



CIS 112

Intro to Programming Using Python

Module 5 Part 2

Agenda for the Day!

— — —

- Network Programming
 - CRUD,
 - building a simple REST client,

CRUD and RESTful APIs

Intro to RESTful

RESTful stands for **Representational State Transfer** and is an architectural style for designing networked applications.

RESTful systems are characterized by the following key principles:

- **Client-Server Architecture:** The system is divided into clients and servers, where clients initiate requests and servers process those requests and return responses.
- **Statelessness:** Each request from a client to the server must contain all the necessary information to understand and process the request. The server does not store any client state between requests.
- **Cacheability:** Responses from the server can be marked as cacheable or non-cacheable. Cacheable responses can be reused by the client or intermediate caching servers to improve performance and reduce latency.
- **Uniform Interface:** The interface between clients and servers should be uniform and consistent, making it easy for clients to understand how to interact with the server. This includes using standard methods for CRUD operations and following conventions for resource naming and representation.
- **Layered System:** The architecture is composed of multiple layers, with each layer responsible for a specific aspect of the system. This allows for scalability, flexibility, and separation of concerns.

In practice, a RESTful API adheres to these principles by exposing resources as URLs and using standard HTTP methods to perform CRUD operations on those resources. For example.

Intro to CRUD

CRUD is an acronym that stands for Create, Read, Update, and Delete. It represents the four basic operations that can be performed on persistent storage, such as databases. These operations are fundamental to most database applications and form the basis of many data-driven applications and APIs. Here's an overview of each operation:

Create (C):

- The Create operation involves adding new data to the database.
- It typically corresponds to the INSERT SQL statement in relational databases.
- In an API context, it involves sending data to the server to be stored.

Read (R):

- The Read operation involves retrieving existing data from the database.
- It corresponds to the SELECT SQL statement in relational databases.
- In an API context, it involves fetching data from the server.

Update (U):

- The Update operation involves modifying existing data in the database.
- It corresponds to the UPDATE SQL statement in relational databases.
- In an API context, it involves sending updated data to the server to replace or modify existing records.

Delete (D):

- The Delete operation involves removing data from the database.
- It corresponds to the DELETE SQL statement in relational databases.
- In an API context, it involves sending a request to the server to remove specific data.

Quick note on HTTP...

- The Hypertext Transfer Protocol (HTTP) is the application layer protocol that powers the World Wide Web
 - It is used to load and interact with webpages using hypertext links.
- A typical flow involves a client machine making an HTTP request to a server, which then sends a response message.
- Each HTTP request made across the Internet carries with it a series of encoded data that carries different types of information.
- For instance each request includes a 'Method', sometimes referred to as an HTTP verb, indicates the action that the HTTP request expects from the queried server.
 - For example, two of the most common HTTP methods are 'GET' and 'POST';
 - a 'GET' request expects information back in return (usually in the form of a website),
 - a 'POST' request typically indicates that the client is submitting information to the web server (such as form information, e.g. a submitted username and password).



▼ Request Headers

```
:authority: www.google.com
:method: GET
:path: /
:scheme: https
accept: text/html
accept-encoding: gzip, deflate, br
accept-language: en-US,en;q=0.9
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0
```

RESTful meet CRUD!

— — —

These CRUD operations form the basis of most database interactions and are essential for managing and manipulating data in a systematic way. They are often mapped to corresponding HTTP methods in RESTful APIs:

- **Create:** Typically mapped to the **HTTP POST** method.
- **Read:** Mapped to the **HTTP GET** method.
- **Update:** Mapped to the **HTTP PUT** or **PATCH** method.
- **Delete:** Mapped to the **HTTP DELETE** method.

CRUD operations are not limited to traditional databases; they are also applicable to other forms of data storage and manipulation, such as file systems or in-memory data structures.

They provide a simple and intuitive framework for performing common data management tasks in software applications.

Python and RESTful

Working in Python

- In Python, you can use various methods to perform HTTP requests. These methods are typically available through different libraries or modules. Some of the most common methods and the corresponding libraries/modules in Python are:
 - Requests
 - Urllib.request
 - http.client
- Using requests Module:
 - The requests module is a popular third-party library for making HTTP requests in Python. It provides a high-level interface for sending HTTP requests and handling responses.

Requests Example

— — —

The **requests** module is a popular third-party library for making HTTP requests in Python. It provides a high-level interface for sending HTTP requests and handling responses.

- Get

```
import requests

response = requests.get('http://example.com')
```

- Post

```
data = {'key': 'value'}
response = requests.post('http://example.com', data=data)
```

- Put/Delete

```
response = requests.put('http://example.com/put')
response = requests.delete('http://example.com/delete')
```

