

Project 1 Reflection

Team_2

Team members:

Yucheng An (989444449)

Henglay Eung (989468833)

Harsh Umesh Bhanushali (989467927)

Task 1:

Define an **IndexedDB** object store named "TodoList" and populate it with 100,000 randomly generated objects, each containing `task`, `status`, and `dueDate` properties and display it on the console or the browser.

We learned to populate large amounts of data (100k random objects) into IndexedDB with efficient use of transactions. Start from building DB, limit data format, data generated functions, to add objects into IndexedDB. We need to deal with how to prevent duplicated generating the same data when the page refreshes.

Task 1:

Set 1000 objects to status "completed" and the remaining ones to status "progress"

We learned updating statuses for 1000 tasks to "completed" and the rest to "pending" was successful. We faced some challenges on managing and keeping counters synchronized during cursor traversal required attention to avoid incorrect status assignments.

Task 2:

Measure and display the time (in milliseconds) required to read all objects with `status` set to "completed" on the console or the browser

We learned how to measure and display the time between one action to another action. Because of async and await functions, we cannot deal with measures in the normal way. We used function parameters to transfer the action we want to do such as assigning status, what status we want to set, decide different flags.

Task 3:

Apply a **read-only flag** to the object store and measure and display the time to read all completed tasks again on the console or the browser.

We learned to switch different modes between "readwrite" and "read-only" to access the data and re-set like task 3. To make code clear and readable, we design one function to pass parameters as different statuses. Read-only mode improves efficiency over "readwrite" mode in IndexedDB, especially in the field of large data processing.

Task 4:

Create an index on the `status` field, then measure and display the time to read all completed tasks on the console or the browser.

We learned we can create different index fields by using different parameters such as id, status, date, etc. This will significantly improve the efficiency for reading all status with 'completed'. We need to know before, the index must be created during the "onupgradeneeded" process of IndexedDB initialization. Without this, the index might not exist, leading to errors while running the code.

Task 5:

Define a new object store called "TodoListCompleted", copy all completed tasks from "TodoList" to this new store, and measure and display the time required to read all completed tasks from "TodoListCompleted" on the console or the browser.

We learned that we could use it to create a new store for some special cast in this example, we make a new one to save that task status are "completed". We may separate the object store for different states, dates, or names. This will make it more efficient and easier to structure of data while we are dealing with large datasets.

Key Insights

After finishing this project and previous labs related to IndexedDB, IndexedDB is efficient for large datasets based on the local browser but requires careful transaction management, and index setting, especially with cursor traversal. Correctly using the correct different transaction modes such as read-only, and readwrite in special casts can significantly improve performance. (We already compare by using this project). Proper indexing is essential for optimizing a lot of performance, and creating an index on the right field is crucial. Not only that, we can also re-organize data into separate object stores based on states or categories (like completed tasks, a special range of date, or more variables) can simplify querying and improve maintainability for large databases.

Screenshot

Database Management Project 1

Set up Database

Task 1

Task 2

Task 3

Task 4

Task 5

Database Management System 2 - Project - 1

Team members: Yucheng An, Henglay Eung, Harsh Umesh Bhanushali

Press F12, check result in Console part, check IndexedDB in Application part

Project has been submitted to: [GitHub](#)

© 2024 Database Management Systems 2 - Project Part 1: Data and Scalability

100k objects added.

1000 Status update completed.

Completed: 1000 In Progress: 99900 Pending: 0

Found status 'completed': 1000 By using method: readwrite !!!

Time took to read all objects with status "completed" with readwrite method: 581.70 ms

Found status 'completed': 1000 By using method: readonly !!!

Time took to read all objects with status "completed" with read-only method: 435.58 ms

Found status 'completed': 1000 By using index ('status')!!!

Time took to read all objects with status "completed" with indexing on "status": 69.68 ms

Found status 'completed': 1000 By using method: readonly !!!

Time took to read all objects with status "completed" with read-only method from TodoListCompleted: 31.20 ms

Index: 16:89

Index: 16:134

Index: 16:135

Index: 16:157

Index: 16:205

Index: 16:157

Index: 16:218

Index: 16:182

Index: 16:215

Index: 16:157

Index: 16:228

Database Management Project 1

Database Management System 2 - Project - 1

Team members: Yucheng An, Henglay Eung, Harsh Umesh Bhanushali

Press F12, check result in Console part, check IndexedDB in Application part

© 2024 Database Management Systems 2 Project Part 1: Data and Scalability

Application

Manifest

Service workers

Storage

Local storage

Session storage

IndexedDB

Project1DB

TodoList

id

status

TodoListCompleted

Private state tokens

Interest groups

Shared storage

Cache storage

Storage buckets

Background services

Backforward cache

Background fetch

Background sync

Bounce tracking mitigation

Notifications

Payment handler

Periodic background sync

Speculative loads

Push messaging

Reporting API

Frames

top

#

Key (Key)

Value

0

0

{id: 0, task: 'Update internal knowledge base', status: 'completed', dueDate: '2024-10-13'}
dueDate: '2024-10-13'
id: 0
status: 'completed'
task: 'Update internal knowledge base'

1

1

{id: 1, task: 'Review legal documents', status: 'completed', dueDate: '2024-10-22'}

2

2

{id: 2, task: 'Test new feature', status: 'completed', dueDate: '2024-10-20'}

3

3

{id: 3, task: 'Prepare financial report', status: 'completed', dueDate: '2024-10-18'}

4

4

{id: 4, task: 'Prepare presentation slides', status: 'completed', dueDate: '2024-10-22'}

5

5

{id: 5, task: 'Prepare financial report', status: 'completed', dueDate: '2024-10-15'}

6

6

{id: 6, task: 'Perform system integration tests', status: 'completed', dueDate: '2024-10-13'}

7

7

{id: 7, task: 'Attend team meeting', status: 'completed', dueDate: '2024-10-12'}

8

8

{id: 8, task: 'Perform system integration tests', status: 'completed', dueDate: '2024-10-21'}

9

9

{id: 9, task: 'Organize team-building event', status: 'completed', dueDate: '2024-10-20'}

10

10

{id: 10, task: 'Review technical documentation', status: 'completed', dueDate: '2024-10-20'}

11

11

{id: 11, task: 'Complete project plan', status: 'completed', dueDate: '2024-10-16'}

12

12

{id: 12, task: 'Optimize performance issues', status: 'completed', dueDate: '2024-10-20'}

13

13

{id: 13, task: 'Organize feedback from users', status: 'completed', dueDate: '2024-10-15'}

14

14

{id: 14, task: 'Conduct code review session', status: 'completed', dueDate: '2024-10-17'}

15

15

{id: 15, task: 'Research new software tools', status: 'completed', dueDate: '2024-10-18'}

16

16

{id: 16, task: 'Implement feedback from team', status: 'completed', dueDate: '2024-10-20'}

17

17

{id: 17, task: 'Check system backups', status: 'completed', dueDate: '2024-10-11'}

18

18

{id: 18, task: 'Perform database maintenance', status: 'completed', dueDate: '2024-10-17'}

19

19

{id: 19, task: 'Prepare financial report', status: 'completed', dueDate: '2024-10-13'}

20

20

{id: 20, task: 'Perform database maintenance', status: 'completed', dueDate: '2024-10-12'}

21

21

{id: 21, task: 'Write documentation', status: 'completed', dueDate: '2024-10-17'}

22

22

{id: 22, task: 'Check system backups', status: 'completed', dueDate: '2024-10-11'}

23

23

{id: 23, task: 'Schedule code deployment', status: 'completed', dueDate: '2024-10-12'}

24

24

{id: 24, task: 'Organize project files', status: 'completed', dueDate: '2024-10-15'}

25

25

{id: 25, task: 'Prepare meeting agenda', status: 'completed', dueDate: '2024-10-14'}

26

26

{id: 26, task: 'Analyze user feedback', status: 'completed', dueDate: '2024-10-20'}

27

27

{id: 27, task: 'Track project milestones', status: 'completed', dueDate: '2024-10-14'}

28

28

{id: 28, task: 'Review legal documents', status: 'completed', dueDate: '2024-10-15'}

29

29

{id: 29, task: 'Schedule code deployment', status: 'completed', dueDate: '2024-10-15'}

30

30

{id: 30, task: 'Test new feature', status: 'completed', dueDate: '2024-10-16'}

31

31

{id: 31, task: 'Review technical documentation', status: 'completed', dueDate: '2024-10-19'}

32

32

{id: 32, task: 'Prepare presentation slides', status: 'completed', dueDate: '2024-10-16'}

33

33

{id: 33, task: 'Update project roadmap', status: 'completed', dueDate: '2024-10-13'}

34

34

{id: 34, task: 'Prepare meeting agenda', status: 'completed', dueDate: '2024-10-21'}

Total entries: 100000

Files

main

Go to file

indexddb

labs

mongodb

neo4j

projects/project-1

TEAM 7

Team_1

Team_11

Team_12

Team_2

Team_5

project_1_team_2

index.html

index.js

project_1_team_2_sumeet_venkat

project_1_team_8

team- Shcharath Pavan Chalama...

team-6

team-griswa-makwana-nguyen

dms2 / projects / project-1 / project_1_team_2

Add file

...

Henglay-Eung

Change Team 2 directory

5a81c20 · 2 minutes ago

History

Name	Last commit message	Last commit date
..		
index.html	Change Team 2 directory	2 minutes ago
index.js	Change Team 2 directory	2 minutes ago