Purwadhika
Digital Technology School

Full Stack Web Development

Advanced Topic

Caching (redis), error handling, logging and debugging

Job Connector Program

Outline

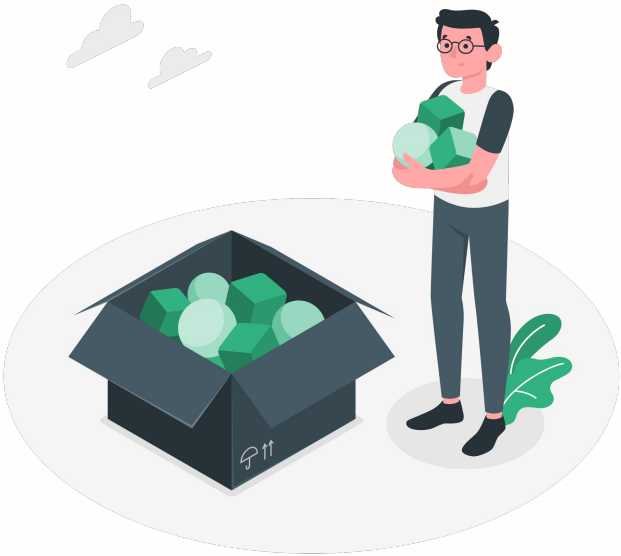
Purwadhika
Digital Technology School

- Caching data using Redis
- Error handling
- Logging

What is Caching ?

Purwadhika
Digital Technology School

Caching is the process of storing copies of files in a cache or a temporary storage location so that they can be accessed more quickly.



Why do we cache ?

- To save cost. Such as paying for bandwidth or even volume of data sent over the network.
- To reduce app response time.

What is Redis ?

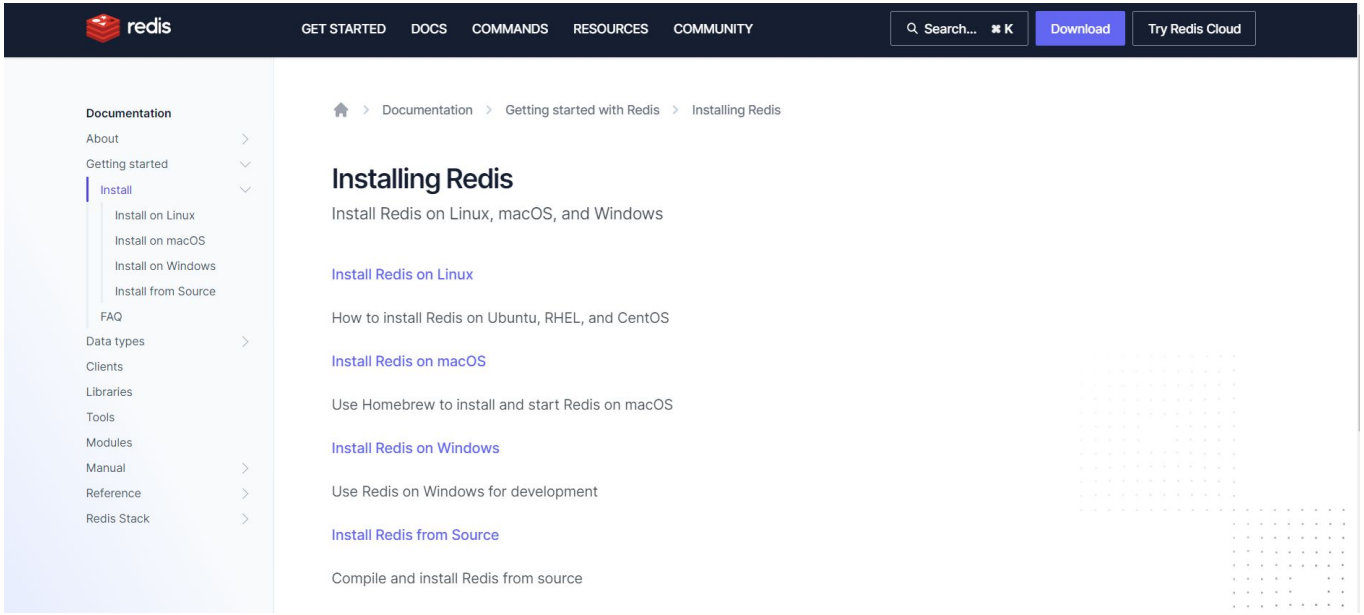
Purwadhika
Digital Technology School

Redis Stands for **Remote Dictionary Server**. The open source, in-memory data store used by millions of developers as a database, cache, streaming engine, and message broker.

Install Redis Server

Purwadhika Digital Technology School

Go to this link to download redis and select installer based on your OS
https://redis.io/docs/getting-started/installation/



Install Redis Server

Purwadhika
Digital Technology School

Or you can just download from this link
https://drive.google.com/file/d/1Rxx328YzOlwOEOEqYwCmMinZHYlXs8of/view

Extract and open the folder, and run **redis-server.exe**

This only work on npm redis package

redis: ^2.8.0

```
[15064] 28 Sep 12:58:18.837 # Warning: no config file specified, using the default config. In order to specify a config
file use C:\Users\62821\Downloads\redis-latest\redis-latest\redis-server.exe /path/to/redis.conf

                Redis 3.0.503 (00000000/0) 64 bit

                Running in standalone mode
                Port: 6379
                PID: 15064

                      http://redis.io

[15064] 28 Sep 12:58:18.840 # Server started, Redis version 3.0.503
[15064] 28 Sep 12:58:18.840 * DB loaded from disk: 0.000 seconds
[15064] 28 Sep 12:58:18.840 * The server is now ready to accept connections on port 6379
```

Using Redis in NodeJS

Purwadhika
Digital Technology School

Initiate nodejs new project and install redis package into your projects.

In this case we will working on axios to fetch data from existing API

```
npm init --y
npm install express redis axios --save
```

Create Simple Express App

Purwadhika
Digital Technology School

index.js

```
const express = require('express');
const port = 3000;
const app = express();

app.listen(port, () => {
  console.log(`Server runing on port ${port}`);
});
```

Create Simple Express App

Purwadhika
Digital Technology School

Create new route to get data from external API. In this example, we will use dog API. It takes about 500 - 800 ms to fetch the datas.

We will improve fetching speed with caching the data with **Redis**.

index.js

```
const axios = require('axios');

app.get('/dogs/:breed', async (req, res) => {
  try {
    const { breed } = req.params;
    const { data } = await axios.get(
      `https://dog.ceo/api/breed/${breed}/images`,
    );
    res.status(200).send(data);
  } catch (error) {
    res.status(400).send(error);
  }
});
```

Implementing Redis in Express

Purwadhika
Digital Technology School

Lets implement redis into our code. Put this code below your axios import.

This is a setup for redis in your server app.

By default redis is using port 6379.

index.js

```
//import redis
const redis = require('redis');

//define connection
const client = redis.createClient(6379);

//check connection
client.on('error', (error) => {
  console.log(error);
});
```

Implementing Redis in Express

Purwadhika
Digital Technology School

Client.get method is used to get data source from cache storage.

Client.setex method is used to keep and cache the data with expires time. Data would be keep in string type.

index.js

```
app.get('/dogs/:breed', async (req, res) => {
  try {
    const { breed } = req.params;
    //check data with breed key
    client.get(breed, async (err, dogs) => {
      if (dogs) {
        return res.status(200).send(dogs);
      }
      const { data } = await axios.get(
        `https://dog.ceo/api/breed/${breed}/images`,
      );
      // cache data with breed as a key and set expires time
      client.setex(breed, 100, JSON.stringify(data));

      return res.status(200).send(data);
    });
  } catch (error) {
    console.log(error);
    res.status(400).send(error);
  }
});
```

Error Handling

Purwadhika
Digital Technology School

Errors in Node.js are handled through exceptions. An exception is created using the **throw** keyword. After an error occurs, the normal program flow is halted and the control is held back to the nearest exception handler.

```
const doSomething1 = () => {
  // ...
  try {
    // ...
    // error here, execute 'throw'
  } catch (err) {
    // ... handle it locally
  }
  // ...
};
```

Error Handling in ExpressJS

Purwadhika
Digital Technology School

```
try {

  // an error occurs ...

  return res.status(200).send(data);
} catch (error) {
  // handle error, send HTTP 500 to the client
  return res.status(500).send("Internal server error");
}
```

Logging

Purwadhika
Digital Technology School

Logging are crucial for monitoring and troubleshooting your Node.js app. There are several ways to implement logging :

- Using runtime’s console
- Using 3rd party library

Using Runtimes Console

Purwadhika
Digital Technology School

- console.log
- console.warn
- console.error

```
Failed to load resource: the server responded with a status of 403 ()                chromewebdata/:1
Refused to display 'https://cdn.connectad.io/' in a frame because it set 'X-Frame-Options' to 'sameorigin'.   chromewebdata/:1
Refused to display 'https://cdn.connectad.io/' in a frame because it set 'X-Frame-Options' to 'sameorigin'.   chromewebdata/:1
▶crbug/1173575, non-JS module files deprecated.
▶crbug/1173575, non-JS module files deprecated.                                       (index):6789
Access to XMLHttpRequest at 'https://i.connectad.io/api/v2' from origin 'https://www.w3schools.com' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.   tryit.asp:1
Failed to load resource: net::ERR_FAILED                                             i.connectad.io/api/v2:1
DevTools failed to load source map: Could not load content for https://c.amazon-adsystem.com/aax2/apstag.js.map: HTTP error: status code 403, net::ERR_HTTP_RESPONSE_CODE_FAILURE
Access to XMLHttpRequest at 'https://i.connectad.io/api/v2' from origin 'https://www.w3schools.com' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.   tryit.asp:1
▶POST https://i.connectad.io/api/v2 net::ERR_FAILED 403   prebid.js?v=4759-1664277844090:9
```

Using 3rd Party Library

Purwadhika
Digital Technology School

There are several popular library to use for logging, for example Pino and Winston

Thank You!

Purwadhika
Digital Technology School