

## Implementation:

### Fetch Event

- You can track and manage page network traffic with this event.
- You can check existing cache, manage “cache first” and “network first” requests and return a response that you want.

```
self.addEventListener('fetch', function(event) {  
  console.log('fetch successfull');  
});
```

### Sync Event

- Background Sync is a Web API that is used to delay a process until the Internet connection is stable.
- We can adapt this definition to the real world; there is an e-mail client application that works on the browser and we want to send an email with this tool.
- Internet connection is broken while we are writing e-mail content and we didn't realize it.
- When completing the writing, we click the send button.

```
self.addEventListener('sync', function(event) {  
  console.log('Sync successfull');  
});
```

### Push Event

- This is the event that handles push notifications that are received from the server.
- You can apply any method with received data.

Sw.js file

```
self.addEventListener('install', function(event) {  
  console.log('service worker has been installed');  
});
```

```
self.addEventListener('activate', function(event) {  
  console.log('service worker has been activated');  
});
```

```
self.addEventListener('sync', function(event) {  
  console.log('Sync successfull');  
});
```

```
self.addEventListener('fetch', function(event) {  
  console.log('fetch successfull');  
});
```

```
self.addEventListener('push', (event) => {  
  const options = {  
    body: 'This notification was generated from a push!',  
    icon: "",  
    data: {  
      dateOfArrival: Date.now(),  
      primaryKey: '2'  
    },  
    actions: [{  
      action: 'explore',  
      title: 'Explore this new world',  
      icon: ""  
    },  
    {  
      action: 'close',  
      title: 'Close',  
      icon: ""  
    },  
  ]  
};  
event.waitUntil(  
  self.registration.showNotification('Title', options)  
)  
});
```

## Output

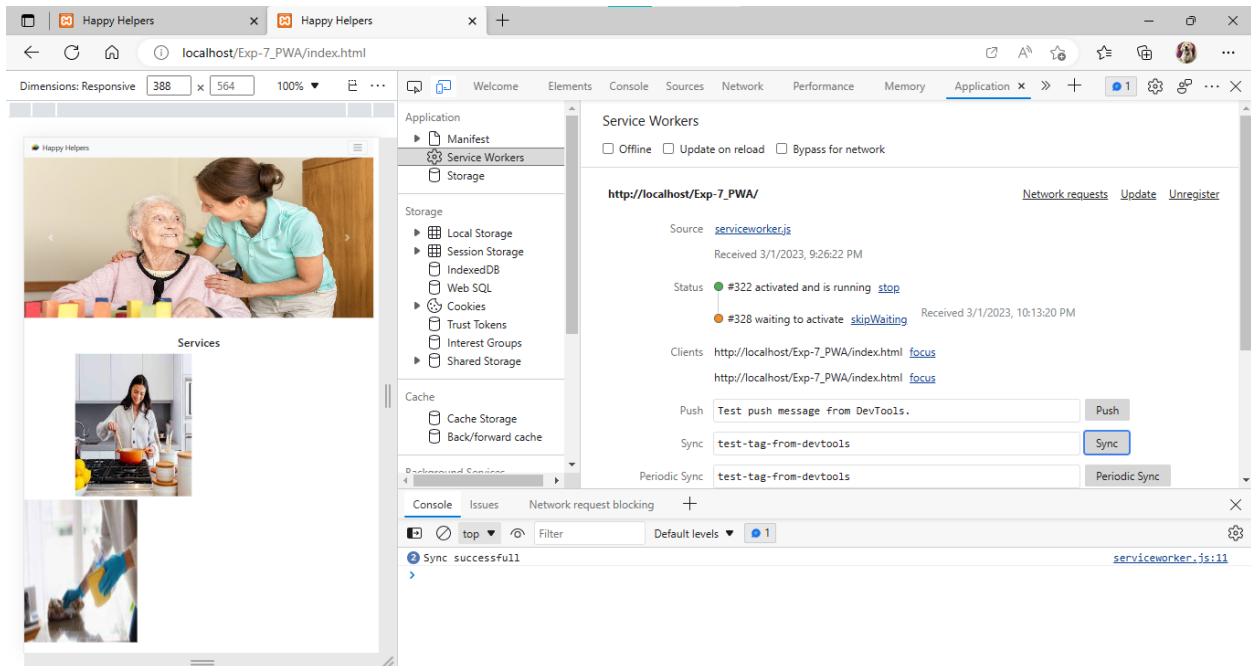
- **Fetch Event**

```
service worker registered ► ServiceWorkerRegistration
service worker has been installed
3 fetch successfull
> |
```

- **Push Event**

The screenshot shows the Chrome DevTools Application panel with the Service Workers section selected. The browser window displays a page titled "Happy Helpers" with a video of an elderly woman being assisted by a caregiver. The Service Workers panel shows a single worker for the origin `http://localhost/Exp-7_PWA/` with source `serviceworker.js`. The worker's status is "activated and is running" (green dot). Below the status, there are input fields for "Push" (containing "Test push message from DevTools.") and "Sync" (containing "test-tag-from-devtools"), each with a corresponding button. The "Update Cycle" section shows a timeline for version #322, with states "Install", "Wait", and "Activate". The "Background Services" section lists various background features like Background Fetch, Background Sync, Notifications, etc.

- Sync Event



**Conclusion:** In this experiment, we studied fetch, sync and push events. We can track and manage page network traffic with this event. Background Sync is a Web API that is used to delay a process until the Internet connection is stable. Push event handles push notifications that are received from the server. We successfully implemented fetch, sync and push events work for Service Workers.