

MỤC TIÊU

- Tạo Firebase REST API
- Kiểm thử REST API với Postman
- Sử dụng REST API với axios
- Xây dựng ứng dụng CRUD với REST API, jquery và axios

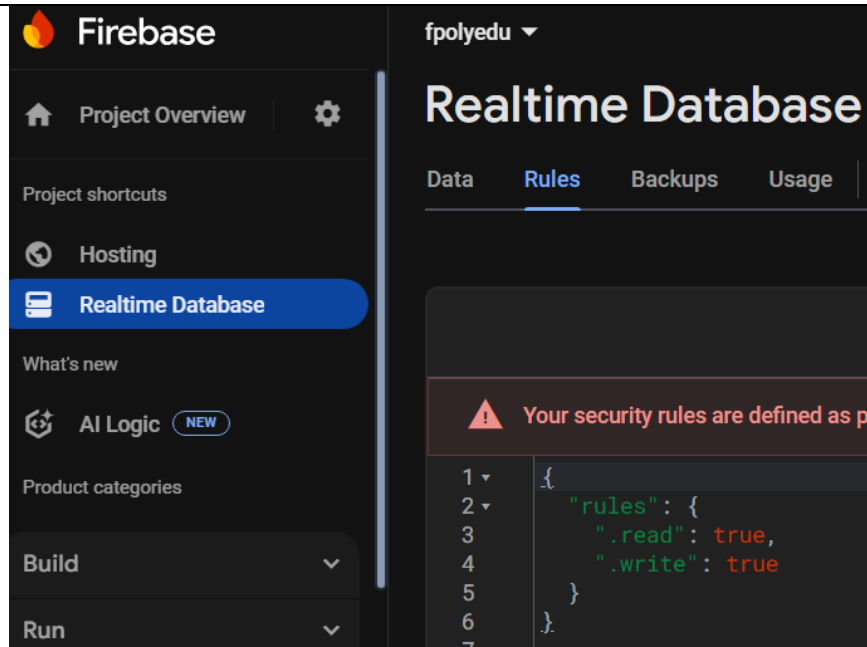
BÀI 1: TẠO FIREBASE REST API (2 ĐIỂM)

Tạo Firebase REST API và kiểm thử với Postman để quản lý danh sách sinh viên với các thuộc tính được mô tả dạng JSON như sau:

```
{  
  "id": "SV002", // mã sinh viên  
  "name": "Sinh viên 002", // họ và tên  
  "mark": 5.5, // điểm trung bình  
  "gender": true // giới tính  
}
```

Hướng dẫn thực hiện:

- Bước 1: Tạo Firebase Realtime Database
 - Bước 1: Đăng nhập Google Mail
 - Bước 2: <https://console.firebase.google.com>
 - Bước 3: tạo 1 dự án (project)
 - Bước 4: Vào mục **Realtime Database**
 - Bước 5: Chọn tab **Rules** và thay đổi quyền đọc/ghi như sau

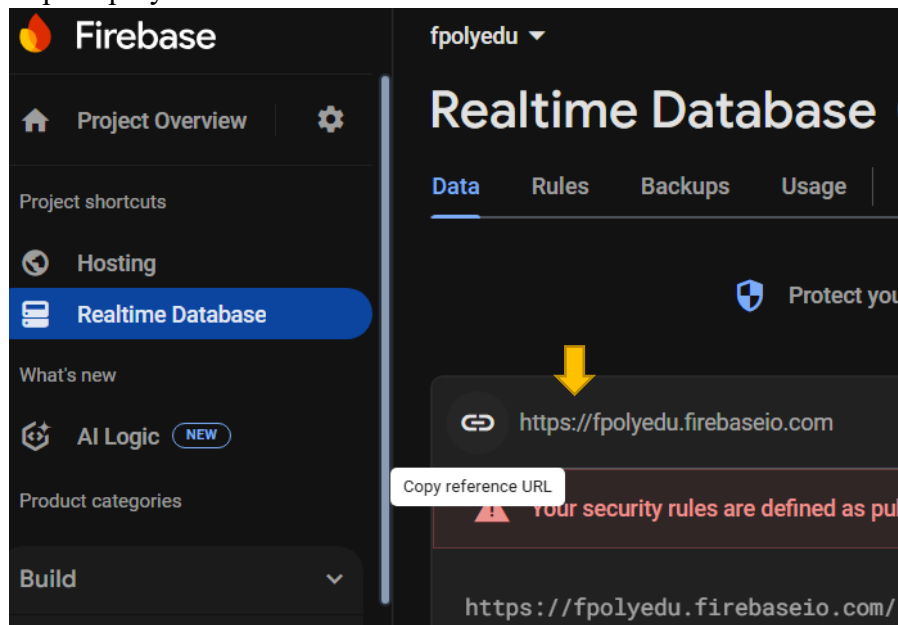


```

{
  "rules": {
    ".read": true,
    ".write": true
  }
}

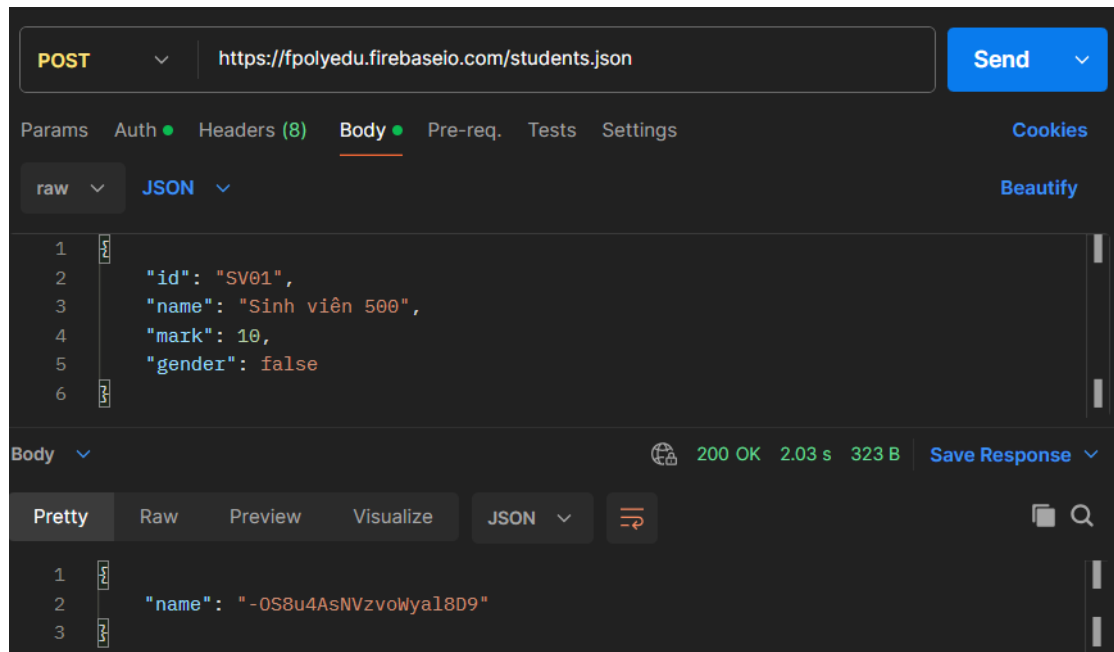
```

- Chép lấy đường dẫn cơ sở (host) của Realtime Database. Trong trường hợp này là: <https://fpolyedu.firebaseio.com>



- Bước 2: Sử dụng Firebase REST API với Postman
 - Tải và cài đặt Postman để thực hiện kiểm thử REST API

- Chạy Postman và tiến hành kiểm thử các hoạt động cơ bản với REST API của Firebase được cung cấp sẵn và lưu trữ dữ liệu vào Firebase Realtime



- Truy vấn tất cả các sinh viên:
 - ✓ GET (host/*students.json*) => {key1:student1, key2:student2,...}
- Truy xuất một sinh viên theo key:
 - ✓ GET (host/*students/key.json*) => student
- Thêm một sinh viên mới:
 - ✓ POST (host/*students.json*, *student*) => {name: key}
- Cập nhật sinh viên theo key:
 - ✓ PUT (host/*students/key.json*, *student*) => student
- Xóa sinh viên theo key:
 - ✓ DELETE (host/*students/key.json*)

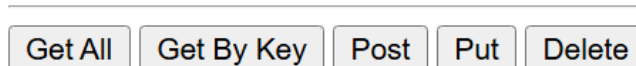
Chú ý: với các hoạt động POST và PUT thì phải có data dạng JSON trong phần body.

BÀI 2: SỬ DỤNG REST API VỚI AXIOS (2 ĐIỂM)

Sử dụng thư viện JavaScript axios để tương tác với REST API thay vì sử dụng Postman. Hãy thực hiện theo hướng dẫn như sau để thực hiện các hoạt động tương tác cơ bản với REST API.

- Bước 1: Tạo trang rest-client.html

RESTful Web API



- Giao diện gồm 5 nút bấm gọi đến 5 hàm của JavaScript
- Thêm thư viện axios vào trang web

```

<h3>RESTful Web API</h3>
<hr>
<button onclick="doGetAll()">Get All</button>
<button onclick="doGetByKey()">Get By Key</button>
<button onclick="doPost()">Post</button>
<button onclick="doPut()">Put</button>
<button onclick="doDelete()">Delete</button>

<script
src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script>
const host = "https://fpolyedu.firebaseio.com";
function doGetAll() {}
function doGetByKey() {}
function doPost() {}
function doPut() {}
function doDelete() {}
</script>

```

- Cài đặt mã cho các hàm

doGetAll()

```

var url = `${host}/students.json`;
axios.get(url).then(resp => {
    console.log("Success", resp.data);
}).catch(error => {
    console.log("Error", error);
});

```

doGetByKey()

```

var key = "<<key>>";
var url = `${host}/students/${key}.json`;
axios.get(url).then(resp => {
    console.log("Success", resp.data);
}).catch(error => {
    console.log("Error", error);
})

```

doPost()

```
var url = `${host}/students.json`;
var data = {
  "id": "SV99",
  "name": "Sinh viên 99",
  "mark": 4.5,
  "gender": true
}
axios.post(url, data).then(resp => {
  console.log("Success", resp.data);
}).catch(error => {
  console.log("Error", error);
})
```

doPut()

```
var key = "<<key>>";
var url = `${host}/students/${key}.json`;
var data = {
  "id": "SV00",
  "name": "Sinh viên 00",
  "mark": 8.5,
  "gender": true
}
axios.put(url, data).then(resp => {
  console.log("Success", resp.data);
}).catch(error => {
  console.log("Error", error);
})
```

doDelete()

```
var key = "<<key>>";
var url = `${host}/students/${key}.json`;
axios.delete(url).then(resp => {
  console.log("Success", resp.data);
}).catch(error => {
  console.log("Error", error);
})
```

- Chạy và kiểm tra kết quả qua Console

Click nút [Get All]

Success	rest-client.html:15
<ul style="list-style-type: none"> ▼ { -OS8u4AsNVzvWyaL8D9: {...}, -OS91uERpGdZ2gcFVkhT: {...}, -OS91xw-TPQB6J8PwB5y: {...} } ⓘ ▼ -OS8u4AsNVzvWyaL8D9: <ul style="list-style-type: none"> gender: false id: "SV01" mark: 10 name: "Sinh viên 500" ▶ [[Prototype]]: Object ▶ -OS91uERpGdZ2gcFVkhT: {gender: false, id: 'SV02', mark: 7.5, name: 'Sinh viên 02'} ▶ -OS91xw-TPQB6J8PwB5y: {gender: true, id: 'SV03', mark: 5.5, name: 'Sinh viên 03'} 	
Click nút [Get By Key]	
Success	rest-client.html:24
<ul style="list-style-type: none"> ▼ {gender: false, id: 'SV01', mark: 10, name: 'Sinh viên 500'} ⓘ gender: false id: "SV01" mark: 10 name: "Sinh viên 500" 	
Click nút [Post]	
Success	rest-client.html:38
<ul style="list-style-type: none"> ▼ {name: '-OS93eaGIKyvryCVXkbW'} ⓘ name: "-OS93eaGIKyvryCVXkbW" 	
Click nút [Put]	
Success	rest-client.html:53
<ul style="list-style-type: none"> ▼ {gender: true, id: 'SV006', mark: 8.5, name: 'Sinh viên 600'} ⓘ gender: true id: "SV006" mark: 8.5 name: "Sinh viên 600" 	
Click nút [Delete]	
Success null	rest-client.html:62

BÀI 3: XÂY DỰNG ỨNG DỤNG CRUD VỚI REST API (4 ĐIỂM)

Xây dựng trang web quản lý sinh viên với REST API có giao diện và hoạt động tương tác được mô tả như sau:

QUẢN LÝ SINH VIÊN

Id:

Full Name:

Everage Mark:

Gender: ☐ Male ☒ Female

Create

Update

Delete

Reset

Id	Name	Mark	Gender	
SV02	Lê Thị Hương Thảo	7.5	Female	Edit Delete
SV03	Nguyễn Đình Thiên Long	5.5	Male	Edit Delete
SV01	Nguyễn Nghiệm	9.5	Male	Edit Delete
SV04	Nguyễn Đình Hoàng Long	7.6	Male	Edit Delete
SV05	Phạm Thị Nở	5.5	Female	Edit Delete
SV06	Nguyễn Chí Phèo	4.5	Male	Edit Delete

TƯƠNG TÁC	XỬ LÝ
Khởi đầu	<ul style="list-style-type: none"> Tải và hiển thị danh sách sinh viên bảng
Edit.Click	<ul style="list-style-type: none"> Tải và hiển thị sinh viên lên form
Create.Click	<ul style="list-style-type: none"> Thêm mới sinh viên Xóa trắng form Tải và hiển thị danh sách sinh viên bảng
Update.Click	<ul style="list-style-type: none"> Cập nhật sinh viên Tải và hiển thị danh sách sinh viên bảng
Delete.Click	<ul style="list-style-type: none"> Xóa sinh viên Xóa trắng form Tải và hiển thị danh sách sinh viên bảng

Reset.Click

- Xóa trắng form

Hướng dẫn thực hiện:

- Thiết kế student-crud.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quản lý sinh viên</title>
</head>
<body>
  <h3>QUẢN LÝ SINH VIÊN</h3>
  <div>
    <p> Id: <input id="id">
    <p> Full Name: <input id="name">
    <p> Everage Mark: <input id="mark">
    <p> Gender:
    <input name="gender" type="radio" id="male">Male
    <input name="gender" type="radio" id="female">Female
  </div>
  <hr>
  <button onclick="$api.create()">Create</button>
  <button onclick="$api.update()">Update</button>
  <button onclick="$api.delete()">Delete</button>
  <button onclick="$api.reset()">Reset</button>
  <hr>
  <table border="1" style="width:100%">
    <thead>
      <tr>
        <th>Id</th>
        <th>Name</th>
        <th>Mark</th>
        <th>Gender</th>
        <th></th>
      </tr>
    </thead>
    <tbody/> <--tbody chứa các <tr-->
```



```

</table>
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script
>
<script src="student-crud.js"></script>
</body>
</html>

```

- Tổ chức mã student-crud.js

```

const host = "https://fpolyedu.firebaseio.com";
const $api = {
  get student(){...},
  set student(e){...},
  fillToTable() {...},
  edit(key) {...},
  create() {...},
  update() {...},
  delete(key) {...},
  reset(){...}
}
$api.fillToTable();

```

- Cài đặt mã student-crud.js

Get student(): đọc dữ liệu từ form

```

return {
  id: $("#id").val(),
  name: $("#name").val(),
  mark: $("#mark").val(),
  gender: $("#male").prop("checked")
}

```

Set student(e): hiển thị dữ liệu lên form

```

$("#id").val(e.id);
$("#name").val(e.name);
$("#mark").val(e.mark);
$("#male").prop("checked", e.gender);
$("#female").prop("checked", !e.gender);

```

fillToTable(): Tải và hiển thị danh sách sinh viên lên bảng

```

var url = `${host}/students.json`;
axios.get(url).then(resp => {
    $("tbody").empty();
    Object.keys(resp.data).forEach(key => {
        var e = resp.data[key];
        var tr = `<tr>
            <td>${e.id}</td>
            <td>${e.name}</td>
            <td>${e.mark}</td>
            <td>${e.gender?'Male':'Female'}</td>
            <td>
                <a href="#" onclick="$api.edit('${key}')">Edit</a>
                <a href="#" onclick="$api.delete('${key}')">Delete</a>
            </td>
        </tr>`;
        $("tbody").append(tr);
    });
}).catch(error => {
    alert("Lỗi tải danh sách sinh viên!");
})

```

Edit(key): Tải sinh viên và hiển thị lên form

```

this.key = key.trim();
var url = `${host}/students/${key}.json`;
axios.get(url).then(resp => {
    this.student = resp.data;
}).catch(error => {
    alert("Lỗi tải sinh viên!");
})

```

Create(): Thêm sinh viên mới

```

var url = `${host}/students.json`;
axios.post(url, this.student).then(resp => {
    this.fillToTable();
    this.reset();
}).catch(error => {
    alert("Lỗi thêm sinh viên mới!");
})

```

Update(): Cập nhật thông tin sinh viên

```
var url = `${host}/students/${this.key}.json`;
axios.put(url, this.student).then(resp => {
    this.fillToTable();
}).catch(error => {
    alert("Lỗi tải cập nhật sinh viên!");
})
```

Delete(key): Xóa sinh viên

```
var url = `${host}/students/${key || this.key}.json`;
axios.delete(url).then(resp => {
    this.fillToTable();
    this.reset();
}).catch(error => {
    alert("Lỗi xóa sinh viên!");
})
```

Reset(): Xóa trắng form

this.student = {}

BÀI 4: GIẢNG VIÊN CHO THÊM (2 ĐIỂM)