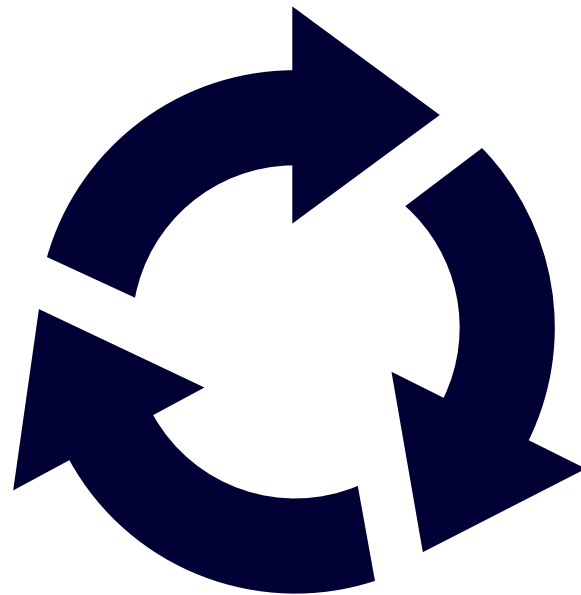


Pythonkurs - En kort introduksjon til API

Klient

Tjener



```
In [17]:  
  
# Detour: Decoding a base64 string.  
import base64  
  
decoded_string = base64.b64decode("cG9zdG1hbJpwYXNkd29yZA==")  
print(f"The decoded_string is: {decoded_string}")  
  
encoded_string = base64.b64encode(b"postman:password")  
  
print(f"The encoded_string is: {encoded_string}")  
  
The decoded_string is: b'postman:password'  
The encoded_string is: b'cG9zdG1hbJpwYXNkd29yZA=='
```

```
In [17]:  
  
# Detour: Decoding a base64 string.  
import base64  
  
decoded_string = base64.b64decode("cG9zdG1hbJpwYXNkd29yZA==")  
print(f"The decoded_string is: {decoded_string}")  
  
encoded_string = base64.b64encode(b"postman:password")  
  
print(f"The encoded_string is: {encoded_string}")  
  
The decoded_string is: b'postman:password'  
The encoded_string is: b'cG9zdG1hbJpwYXNkd29yZA=='
```

```
In [17]:  
  
# Detour: Decoding a base64 string.  
import base64  
  
decoded_string = base64.b64decode("cG9zdG1hbJpwYXNkd29yZA==")  
print(f"The decoded_string is: {decoded_string}")  
  
encoded_string = base64.b64encode(b"postman:password")  
  
print(f"The encoded_string is: {encoded_string}")  
  
The decoded_string is: b'postman:password'  
The encoded_string is: b'cG9zdG1hbJpwYXNkd29yZA=='
```

```
In [20]:  
  
# The POST method is used to post data, such as files or forms on the we  
b. Essentially the POST method enables the creation of new resources. A  
successful POST request will typically return a status code of 201.  
  
import requests  
  
url = "https://jsonplaceholder.typicode.com/posts"  
data = {"title" : "Sample POST request", "body" : "How to make a POST re  
quest"}  
response = requests.post(url=url, data=data)  
print(f"The response status is {response.status_code}\n")  
  
The response status is 201
```

-
- GET: Gir oss mulighet til å hente data. Svaret kommer ofte i form av en JSON datastruktur.
 - POST: Gir oss mulighet til å sende inn data.
 - DELETE: Gir oss mulighet til å slette data.
 - PUT: Gir oss mulighet til å oppdatere data.
-

Eksempel GET

% curl <https://jsonplaceholder.typicode.com/posts>

```
[
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum re
rum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro
vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  },
  {
    "userId": 1,
    "id": 3,
    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
    "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nm
olestiae porro eius odio et labore et velit aut"
  },
  {
    "userId": 1,
    "id": 4,
    "title": "eum et est occaecati",
    "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt
rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
  },
  {
    "userId": 1,
    "id": 5,
    "title": "nesciunt quas odio",
    "body": "repudiandae veniam quaerat sunt sed\nalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus qui
s\nest aut tenetur dolor neque"
  },
  {
    "userId": 1,
    "id": 6,
    "title": "dolorem eum magni eos aperiam quia",
    "body": "voluptatem doloremque illo voluptate perferendis nostrum nostrum"
  }
]
```

Eksempel POST

```
% curl -X POST -d "userId=5&title=Hello World&body=Post body." https://jsonplaceholder.typicode.com/posts
```

```
-bash-4.2$ curl -X POST -d "userId=5&title=Hello World&body=Post body." https://jsonplaceholder.typicode.com/posts
{
  "userId": "5",
  "title": "Hello World",
  "body": "Post body.",
  "id": 101
}-bash-4.2$ █
```

Eksempel DELETE

% curl -X DELETE <https://jsonplaceholder.typicode.com/posts/5>

```
-bash-4.2$ curl -X DELETE https://jsonplaceholder.typicode.com/posts/5  
{}-bash-4.2$ █
```

Eksempel PUT

% curl -X PUT -d "userId=5&title=Hello World&body=Post body." <https://jsonplaceholder.typicode.com/posts/5>

```
-bash-4.2$ curl -X PUT -d "userId=5&title=Hello World&body=Post body." https://jsonplaceholder.typicode.com/posts/5
{
  "userId": "5",
  "title": "Hello World",
  "body": "Post body.",
  "id": 5
}-bash-4.2$ █
```

-
- 200 : OK.
 - 401 : Authentication failed!
 - 403 : Access is forbidden by the API service.
 - 404 : The requested API service is not found on the server/web.
 - 500 : Internal Server Error has occurred.

Eksempel GET Python

```
import requests
api_url = "https://jsonplaceholder.typicode.com/todos/1"
response = requests.get(api_url)
response.json()
```

`{'userId': 1, 'id': 1, 'title': 'delectus aut autem', 'completed': False}`

```
In [1]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos/1"
response = requests.get(api_url)
response.json()
```

```
Out[1]: {'userId': 1, 'id': 1, 'title': 'delectus aut autem', 'completed': False}
```

Eksempel POST Python

```
import requests
api_url = "https://jsonplaceholder.typicode.com/todos"
todo = {"userId": 1, "title": "Buy milk", "completed": False}
response = requests.post(api_url, json=todo)
response.json()

response.status_code
```

201

```
In [2]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos"
todo = {"userId": 1, "title": "Buy milk", "completed": False}
response = requests.post(api_url, json=todo)
response.json()

response.status_code
```

Out[2]: 201

Eksempel Delete Python

```
import requests
api_url = "https://jsonplaceholder.typicode.com/todos/10"
response = requests.delete(api_url)
response.json()

response.status_code
```

200

```
In [4]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos/10"
response = requests.delete(api_url)
response.json()

response.status_code
```

Out[4]: 200

Eksempel PUT Python

```
import requests
api_url = "https://jsonplaceholder.typicode.com/todos/10"
response = requests.get(api_url)
response.json()
```

```
todo = {"userId": 1, "title": "Wash car", "completed": True}
response = requests.put(api_url, json=todo)
response.json()
```

```
response.status_code
200
```

```
In [3]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos/10"
response = requests.get(api_url)
response.json()

todo = {"userId": 1, "title": "Wash car", "completed": True}
response = requests.put(api_url, json=todo)
response.json()

response.status_code
```

```
Out[3]: 200
```

```
In [1]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos/1"
response = requests.get(api_url)
response.json()
```

```
Out[1]: {'userId': 1, 'id': 1, 'title': 'delectus aut autem', 'completed': False}
```

```
In [2]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos"
todo = {"userId": 1, "title": "Buy milk", "completed": False}
response = requests.post(api_url, json=todo)
response.json()

response.status_code
```

```
Out[2]: 201
```

```
In [3]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos/10"
response = requests.get(api_url)
response.json()

todo = {"userId": 1, "title": "Wash car", "completed": True}
response = requests.put(api_url, json=todo)
response.json()

response.status_code
```

```
Out[3]: 200
```

```
In [4]: import requests
api_url = "https://jsonplaceholder.typicode.com/todos/10"
response = requests.delete(api_url)
response.json()

response.status_code
```

```
Out[4]: 200
```