**API:**

* It is the software responsible for the connection for the communication and information exchange between two apps

**TYPES OF WEB API ARCHITECUTRE AND PROTOCOL:**

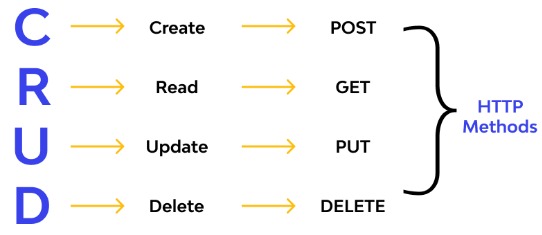
* REST (Representation State Transfer)
* SOAP (Simple Object Access Protocol)
* JSON and XML RPC (Remote Procedural Call)

**REST API:**

* **Stands for Representation State Transfer**
* **A** software architectural style that defines the constraints to create web services
* Constraints:
  + Uniform Interface
  + Stateless
  + Cacheable
  + Client-Server
  + Layered System
  + Code on demand

**CRUD OPERATIONS:**

Used to manage records in a database table

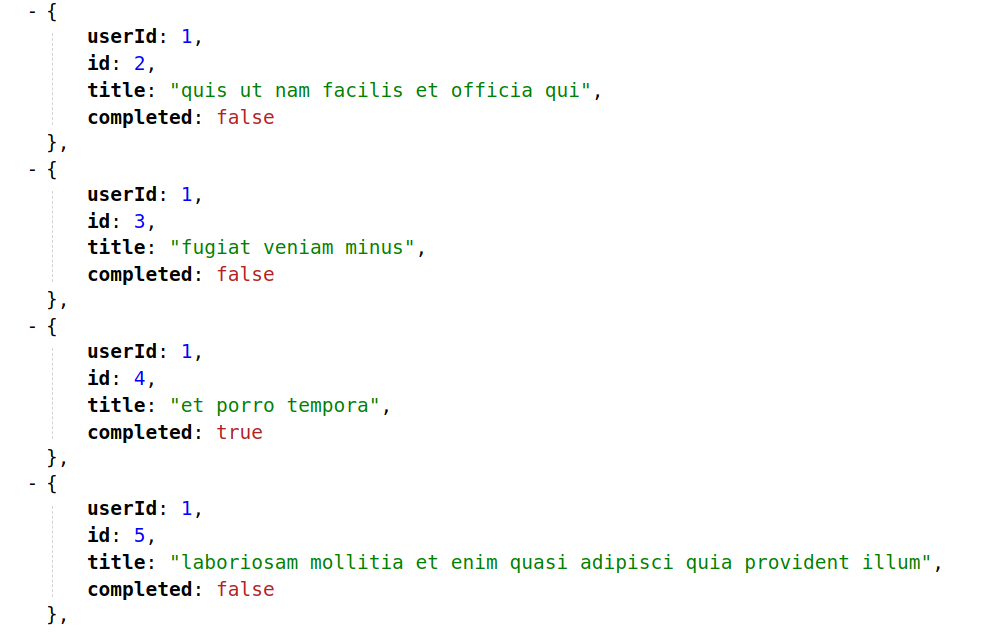


**END POINT:**

* The specific digital location where requests for information are sent by one program to retrieve the digital resource that exists there
* We use HTTP methods (like get, post, put, delete) to make use of end point

**JSON:**

* Stands for **JavaScript Object Notation**
* A text format for storing and transporting data
* Light weighted data interchange format
* Example:



**SPRING:**

* Spring Framework is the most popular application development framework of Java.
* The main feature of the Spring Framework is **dependency Injection** or **Inversion of Control** (IoC).
* With the help of Spring Framework, we can develop a **loosely** coupled application. It is better to use if application type or characteristics are purely defined.

**SPRING BOOT:**

* Spring Boot is a module of Spring Framework.
* It allows us to build a stand-alone application with minimal or zero configurations.
* It is better to use if we want to develop a simple Spring-based application or RESTful services.

**SPRING BOOT ARCHITECTURE:**



**Presentation Layer:**

* Handles the HTTP requests, translates the JSON parameter to object, and authenticates the request and transfer it to the business layer.
* It consists of **views** i.e., frontend part.

**Business Layer:**

* Handles all the **business logic**.
* It consists of service classes and uses services provided by data access layers.
* It also performs **authorization** and **validation**.

**Persistence Layer:**

* Contains all the **storage logic** and translates business objects from and to database rows.

**Database Layer:**

* **CRUD** (create, retrieve, update, delete) operations are performed.

**SPRING BOOT FLOW ARCHITECTURE:**



**ANNOTATIONS:**

* **@SpringBootApplication**
* **@RestController**
* **@Service**
* **@Repository**
* **@GetMapping**
* **@PostMapping**
* **@RequestMapping**
* **@Autowired**
* **@Value**
* **@Jsonignore**
* **@Jsonproperty**
* **@RequestBody**
* **@Entity**
* **@Id**