Technical Module: API (Application Programming Interface)

Introduction

As software technology evolves, the need for integration between applications is increasing. Applications no longer stand alone but are interconnected to exchange data and services. The API (Application Programming Interface) serves as the main bridge that enables this.

Basic Concepts of API

1. Client–Server: APIs operate within a client-server architecture.
2. Request–Response Cycle:
- Request contains endpoint, method, header, and body.
- Response contains data (JSON/XML) and status codes.
3. Endpoints and Resources:
- Endpoint is a specific URL.
- Resource is the object managed by the API.

Types of API

1. Public API (Open API) – accessible to everyone.
2. Private API – used only internally.
3. Partner API – for official partners.
4. Composite API – combines multiple requests at once.

API Architecture

1. REST API: uses HTTP, JSON, and is stateless.
2. SOAP API: uses XML, strict rules, suitable for transactions.
3. GraphQL: the client defines the data it needs.

Main Components of an API

- Endpoint: example /api/v1/users
- Method: GET, POST, PUT, DELETE
- Header: additional information
- Body: data sent in the request
- Response: server’s reply

API Security

- API Key: unique code for identification
- OAuth 2.0: modern authentication standard
- Rate Limiting: limits the number of requests

- CORS: controls cross-domain access

Example CRUD Application with Node.js + Express

The following code creates a simple CRUD API for user data:

```
const express = require('express');
const app = express();
app.use(express.json());

let users = [
  { id: 1, name: "Samuel", email: "samuel@example.com" }
];

// GET all users
app.get('/api/users', (req, res) => res.json(users));

// GET user by ID
app.get('/api/users/:id', (req, res) => {
  const user = users.find(u => u.id == req.params.id);
  user ? res.json(user) : res.status(404).json({ message: "Not found" });
});

// POST new user
app.post('/api/users', (req, res) => {
  const newUser = { id: users.length + 1, ...req.body };
  users.push(newUser);
  res.status(201).json(newUser);
});

// PUT update user
app.put('/api/users/:id', (req, res) => {
  let user = users.find(u => u.id == req.params.id);
  if (user) {
    user.name = req.body.name || user.name;
    user.email = req.body.email || user.email;
    res.json(user);
  } else {
    res.status(404).json({ message: "Not found" });
  }
});

// DELETE user
app.delete('/api/users/:id', (req, res) => {
  users = users.filter(u => u.id != req.params.id);
  res.json({ message: "User deleted" });
});

app.listen(3000, () => console.log('API running at http://localhost:3000'));
```

Conclusion

APIs are the foundation of modern application development. With APIs, application integration becomes faster, more efficient, and more secure. This module covers the basics of APIs, their types, architecture, security, and provides a simple CRUD implementation example.