

```
//funzione invia

function invia(nMatrici){
    var matrici = []
    for (let m = 0; m < nMatrici; m++){
        let matrice = {}
        for(let row = 0; row < 3; row++){
            for (let col =0; col < 3; col++){
                let cell = document.getElementById("cell-"+m-
                if (cell.value == ""){
                    alert("Riempire tutte le celle");
                }
                matrice[""+row + "-" + col]= parseFloat(cell
            }
        }
        matrici.push(matrice);
    }
    console.log(matrici);
    var results = [];

    matrici.map((matrice, idx) => {
        let timeInit = performance.now();
        const queryString = new URLSearchParams(matrice).toSt
        fetch(`Determinante?${queryString}`, {
            method: 'GET'})
        .then(response => {
```

```
            return response.text();
        })
        .then(data => {
            results.push({
                time: performance.now()-timeInit,
                index: idx,
                det: data
            });
            console.log("Tempo: " + (performance.now()-timeIn
        })
        .catch(error => {
            console.log("Error");
        });
    })
    console.log(results);
}



//esempio di ricezione servlet col calcolo del DET

public class Determinante extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private static float get_cell(HttpServletRequest request,
        return Float.parseFloat(request.getParameter(""+row+"
    }

    protected void doGet(HttpServletRequest request, HttpServ
        float det = 0;
        det += this.get_cell(request, 0, 0)*this.get_cell(re
        det += this.get_cell(request, 1, 0)*this.get_cell(re
        det += this.get_cell(request, 2, 0)*this.get_cell(re

        det -= this.get_cell(request, 0, 2)*this.get_cell(re
        det -= this.get_cell(request, 1, 2)*this.get_cell(re
```

```
        det -= this.get_cell(request, 1, 0)*this.get_cell(re

        response.getWriter().print(det);
    }
}
```

```
//la fetch di prima si può modificare con :

var results = [];
var completedRequests = 0;

matrici.forEach((matrice, idx) => {
    let timeInit = performance.now();
    const queryString = new URLSearchParams(matrice).toStrin

    const xhr = new XMLHttpRequest();
    xhr.open('GET', `Determinante?${queryString}`, true);

    xhr.onreadystatechange = function () {
        if (xhr.readyState == 4) {
            if (xhr.status == 200) {
                const data = xhr.responseText;
                results.push({
                    time: performance.now() - timeInit,
```

```
                    index: idx,
                    det: data
                });
                console.log("Tempo: " + (performance.now() -
            } else {
                console.log("Error");
            }

            completedRequests++;

            if (completedRequests === matrici.length) {
                // All requests are completed, log the result
                console.log(results);
            }
        }
    };

    xhr.send();
});
```

▼ Esami-React

1. 11-01-2022: Campo minato. Configurazione di dim e step, realizzazione di una matrice dim*dim, e l'utente che deve selezionare step caselle (evitando due mine per riga, casuali) per vincere.

2. 09-06-2022: Schedina. Inserimento di 5 numeri ed estrazione casuali di altrettanti. Verifica di ambo/terna/quaterna/cinquina.
3. 07-09-2022: Caccia al Tesoro. Configurazione di length e width, creazione di una matrice length*width e definizione casuale di una cella T. L’utente deve trovare la T, e in base al numero di tentativi impiegati riceve un punteggio.

▼