Use the Fetch API to POST, PUT, and DELETE Data Labs

Perform these labs on your own computer using VS Code or any editor and command prompt to ensure you understand the lessons presented in the corresponding videos and lectures.

Lab 1: POST Data Using the Fetch API

Open the **index.html** file and just **after** the <div> tag that wraps up the personId input field, add some more input fields.

```
<div>
  <label for="firstName">First Name</label>
  <input type="text" id="firstName" value="Sheila" />
</div>
<div>
  <label for="lastName">Last Name</label>
  <input type="text" id="lastName" value="Cleverly" />
</div>
<div>
  <label for="emailAddress">Email Address</label>
  <input type="email" id="emailAddress"
value="sheilac@netinc.com" />
</div>
<div>
  <label for="startDate">Start Date</label>
  <input type="date" id="startDate" value="2001-10-10"
/>
</div>
```

Just **after** the *<button onclick="getRow();">Get a Row</button>*, add a new button.

```
<button onclick="insertRow();">Insert a Row</button>
```

Just after the **getRow**() function, add a new function to gather all the input field data and create a JSON object that can be used to either insert or update a row of data.

```
function createPersonObjectFromInput() {
  return {
    "personId":
Number.parseInt(document.getElementById("personId").valu
e),
    "firstName":
document.getElementById("firstName").value,
    "lastName":
document.getElementById("lastName").value,
    "emailAddress":
document.getElementById("emailAddress").value,
    "startDate":
document.getElementById("startDate").value
  };
}
```

Add a new function named insertRow() to perform the insert.

```
function insertRow() {
  let entity = createPersonObjectFromInput();

  fetch(`${SERVER_URI}/people`, {
    method: 'POST',
    body: JSON.stringify(entity),
    headers: {
      "content-type": 'application/json'
    }
  })
    .then(response => processResponseObject(response))
    .then(response => console.log(response))
    .catch(error => console.error(error));
}
```

Try It Out

Save all your changes.

Refresh your **http://localhost:3000** page.

Put a **99** into the **Person ID** input field.

Copyright © 2024 by Paul D. Sheriff
All rights reserved worldwide. Reproduction is strictly prohibited.

Click on the **Insert a Row** button.

Bring up the Console Window and you should see a single JSON object.

Lab 2: PUT Data Using the Fetch API

Just **after** the *<button onclick="insertRow();">Insert a Row</button>*, add a new button.

```
<button onclick="updateRow();">Update a Row</button>
```

Just after the **insertRow**() function, add a new function named updateRow() to perform the update.

```
function updateRow() {
  let entity = createPersonObjectFromInput();

  fetch(`${SERVER_URI}/people/${entity.personId}`, {
    method: 'PUT',
    body: JSON.stringify(entity),
    headers: {
      "content-type": 'application/json'
    }
  })
    .then(response => processResponseObject(response))
    .then(response => console.log(response))
    .catch(error => console.error(error));
}
```

Try It Out

Save all your changes.

Refresh your **http://localhost:3000** page.

Put a **99** into the **Person ID** input field.

Change the **First Name** input field to "Ginger".

Change the **Last Name** input field to "Hollywood".

Click on the **Update a Row** button.

Bring up the Console Window and you should see a single JSON object with the changes you made.

Copyright © 2024 by Paul D. Sheriff
All rights reserved. Reproduction is strictly prohibited.

Lab 3: DELETE Data Using the Fetch API

Just **after** the *<button onclick="updateRow();">Update a Row</button>*, add a new button.

```
<button onclick="deleteRow();">Delete a Row</button>
```

Just after the **updateRow**() function, add a new function named deleteRow() to perform the update. Notice the use of **response** instead of response.json(). Because the response object sent back from a 204 no content does not have a data property to convert to a JSON object, you do not need to call the .json() method.

```
function deleteRow() {
  let id = document.getElementById("personId").value;

  fetch(`${SERVER_URI}/people/${id}`, {
    method: 'DELETE'
  })
    .then(response => processResponseObject(response))
    .then(response => console.log(response))
    .catch(error => console.error(error));
}
```

Try It Out

Save all your changes.

Refresh your **http://localhost:3000** page.

Put a **99** into the **Person ID** input field.

Click on the **Delete a Row** button.

Bring up the Console Window and you should see a 204 is returned.

Copyright © 2024 by Paul D. Sheriff
All rights reserved worldwide. Reproduction is strictly prohibited.