Atitit pwa实施流程

<!-- pwa -->  
  
  
<link rel="manifest" href="/manifest.json">  
<script>***navigator***.serviceWorker.register('/service-worker.js')</script>  
<script>  
 ***console***.log("before next");  
</script>  
  
  
<!-- pwa end-->

## Manifest.json设置icon

## service-worker.js开启离线cache模式

//die()...  
// "display": "minimal-ui", ios cant suppt this meta,,need ios meta add  
*/\*\*  
 \* <meta name="apple-mobile-web-app-capable" content="yes">  
 \** ***@type*** *{string}  
 \*/*var ***cacheStoK*** = 'stokey-89789'  
 var ***cacheList\_iniIntall*** = [  
  
 "/",  
  
 "/index.php/index/index.html",  
  
 "/index.php/vod/search/by/time\_add.html",  
   
  
]  
bekcalst = [];  
alwasNetList = ["/login/chklogin.php", "/login/logout.php","/login/logoutBef.php"];  
***self***.addEventListener('install', e => {  
 ***console***.log(" self.install evt....")  
 e.waitUntil(  
 ***caches***.open(***cacheStoK***)  
 .then(cache => cache.addAll(***cacheList\_iniIntall***))  
 .then(() => ***self***.skipWaiting())  
 )  
 ***console***.log(" self.install finish ..")  
});  
  
***String***.prototype.startWith = function (str) {  
 var reg = new ***RegExp***("^" + str);  
 return reg.test(this);  
}  
  
//测试ok，直接使用str.endWith("abc")方式调用即可  
//有bug ，如果包含特殊符号，比如url  
***String***.prototype.endWith = function (str) {  
 var reg = new ***RegExp***(str + "$");  
 return reg.test(this);  
}  
  
function endWith(str, suffix) {  
 return str.indexOf(suffix, str.length - suffix.length) !== -1;  
}  
  
function startWithx(url2, needX) {  
 var url = url2;  
 var arr = needX.split(",");  
 for (const v of arr) {  
 if (url.startWith(v))  
 return true;  
 }  
 return false;  
}  
  
function endWith\_NetAwyslist(url, NetAwyslist) {  
 // var url = url2;  
// var arr = needX.split(",");  
 for (const cached\_url of NetAwyslist) {  
 // console.log('url.endWith(cached\_url):' + url + "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + cached\_url)  
 // console.log(url)  
 // console.log(cached\_url)  
 if (endWith(url, cached\_url)) {  
 // console.log("compRzt true")  
  
 return true;  
 } else {  
 // console.log("compRzt false")  
 }  
 // console.log('url.endWith(cached\_url)==> ( '+url+":"+ cached\_url + " ,rzt:"+);  
 }  
 return false;  
}  
  
function endWithx(url2, needX) {  
 var url = url2;  
 var arr = needX.split(",");  
 for (const v of arr) {  
 if (url.endWith(v))  
 return true;  
 }  
 return false;  
}  
  
  
// {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)  
  
  
 // *todo fetchAwasysList* if (endWith\_NetAwyslist(event.request.url, alwasNetList)) {  
 ***console***.log(" fetch alwas ", event.request.url)  
 return fetch(event.request); //fet from 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 {  
 //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 => {  
 //must cache then ret,,beir..asyn said reposen already use  
 cacheObj1.put(event.request.url, response.clone());  
 return response;  
 })  
 ).catch(() => {  
 return ***caches***.match(event.request.url);  
 })  
 //fetch end  
 );  
 //end rspwz  
  
});  
  
  
// hamyar d not must  
*/\*\*  
 \* 错误监控  
 SW 运行在 worker 线程里，其抛出的错误在页面是捕捉不到的，因此需要在 SW 里引入错误监控。（感谢题叶老师提供错误监控 SDK 的 SW 版本）  
  
  
 \*/  
  
/\*\*  
 \* self.addEventListener('error', event => {  
 // 上报错误信息  
 // 常用的属性：  
 // event.message  
 // event.filename  
 // event.lineno  
 // event.colno  
 // event.error.stack  
 console.log('sw error lstnr:' + JSON.stringify(event) );  
})  
 self.addEventListener('unhandledrejection', event => {  
 // 上报错误信息  
 // 常用的属性：  
 // event.reason  
 console.log('sw unhandledrejection lstnr:' + JSON.stringify(event));  
})  
 \*/  
  
  
/\*\* 一般动态不存缓存里面去取 部分可以cache  
 \* self.addEventListener('fetch', function(event) {  
 event.respondWith(  
 caches.match(event.request)  
 .then(function(response) {  
 // Cache hit - return response  
 if (response) {  
 return response;  
 }  
 return fetch(event.request);  
 }  
 )  
 );  
});  
 \*/*function randomNotification() {  
 var randomItem = ***Math***.floor(***Math***.random() \* games.length);  
 var notifTitle = games[randomItem].name;  
 var notifBody = 'Created by ' + games[randomItem].author + '.';  
 var notifImg = 'data/img/' + games[randomItem].slug + '.jpg';  
 var options = {  
 body: notifBody,  
 icon: notifImg  
 }  
 var notif = new ***Notification***(notifTitle, options);  
 setTimeout(randomNotification, 30000);  
}  
  
  
function endWith\_cachelist(url, cacheList) {  
 // var url = url2;  
// var arr = needX.split(",");  
 for (const cached\_url of cacheList) {  
 // console.log('url.endWith(cached\_url):' + url + "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + cached\_url)  
 // console.log(url)  
 // console.log(cached\_url)  
 if (endWith(url, cached\_url)) {  
 // console.log("compRzt true")  
  
 return true;  
 } else {  
 // console.log("compRzt false")  
 }  
 // console.log('url.endWith(cached\_url)==> ( '+url+":"+ cached\_url + " ,rzt:"+);  
 }  
 return false;  
}