

Tuntitehtävä 3:

Tee alla olevan kuvan mukainen HTML sivu, joka vastaanottaa GET viestin Servlet:stä.

Käytä tässä html:n table elementissä olevaa painonappulaac, joka käynnistää haeTiedot() JavaScript funktion:

```
<table id="Listaus">
  <thead>
    <tr>
      <input type="button" id="hae" value="Hae" onclick="haeTiedot()">
    </tr>
  </thead>
  <tbody id="tbody">
  </tbody>
</table>
```

JavaScript funktio on puolestaan seuraavanlainen:

```
function haeTiedot(){
    fetch("laheta/",{
        method: 'GET'
    })
    .then( function (response) {
        return response.json()
    })
    .then( function (responseJson) {
        var viesti = responseJson.viesti;
        console.log(viesti);
        var htmlStr=viesti;

        document.getElementById("tbody").innerHTML = htmlStr;
    })
}
```

Tämä haeTiedot() funktio käynnistää fetch() funktion, jonka parametrit ovat seuraavanlaiset:

```
fetch(käynnitettäväServlet, käynnitettäväMetodiServletissä)
    .then(mitäTehdäänSitten) .then (mitäTehdäänSenJälkeen)
```

Servletissä puolestaan lähetetään doGet metodissa JavaScript tietorakenne takaisin sitä pyytäneelle HTML sivulle. Tämä JavaScript tietorakenne on:

```
{"viesti": "Tervehdys!"}
```

Servlet:iin tuleva ohjelma on siis seuraavanlainen:

```
String strJSON="";
String viesti = "{\"viesti\": \"Tervehdys!\"}";
strJSON = new JSONObject().put("viesti", viesti).toString();
response.setContentType("application/json");
response.setCharacterEncoding("UTF-8");
PrintWriter out = response.getWriter();
out.println(strJSON);
```

Tässä ohjelmassa talletetaan `{"viesti": "Tervehdys!"}` JavaScript tietorakenne Java muuttujaan. Sen jälkeen se muutetaan JSON tietorakenteeksi, jonka pystyy lähettämään Internetin yli sitä pyytäneelle HTML sivulle. Tämä tehdään `JSONObject().put("parametrinimi", javamuuttuja).toString()` lauseella. `JSONObject()` puolestaan löytyy `json-20180813.jar` Java kirjastosta, jonka joudut kopioimaan kurssin Moodle sivuilta Java projektisi `WebContent/WEB-INF/lib` kansioon. Sinne voi lisätä mitä tahansa Java kirjastoja, joita tarvitset kun sovellutustasi ajetaan Tomcat web-palvelimella. Tämän jälkeen kerrotaan `response.setContentType("application/json");` lauseella, että tietorakenne on JSON. Seuraavassa lauseessa puolestaan kerrotaan, että käytetään kansainvälistä utf-8 merkistöä, josta löytyvät esimerkiksi Suomen kielessä käytetyt ääkköset ja öökköset. Sitten luodaankin tarvittava olio, jonka avulla voidaan lähettää `{"viesti": "Tervehdys!"}` tietorakenne yli Internetin sitä pyytäneelle HTML sivulle. Tämä tehdään lauseilla:

```
PrintWriter out = response.getWriter();
out.println(strJSON);
```

HTML sivu näyttää alla olevalta heti kun käyttäjä on klikannut Hae painonappulaa pyytääkseen Servletiä lähettämään takaisin viestin:

