API Application Programming Interfaces

- An API is a way for a computer program to communicate directly with another computer program
- It is typically used to perform CRUD operations
- The two main types of APIs for web services (can run over the Internet, typically use HTTP) are SOAP and REST
- NETCONF and RESTCONF are APIs specifically designed to work with network devices

CRUD

Operation	SQL	HTTP
Create	INSERT	PUT / POST
Read (Retrieve)	SELECT	GET
Update (Modify)	UPDATE	PUT / POST / PATCH
Delete (Destroy)	DELETE	DELETE

HTTP Verbs:

- Put: Update / replace a resource
- Post: Create a new resource
- Patch: Update / modify a resource

SOAP Simple Object Access Protocol

- SOAP is a standard communication protocol system that permits processes using different operating systems like Linux and Windows to communicate.
- The transport is typically HTTP(S), and the data format is always XML.
- Because it is a protocol it has strict standards to adhere to.

REST Representational State Transfer

- REST is an architecture, not a protocol. It gives guidelines for the structure and organization of an API.
- It supports any transport and data format.
- HTTP(S) transport and JSON (or XML) data format are commonly used.
- It typically has faster performance and is easier to work with than SOAP.

REST Constraints

- Client-server architecture: the client sends a request, the server sends a response.
- Uniform Interface: provides simplicity.
- Statelessness: no client context is stored on the server between requests.
- Cacheability: responses must define themselves as either cacheable or non-cacheable.

REST Constraints (Cont.)

- Layered system: any intermediary devices such as load balancers must be transparent to the client and server.
- Code on demand (optional): servers can temporarily extend or customize the functionality of a client by transferring executable code.

REST Authentication Types

- **No authentication**
- **Basic Authentication:** Username and password sent in plain text. Insecure unless encrypted (eg over HTTPS), does not support granular permissions per user.
- **API Key:** Uses encrypted tokens, usually created in admin portal. The same user can request tokens with different permissions. The server must check the API key permissions on every client request. Usually have long expiry.
- **Bearer Token:** Basic or API Key authenticated user requests encrypted Bearer Token which is then automatically included in other requests. Permissions are embedded in the token so it puts less load on server than API Key. The token is transferrable to other applications and usually has a short expiry.

REST Request URL

https://demo.flackbox.com|/api/running/aaa/users/neil|?dryrun

Target Host

Resource

Parameters (Optional)

- Request method must be sent (Get, Post etc)
- Headers with key:value pair information about the request can be added (eg accept:application/json, credentials)
- Post, Put and Patch requests include data in the body

REST Response Codes

- 1xx: Informational
- 2xx: Success
  - 200: OK
  - 201: Created
  - 204: No Content (deleted)
- 3xx: Redirection

REST Response Codes - Errors

- 4xx: Client Error
  - 400: Bad request / malformed syntax
  - 401: Unauthorized
  - 403: Forbidden
  - 404: Not Found
- 5xx: Server Error
  - 500: Internal Server Error

- Responses to Get requests include data in the body
- Headers can also be included in the response