Atitit pwa fet缓存白名单机制

// {ignoreVary: true} igore head,,only need url match..  
// caches.match(event.request) caches.match(event.request, {ignoreVary: true})  
// it can be empty if you just want to get rid of that error  
// caches.match(event.request.url) for post  
// jeig must beir page does-not-work-offline-error  
this.addEventListener('fetch', function (event) {  
  
 // console.log('fetchEvtFun( event.request.url---' + event.request.url);  
 ***console***.log("fetch.url:: " + event.request.url)  
 //font(woff), css white list  
 if (endWith\_cachelist(event.request.url, ***cacheList***)) {  
 //can cache block cacheList,,net first  
 ***console***.log(" end with cachelist whitelist ,can cache " + event.request.url)  
 } else if (event.request.url.startWith('chrome-extension')) {  
 //need jar,,yva sh exteng.png file ,,alway fetch  
 ***console***.log(" chrome-extension ,not need cache ,fetch alwas ")  
 return fetch(event.request); //fet from url..  
 } else if (endWithx(event.request.url, ".css,.woff,.jpg,.png")) {  
 //can cache block ,,cache fist  
 ***console***.log(" end with css jpg ,can cache " + event.request.url)  
 } else {  
 //fet alway block  
 ***console***.log(" not static file, ,alswa fetch new" + event.request.url);  
 return fetch(event.request); //fet from url..  
  
 }  
  
  
 //def tsalue ,,net first  
 return event.respondWith(  
 fetch(event.request).then(response =>  
 ***caches***.open(***cacheStoK***).then(cacheObj1 => {  
 cacheObj1.put(event.request.url, response.clone());  
 return response;  
 })  
 ).catch(() => {  
 return ***caches***.match(event.request.url);  
 })  
 );  
 // return event.respondWith(caches.match(event.request.url).then(response => fetchEvtFun(event, response)));  
});