

## Domanda #3 - Javascript

Si sta progettando un sito di commenti che permette di aggiungere brevi testi ad articoli identificati da un id. Basandosi, dove si ritiene, su uno o più framework Javascript a piacere tra quelli illustrati a lezione, si descrivano:

Si ipotizzi poi a vostra discrezione un'API REST con due servizi:

- `/showComments`, restituisce un elenco di commenti per questo articolo.
- `/addComment`, che permette di aggiungere un commento (mai anonimo) all'articolo *oppure* ad un altro commento dell'articolo.

Potete usare i servizi `http://site202100.tw.cs.unibo.it/info` oppure `http://www.fabioitali.it/TW/test/2021/doResponse.php` per effettuare prove e debugging.

**Parte I** Il codice HTML per visualizzare che contiene la struttura di una pagina e una testata del sito con un menu a piacere (questa struttura non cambia mai), il testo dell'articolo (che dipende da un id), i commenti fin qui ricevuti sull'articolo visualizzato, e gli elementi di un form per

- aggiungere un nuovo commento all'articolo, e
- rispondere ad un altro commento già esistente per questo articolo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>untitled</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta name="generator" content="Geany 1.29" />
    <script src="https://www.fabioitali.it/TW/lib/jquery-3.6.0.js"></script>
</head>

<body>
    <nav>
        <a href="#"> Home </a>
        <a href="#"> Top Articles </a>
    </nav>
    <main>
        <div id="article">
            </div>
            <div id="commentsSection">
                <h3>Posted by Andrea</h3>
                <p> Nice pic </p>
                <input type="text" placeholder="Nome" name="autore"/>
                <input type="text" placeholder="Rispondi" name="testo"/>
            </div>
        </main>
    </body>
</html>
```

```

    <div id="1">
      <h5>Marco Aurelio</h5>
      <p>Risposta al commento di andrea</p>
      <input type="text" placeholder="Nome" name="autore"/>
      <input type="text" placeholder="Rispondi" name="testo"/>
    </div>
  </div>
  <div id="errorSection"></div>
</main>

```

```
</body>
```

```
</html>
```

**Parte II** Uno o più script Javascript che, appena prima di visualizzare la pagina dell'articolo, richiamano il servizio `/showComments` (con i parametri correttamente inizializzati) e popolano correttamente i commenti in fondo all'articolo in apposita area della pagina. Fornire tutti i meccanismi di chiamata, modificare la pagina quando i dati vengono ricevuti e gestire eventuali errori.

```

$.ready(=>{
  let baseUrl = "http://www.fabioitali.it/TW/test/2021/doResponse.php";
  let id = 1;
  //Assumo di avere una API "/loadArticle" che dato un id mi restituisce il t
  $.get(baseUrl + `/loadArticle/${id}`)
    .then( res => JSON.parse(res))
    .then( res => $("#article").html(res))
    .catch(err => console.log(err));

  $.get(baseUrl + `/showComments/${id}`)
    .then( res => {
      res = JSON.parse(res);
      let commSec = $("#commentSection");
      commSec.html();
      let html = "";
      //esempio di commento {id: 2, autore: "...", testo: "...", risposte : [
      //showComments restituisce un array di commenti relativi al posto con id
      for(let comment in res){
        html +=`<div id=${comment[id]}>
          <h3>Posted by ${comment[autore]}</h3>
          <p>${comment[testo]}</p>
          <input type="text" placeholder="Rispondi"/>
          <button onclick=`rispondi(${comment[id]}`>Rispondi al commentor
        for(let respost in comment[risposte]){

```

```

        html += `<div id=${respost[id]}>
                <h5>${respost[autore]}</h5>
                <p>${respost[testo]}</p>
                <input type="text" placeholder="Rispondi"/>
                <button onclick=`rispondi(${respost[id]})`>Rispondi al s
            }
        html += "</div>";
        commSec.append(html);
    }
})
.catch (err => console.log(err));
})

```

**Parte III** Uno o più script Javascript che vengano attivati cliccando sul pulsante di spedizione del nuovo commento, e che invochino il servizio `/addComment` (con gli eventuali parametri correttamente inizializzati) sulla base degli input forniti dall'utente nel form. Si fornisca tutto il meccanismo di invocazione, di gestione della risposta sia in caso positivo sia in caso di eventuali errori, in tutti i tipi possibili di errore. Nel caso di errore, si cerchi una soluzione più soddisfacente che mostrare un alert di errore. Si noti che non sono possibili commenti anonimi.

```

function rispondi(id){
    let container = $("#id");
    let text = $('#id>input[name="testo"]').html();
    let autore = $('#id>input[name="autore"]').html();
    if(text && autore){
        let commentToAdd = {
            autore: autore,
            testo: text
        };
        $.post(baseUrl + `/addComment/${id}`, commentToAdd)
        .then( res => {
            //il servizio POST mi restituisce il commento passato nella forma {
            let comment = JSON.parse(res);
            let commSec = $("#commentSection");
            let html = "";
            html +=`<div id=${comment[id]}>
                    <h3>Posted by ${comment[autore]}</h3>
                    <p>${comment[testo]}</p>
                    <input type="text" placeholder="Nome" name="autore"/>
                    <input type="text" placeholder="Rispondi" name="testo"/>
                    <button onclick=`rispondi(${comment[id]})`>Rispondi al commento
                for(let respost in comment[risposte]){
                    html += `<div id=${respost[id]}>
                            <h5>${respost[autore]}</h5>

```

```

        <p>${respost[testo]}</p>
        <input type="text" placeholder="Nome" name="autore"/>
        <input type="text" placeholder="Rispondi" name="testo"/>
        <button onclick=`rispondi(${respost[id]})`>Rispondi al s
    }
    html += "</div>";
    container.append(html);
  })
  .catch(err => $("#errorSection").html(err));
}
}

```