

# Phonebook API Documentation

---

This API allows you to manage a phonebook using only built-in Node.js modules and a JSON file as the database.

## Base URL

```
http://localhost:3000
```

---

## Endpoints

### 1. Get All Contacts

- **URL:** `/contacts`
- **Method:** `GET`
- **Query Parameters:**
  - `search` (optional): Filter contacts by name or phone (case-insensitive)
- **Response:** `200 OK`
  - Returns an array of contact objects.

### Example Response

```
[  
  { "id": "abc123", "name": "John Doe", "phone": "1234567890" },  
  { "id": "def456", "name": "Jane Smith", "phone": "9876543210" }  
]
```

---

### 2. Add a New Contact

- **URL:** `/contacts`
- **Method:** `POST`
- **Request Body:** (JSON)

```
{  
  "name": "Contact Name",  
  "phone": "Contact Phone"  
}
```

- **Response:** `201 Created`
  - Returns the created contact object with a generated `id`.

## Example Response

```
{
  "id": "xyz789",
  "name": "Contact Name",
  "phone": "Contact Phone"
}
```

---

### 3. Update an Existing Contact

- **URL:** `/contacts/:id`
- **Method:** `PUT`
- **Request Body:** (JSON, at least one field required)

```
{
  "name": "New Name",    // optional
  "phone": "New Phone"  // optional
}
```

- **Response:** `200 OK`
  - Returns the updated contact object.
- **Errors:**
  - `404 Not Found` if contact does not exist.

---

### 4. Delete a Contact

- **URL:** `/contacts/:id`
- **Method:** `DELETE`
- **Response:** `200 OK`
  - Returns the deleted contact object.
- **Errors:**
  - `404 Not Found` if contact does not exist.

---

### 5. CORS Support

- All endpoints support CORS for cross-origin requests.

---

## Error Responses

- All errors are returned as JSON objects with an `error` field.

### Example

```
{
  "error": "Contact not found"
}
```

---

## Example cURL Requests

- **Get all contacts:** `curl http://localhost:3000/contacts`
- **Search contacts:** `curl "http://localhost:3000/contacts?search=john"`
- **Add contact:** `curl -X POST -H "Content-Type: application/json" -d '{"name":"Alice","phone":"5551234"}' http://localhost:3000/contacts`
- **Update contact:** `curl -X PUT -H "Content-Type: application/json" -d '{"phone":"5550000"}' http://localhost:3000/contacts/abc123`
- **Delete contact:** `curl -X DELETE http://localhost:3000/contacts/abc123`

---

## Notes

- The database is stored in `contacts.json` in the project directory.
- All IDs are randomly generated strings.
- No authentication is required.