

# Caching Dynamic Content

---



**Jad Joubran**

WEB CONSULTANT

@joubranjad [www.jadjoubran.io](http://www.jadjoubran.io)

# Overview

**Precache vs. Dynamic Content**

**Workbox Dynamic Routing**

- Caching Strategies
- Storage Optimization

# Precache vs. Dynamic Content

## **Precached Assets**

**App Shell**

**Cached Beforehand**

**Static**

## **Dynamic Content**

**API Calls**

**Cached on Demand**

**Depends on User Actions**

# Workbox Caching Strategies

**Network First**

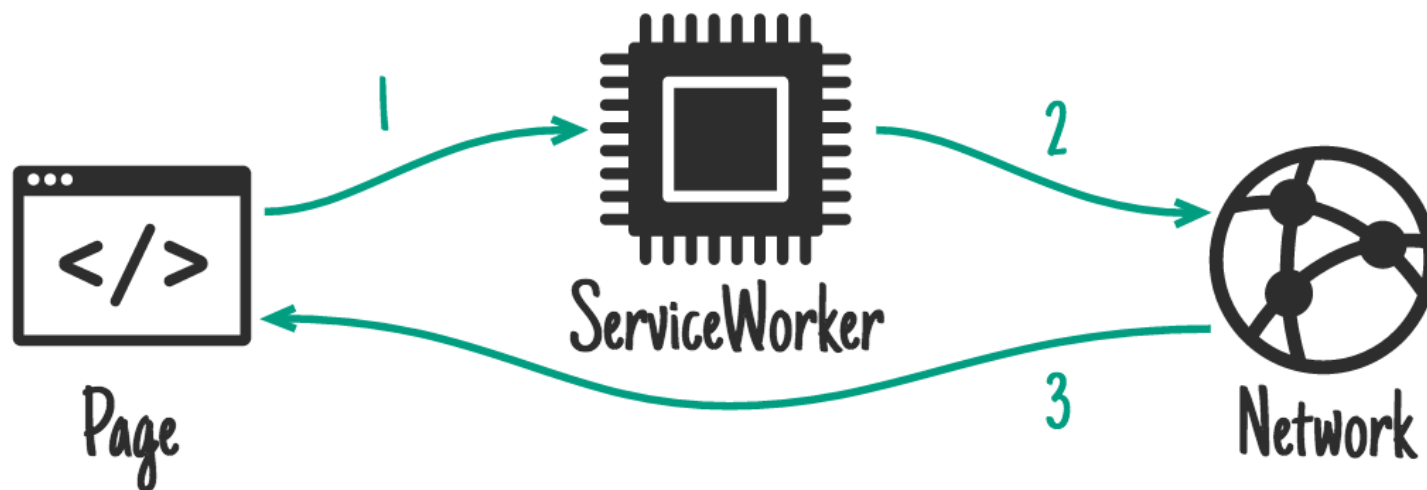
**Cache First**

**Stale While  
Revalidate**

**Network Only**

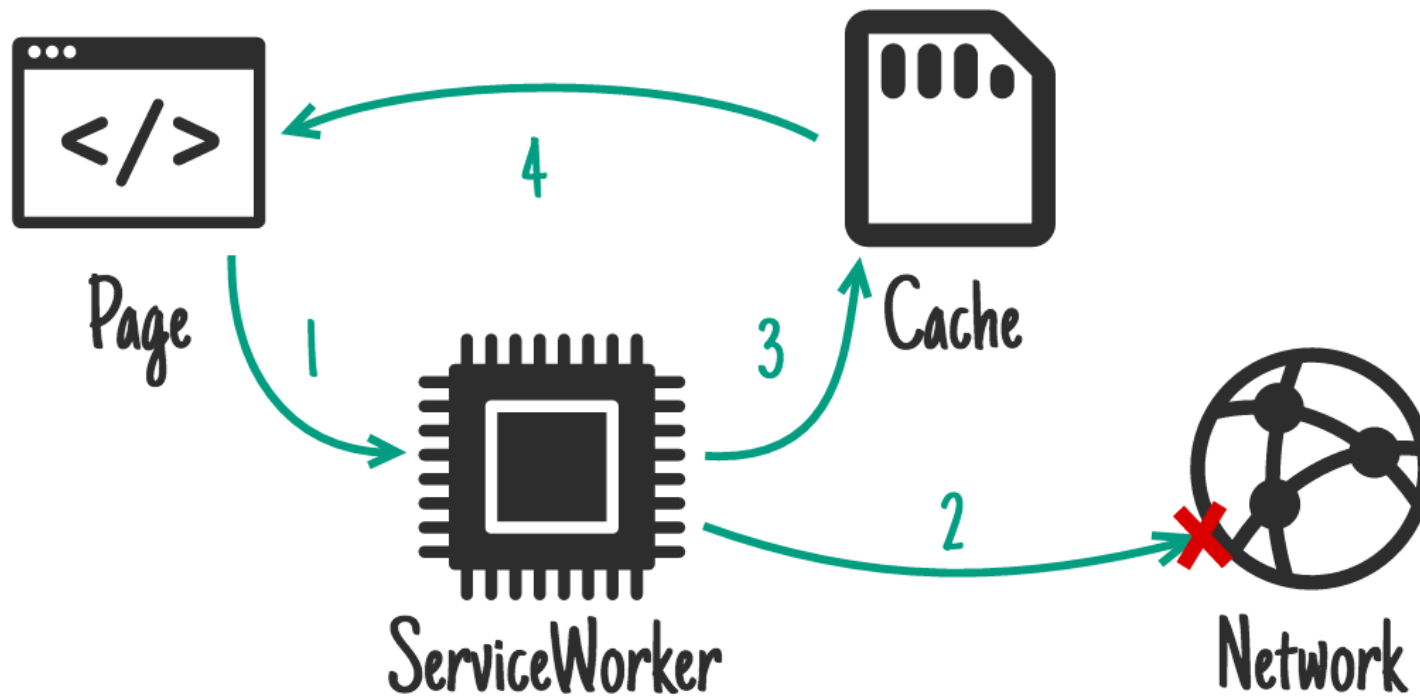
**Cache Only**

## Network Only



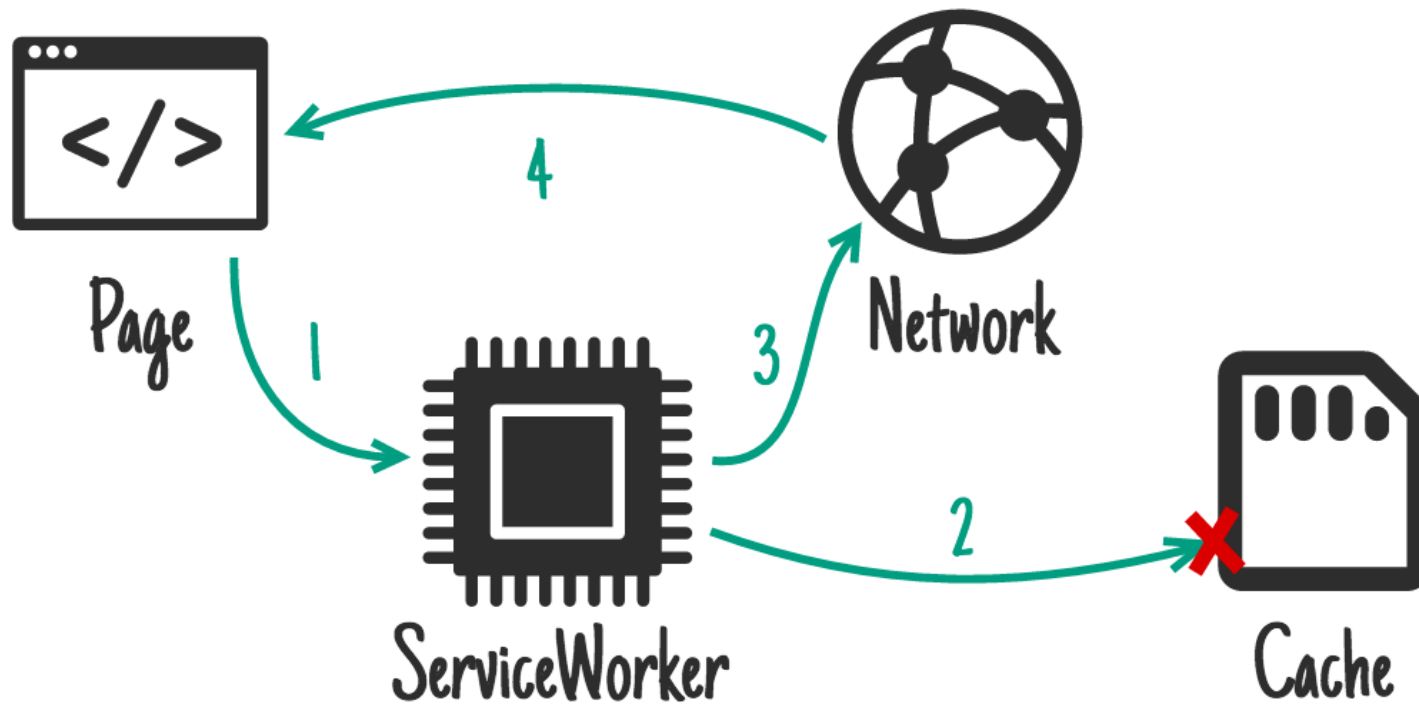
Source: [Offline Cookbook | Web Fundamentals](#)

## Network First



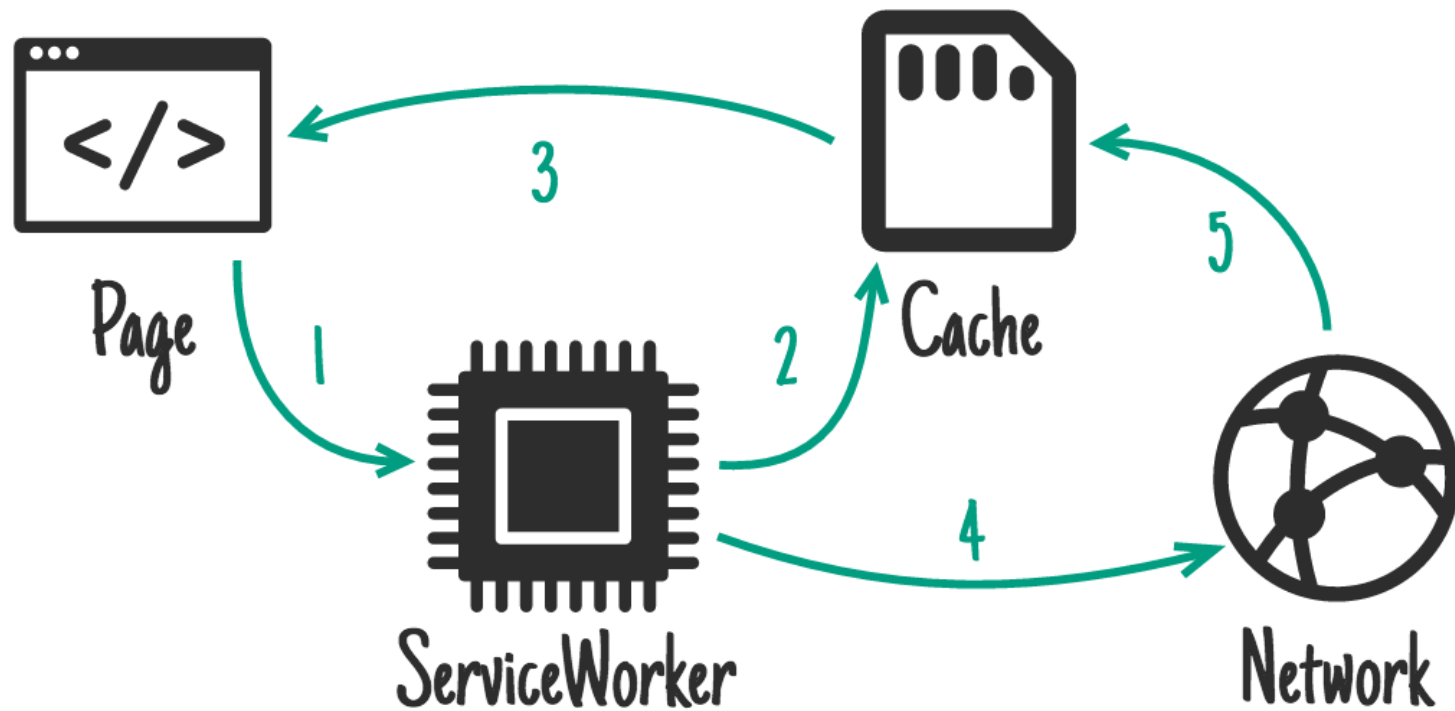
Source: [Offline Cookbook | Web Fundamentals](#)

## Cache First



Source: [Offline Cookbook | Web Fundamentals](#)

# Stale While Revalidate



Source: [Offline Cookbook | Web Fundamentals](#)



# Strategies Depend on Your Use Case



**Can you show stale content?**



**Can you prefer fresh data but fallback to cache?**



**You can only show offline data for a limited duration?**

```
workbox.routing.registerRoute(  
  /api\users\.json/  
  new workbox.strategies.NetworkFirst()  
)
```

Associate Request Path with Caching Strategy

# Workbox Strategies

```
new workbox.strategies.NetworkFirst()  
new workbox.strategies.CacheFirst()  
new workbox.strategies.StaleWhileRevalidate()
```

# Same vs. Different Domain

## Same Domain

Regex should match file name

Example: data.json

## Different Domain

Regex must match file name & full domain

Example: <https://example.com/data.json>

# Demo

**Fetch list of flights**

**Dynamic caching for flights**

# Storage Space



**Storage is not unlimited**



**Be mindful of user's storage**



**Storage space depends on user's device**



**Storage space depends on user's browser**

```
workbox.routing.registerRoute(  
  /api\/.*\.json$/,  
  new workbox.strategies.NetworkFirst({  
    cacheName: "api-cache"  
  })  
)
```

## Name the Cache

**Required step for managing expiration**

## Expire Entries

```
workbox.routing.registerRoute(  
  /api\/.*\.json$/,  
  new workbox.strategies.NetworkFirst({  
    cacheName: "api-cache",  
    plugins: [  
      new workbox.expiration.Plugin({  
        maxEntries: 50,  
        maxAgeSeconds: 10 * 86400, // 10 Days  
        purgeOnQuotaError: true  
      })  
    ]  
  })  
)
```



# Demo

## **Set expiration constraints**

- Name the cache
- Setup the expiration plugin

## **Test expiration in the browser**

# Summary

**Precache vs. Dynamic Content**

**Workbox Dynamic Caching Strategies**

**Storage Optimization**