

How To Auto Add And Send Multiple Data At Once Snippet

```
import React, { useEffect, useState } from 'react';
import { Button, Col, Container, Form, Row, Table } from 'react-bootstrap';
import { Show, ValService } from '../../utils/service';
import { useExternalAPI } from '../../hooks/useExternalAPI';
import Select from 'react-select';
import { apiServer } from '../../context/env.context';

const RequestInterns = () => {
  const { GetRequest, PostRequest } = useExternalAPI();

  const [rowList, setRowValues] = useState([
    { region: '', branch: '', city: '', course: '', slot: '', startDate: '', endDate:
      '', },
  ]);
  const [regionList, setRegions] = useState([]);
  const [courseList, setCourses] = useState([]);
  const [reload, setReload] = useState(false);

  const TheRegions = [
    'Ahafo',
    'Ashanti',
    'Bono',
    'Bono East',
    'Central',
    'Eastern',
    'Greater Accra',
    'North East',
    'Northern',
    'Oti',
    'Savannah',
    'Upper East',
    'Upper West',
    'Volta',
    'Western',
    'Western North',
  ];
  const regionsOptions = TheRegions.map(region => ({
    label: region,
    value: region,
  }));

  const TheCourse = [
    'BTech Heating, Ventilation & Air Conditioning Engineering',
    'Bachelor of Technology in Chemical Engineering',
    'Bachelor of Technology in Building Technology (Top-Up)',
    'Bachelor of Technology in Estate Management (Top-up)',
    'Bachelor of Technology in Entrepreneurship & Finance',
  ];
```

```

'Bachelor of Technology in Electrical/Electronic Engineering (Top-Up)',
'Bachelor of Technology in Civil Engineering (Two Years – Top up)',
'Bachelor of Technology in Civil Engineering (4 years)',
'Bachelor of Technology in Hospitality Management and Catering Technology',
'Bachelor of Technology in Data Science (Regular/Evening)',
'Bachelor of Technology in Environmental Statistics',
'Bachelor of Technology in Agribusiness with Entrepreneurship',
'Bachelor of Technology in Secretaryship and Management',
'Bachelor of Technology in Mechanical Engineering',
'Bachelor of Technology in Fashion Design and Textiles Studies (Regular/Evening)',
'Bachelor of Technology In Accounting With Computing (4 Years)',
'Bachelor of Technology in Accounting with Computing (Top Up)',
'Bachelor of Technology in Electrical/Electronic Engineering',
'Bachelor of Technology in Building Technology',
'Bachelor of Technology in Pharmaceutical Sciences',
'Bachelor of Technology in Marketing (Regular/Weekend) Top-up',
'Bachelor of Technology in Fashion Design And Modeling (Top-up)',
'Bachelor of Technology in Health Statistics (Top-Up)',
'Bachelor of Technology in Library and Information Science',
'Bachelor of Technology in Procurement and Supply Chain
Management(Regular/Evening/Weekend)',

```

```

];

```

```

const CourseOptions = TheCourse.map(course => ({
  label: course,
  value: course,
}));

```

```

useEffect(() => {
  if (reload) {
    window.location.reload();
  }
}, [reload]);

```

```

const handleChangeX = (event, indx, namex) => {
  let newFormValues = [...rowList];
  if (event.target !== undefined) {
    const { name, value } = event.target;
    // Check if the property name is 'slot' and parse it as an integer
    if (name === 'slot') {
      newFormValues[indx][name] = parseInt(value, 10);
    } else {
      newFormValues[indx][name] = value;
    }
  } else {
    newFormValues[indx][namex] = event.value;
  }
  setRowValues(newFormValues);
}

```

```

};
const [userCompanyId, setUserCompanyId] = useState('');

useEffect(() => {
  var id = sessionStorage.getItem("AdminCompanyId");
  setUserCompanyId(id)
}, );
const handleSubmit = async (event) => {
  event.preventDefault();
  try {
    const requestDataArray = rowList.map((row) => {
      return {
        branch: row.branch,
        city: row.city,
        course: row.course,
        endDate: row.endDate,
        region: row.region,
        slot: row.slot,
        startDate: row.startDate,
        userCompanyId: userCompanyId, // Update this with the actual userCompanyId if
        needed
      };
    });
    const requests = requestDataArray.map(async (requestData) => {
      try {
        const response = await fetch(apiServer + '/api/Auth/requestInternNew?CompanyId=' +
          userCompanyId, {
          method: 'POST',
          headers: {
            'Content-Type': 'application/json',
          },
          body: JSON.stringify(requestData),
        });
        if (response.ok) {
          return 'Request Intern Was Successful';
        } else {
          throw new Error('Request Intern failed');
        }
      } catch (error) {
        throw error;
      }
    });
    const responses = await Promise.all(requests);
    responses.forEach((response) => {
      if (response === 'Request Intern Was Successful') {
        Show.Success('Request Intern Was Successful');
      } else {
        Show.Attention('Request Intern failed');
      }
    });
  });

```

```

setReload(true); // Reload the page or update the data as needed
} catch (error) {
Show.Attention(error);
}
};

const addRow = () => {
setRowValues([...rowList, {}]);
};

const removeRow = (i) => {
let newFormValues = [...rowList];
if (newFormValues.length > 1) {
newFormValues.splice(i, 1);
setRowValues(newFormValues);
}
};

return (
<div style={{ marginTop: 20, display: 'flex' }}>
<div style={{ flex: 1 }}>
<div style={{ width: '100%' }}>
<div
style={{
border: '1px solid rgba(0,0,0,0.1)',
borderRadius: 5,
padding: 5,
}}
>
<h3>Request Intern from University</h3>
</div>
</div>

<div style={{ width: '100%', clear: 'both' }}>
<Button
variant="dark"
style={{ float: 'right', marginRight: '10px' }}
className="pull-right"
onClick={addRow}
>
Add Row
</Button>
</div>

<div style={{ clear: 'both', overflowX: 'scroll', height: '350px' }}>
<Table style={{ fontSize: 12, margin: 0, padding: 0 }} striped bordered hover>
<thead>
<tr>
<th>#</th>
<th>Operating Region</th>

```

```

<th>Branch</th>
<th>City</th>
<th>Preferred Course</th>
<th>No. of Slots</th>
<th>Start Date</th>
<th>End Date</th>
<th>Action</th>
</tr>
</thead>
<tbody>
{rowList.map((item, idx) => (
<tr>
<td>{idx + 1}</td>
<td>
{' '}
<Form.Group style={{ width: '150px' }}>
<Select
name="region"
onChange={(e) => handleChange(e, idx, 'region')}
options={regionsOptions}
/>
</Form.Group>
</td>
<td>
<Form.Group className="mb-3" style={{ width: '150px' }}>
<Form.Control
name="branch"
onChange={(e) => handleChange(e, idx, 'branch')}
/>
</Form.Group>
</td>
<td>
<Form.Group className="mb-3" style={{ width: '150px' }}>
<Form.Control
name="city"
onChange={(e) => handleChange(e, idx, 'city')}
type="text"
/>
</Form.Group>
</td>
<td>
<Form.Group style={{ width: '150px' }}>
<Select
name="course"
onChange={(e) => handleChange(e, idx, 'course')}
options={CourseOptions}
/>
</Form.Group>
</td>
<td>

```

```

<Form.Group className="mb-3" style={{ width: '100px' }}>
  <Form.Control
    name="slot"
    type="number"
    onChange={(e) => handleChange(e, idx, 'slot')}
  />
</Form.Group>
</td>
<td style={{ width: '80px' }}>
  <Form.Control type="date" name="startDate" onChange={(e) => handleChange(e, idx, 'startDate')} />
</td>
<td style={{ width: '80px' }}>
  <Form.Control name="endDate" onChange={(e) => handleChange(e, idx, 'endDate')}
    type="date" />
</td>
<td>
  <button type="button" className="button remove" onClick={() => removeRow(idx)}>
    Del
  </button>
</td>
</tr>
  )})
</tbody>
</Table>
</div>

<Container fluid="md">
  <Row>
    <Col sm={12}>
      <hr />
      <Button style={{ float: 'right' }} variant="outline-dark" onClick={handleSubmit}>
        Submit Request
      </Button>
    </Col>
  </Row>
</Container>
</div>
</div>
);
};

```

```
export default RequestInterns;
```