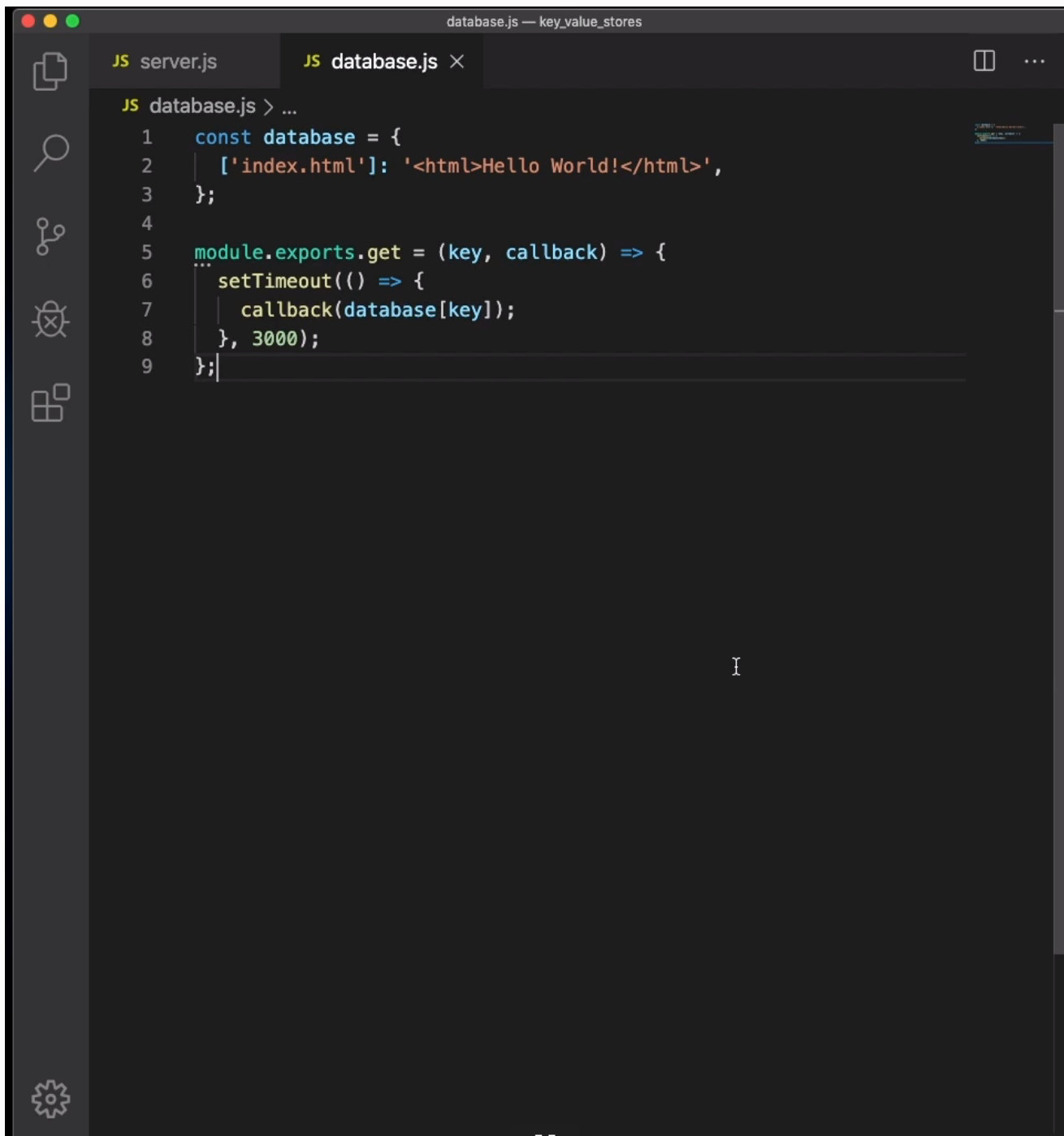


Key-Value Store




The screenshot shows a code editor window titled "database.js — key_value_stores". The editor has two tabs: "server.js" and "database.js". The "database.js" tab is active, showing the following JavaScript code:

```
JS database.js > ...  
1  const database = {  
2    ['index.html']: '<html>Hello World!</html>',  
3  };  
4  
5  module.exports.get = (key, callback) => {  
6    ...  
6    setTimeout(() => {  
7      callback(database[key]);  
8    }, 3000);  
9  };
```

The code defines a constant object named `database` with a single key-value pair: `['index.html']: '<html>Hello World!</html>'`. It also defines a `get` function on `module.exports` that takes a `key` and a `callback` as arguments. The function uses `setTimeout` to call the `callback` with the value from the `database` object after a 3000ms delay.

server.js — key_value_stores

JS server.js JS database.js

JS server.js >  app.get('/nocache/index.html') callback

```
1  const database = require('./database');
2  const express = require('express');
3  const redis = require('redis').createClient();
4
5  const app = express();
6
7  app.get('/nocache/index.html', (req, res) => {
8    database.get('index.html', page => {
9      res.send(page);
10    });
11  });
12
13  app.get('/withcache/index.html', (req, res) => {
14    redis.get('index.html', (err, redisRes) => {
15      if (redisRes) {
16        res.send(redisRes);
17        return;
18      }
19
20      database.get('index.html', page => {
21        redis.set('index.html', page, 'EX', 10);
22        res.send(page);
23      });
24    });
25  });
26
27  app.listen(3001, function() {
28    console.log('Listening on port 3001!');
29  });
```

05:

2 Prerequisites

Relational Database

A type of structured database in which data is stored following a tabular format; often supports powerful querying using SQL.

Non-Relational Database

In contrast with relational database (SQL databases), a type of database that is free of imposed, tabular-like structure. Non-relational databases are often referred to as NoSQL databases.

4 Key Terms

Key-Value Store

A Key-Value Store is a flexible NoSQL database that's often used for caching and dynamic configuration. Popular options include DynamoDB, Etcd, Redis, and ZooKeeper.

Etcd ⚡

Etcd is a strongly consistent and highly available key-value store that's often used to implement leader election in a system.

Redis ⚡

An in-memory key-value store. Does offer some persistent storage options but is typically used as a really fast, best-effort caching solution. Redis is also often used to implement **rate limiting**.

ZooKeeper ⚡

ZooKeeper is a strongly consistent, highly available key-value store. It's often used to store important configuration or to perform leader election.