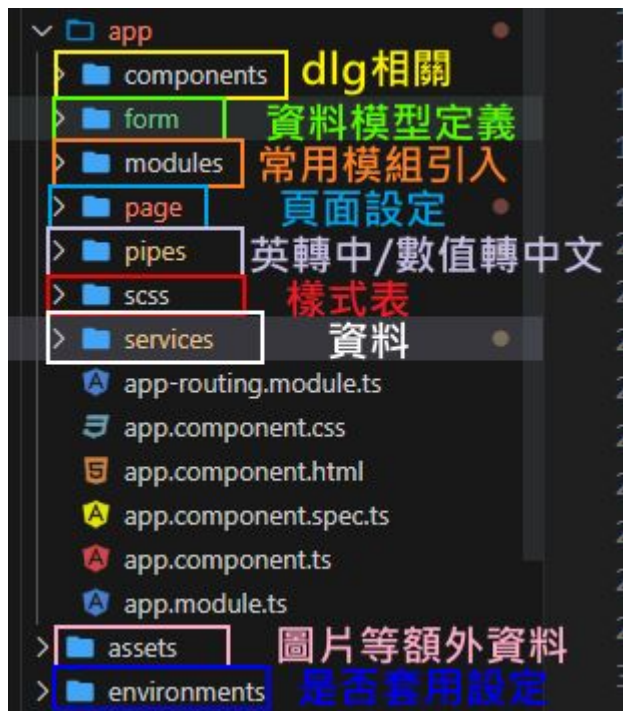

新版專案

報告者: 林建霖

(bbd)app下資料夾作用



- components裡面是放dlg彈跳視窗，但這邊的僅給bbd使用，基本上常用的在tpc專案中
- form是放資料模型的地方
- modules則是一些常用模組就先引用到component中，appModule再直接引用componentModule
- page則是頁面功能和顯示的部分，裡面會依照功能劃分不同資料夾，和form的資料可以對應

每當執行路由時都會觸發event

每次使用routerLink, navigateByUrl() or navigate()

都會觸發一個route.event

event的種類有十五種

這邊要偵測的則是

NavigationStart

filter(event =>

event instanceof

NavigationStart)

這一段程式碼是過濾事件用

在路由改變時會有觸發。

這邊用來檢查跟儲存標籤分頁

```
// pipe + filter 來過濾路由的事件
this.router_subscription = this.router.events.pipe(
  // NavigationStart 用來捕捉切換路由
  filter(event => event instanceof NavigationStart)
).subscribe((res: any) => {
  console.log(" 查詢頁 QuerySuper - ", this.getWorktask());
  console.log(res);
  if (this.getWorktask() != null) {
    // 如果分頁標籤不是空的，就先備份起來
    this.setWorktask().backupsdata[this.getWorktask().time] = {
      datasource: {
        pageindex: this.datasource.pageindex,
        hasMore: this.datasource.hasMore,
        // 將資料以物件形式儲存
        data: JSON.parse(JSON.stringify(this.datasource.data))
      },
      queryoptions: {
        // 將資料以物件形式儲存
        options: JSON.parse(JSON.stringify(this.queryoptions.options))
      },
      filterSortName: this.filterSortName,
      // 分頁前面的icon
      filterSortIcon: this.filterSortIcon,
      locationInputValue: this.locationInputValue
    };
    // 將資料儲存到session裡面
    sessionStorage.setItem('worktaskdatas', JSON.stringify(this.worktaskdataSvc.worktaskdatas))
  }
});
```

worktaskdataSvc: 管理新專案上方的分頁

[首頁](#)[操作功能設定\(查詢\) X](#)[操作功能設定\(新增\) X](#)[角色設定\(查詢\) X](#)[角色設定\(新增\) X](#)

getWorktask 是取得標籤內容

setWorktask 則是會將內容存到worktaskdatas session storage裡面

▼ Session storage	role	user
http://localhost:30108	storename	
IndexedDB	yfscatalog	[[{"_typename": "MgtCatalogQry", "pk": "JQK8XclLu9uaSIUj", "stgb":
Web SQL	allprgs	[[{"progid": "JAC01E060", "route": null, "actions": "A;M;D"}, {"progid":
▶ Cookie	entacctname	海博
Private state tokens	rolename	CACOADMIN
Interest groups	empname	
▶ Shared storage	worktaskdatas	[[{"name": "首頁", "url": "/", "param": "", "icon": "fa-solid fa-house", "nc
Cache storage	progs	[/gws/works/{id}:"263D6B8327A74C89B3A404B4", "listseqno":
	loading	false
background services	jwt	eyJ0eXAiOiJV1QILICJhbGciOiJIUzI1NiJ9.eyJ1cG4iOiI0OTDRk1O0
☐ 上一步/下一步快取	formtitle	銀行
↕ Background fetch	companyname	
↻ Background sync	userid	4DB5937A8ACC4F55A7F1EF04
☐ Bounce tracking mitigation	logintime	上午 08:47 2023/10/12
🔔 通知	websitename	本地測試erp
☑ Payment handler	depname	
🕒 Periodic background sync	worksite	910F00015T
📬 Push messaging	username	管理員
📄 回報 API	whsname	
	managewebsitename	
❗ 先載入	▼ [{"name": "首頁", "url": "/", "param": "", "icon": "fa-solid fa-house", "node": null, "type": "", "pk": "", "-"}]	
↕ 猜測規則	▶ 0: {name: "首頁", url: "/", param: "", icon: "fa-solid fa-house", node: null, type: "", pk: ""}	
↕ 預先載入	▶ 1: {id: "查詢", name: "銀行", url: "/bcb/bank", param: "time=1697071668508", "-}	
↕ 這個頁面	▶ 2: {id: "T1", name: "銀行", url: "/bcb/bank/6PB2PTy8pGfQcRrk", param: "amb=b&time=1697074145445", "-}	
	▶ 3: {id: "17T", name: "銀行", url: "/bcb/bank/X0oKkULMeJNHIPfI", param: "amb=b&time=1697077439917", "-}	
	▶ 4: {id: "ZXCZ", name: "銀行", url: "/bcb/bank/RSqtJQXS3iHdq0Er", param: "amb=b&time=1697077539339", "-}	

QuerySuper內的方法大致說明

- `getWorktask();` // 取得分頁標籤內容
- `setWorktask();` // 新增分頁內容到session, 並取得分頁標籤內容
- `loadWorktaskData();` // 讀取標籤的內容, 若有備份有則讀取內容
- `keyEvent();` // 處理鍵盤事件
- `doQuery();` // 用於graphql查詢
- `addcol(cols: QueryColType[]);` // 設定AG-grid欄位

QuerySuper內的方法大致說明

- `chooseFilterSort();` // 左上排序選擇後重新排序, 同步表單排序
- `gridSort();` // 點擊表單的排序, 同步左上排序
- `dataLocations();` // 排序設定欄位內容, 定位資料用
- `gridAutoExcel(defs: ColType[], worksheet: Worksheet, entity: any[]);`
// 匯出excel
- `getRectype(val: any);` // 取得狀態, 0草稿, 1確認, 9作廢
- `setColumnSort();` // 設定

entitySuper

會設定WorkTask type以及id

然後檢查是否機會已經有新增的
接著使用loadEntity中設定的
路徑呼叫api
成功則將資料填充上去

```
// 有沒有pk，沒有就是新增，有就是編輯
```

```
setTimeout(() => {  
  if (this.pk == "null") {  
    this.createEntity();  
  } else {  
    this.loadEntity();  
  }  
}
```

```
// 設定上一筆下一筆的內容
```

```
this.setUrlParamData();  
}, 200);
```

```
loadEntity() {  
  this.loadingService.startLoading();  
  if (this.backupsEntity()) {  
    this.loadingService.stopLoading();  
    return;  
  }  
  this.httpProxyService.RxCreateJwtJsonGet<EventResultT<E>>(this.getApiPath() + '/' + this.pk).subscribe({  
    next: (res: any) => {  
      if (res.success) {  
        // 分頁標籤  
        // 推測這邊是給gws, work 待辦事項使用的(關係記錄唯碼、交易單別)，先保留不看  
        if (this.getWorktask().type == '' && this.router.url.indexOf(this.pk + "/" ) == -1) {  
          this.setWorktask().type = 'edit';  
          if (this.titlelevel2key && !this.titlelevel3key) {  
            this.setWorktask().id = res.data[this.titlekey]?.[this.titlelevel2key];  
          } else if (this.titlelevel2key && this.titlelevel3key) {  
            this.setWorktask().id = res.data[this.titlekey]?.[this.titlelevel2key]?.[this.titlelevel3key];  
          } else {  
            this.setWorktask().id = res.data[this.titlekey];  
          }  
        }  
        // 填充資料  
        this.fillData(res.data);  
      }  
    }  
  });  
}
```

fillData功能是将
資料填回去options
並依照amb
開放或關閉欄位

```
if (this.amb == "a" || this.amb == "m")  
  this.options.enable()  
else  
  this.options.disable()  
  
fillData(data: E) {  
  let ds = JSON.parse(JSON.stringify(data));  
  console.info(data);  
  console.info(this.amb);  
  if (this.amb == "a" || this.amb == "m")  
    this.options.enable();  
  else  
    this.options.disable();  
  this.patchValueToForm(ds);  
}
```

按鈕對應的方法

query

toNew()

跳到新增頁面



toExit()

離開這個標籤



toEdit(row: RowSelectedEvent, field: string = "")

取得該表格內容, 進入編輯頁

草稿	9999999	117711117777	2023-01-17	2099-12-31
----	---------	--------------	------------	------------

gridAutoExcel(defs: ColType[], worksheet: Worksheet, entity: any[])

匯出excel



按鈕對應的方法

entity

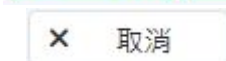
toNew()



doEdit()



doCancel()



doExit()



doSave()



doDraft()



doDelete()



doBan()



doUndo(event: string)



新增按鈕使用出現

編輯後，需要先使用還原按鈕才會有刪除跟作廢

新專案練習銀行

使用i18n轉換

🏠 首頁 銀行(查詢) ✕ 銀行(新增) ✕ 銀行(17T) ✕ 銀行(9999999) ✕ > ✕

+ 新增 ▼ 🔍 進階查詢 → 離開

↑ 通匯銀行編號 ▼

狀態	通匯銀行編號 ↑	銀行簡稱	啟用日期	停用日期
確認	0123456	測試TEST	2023-10-04	2099-12-31
草稿	123	123456	2023-10-04	2099-12-31
確認	17T	17TEST11111	2023-01-17	2099-12-31
草稿	9999999	117711117777	2023-01-17	2099-12-31
確認	T1	test1aaa	2022-10-21	2099-12-31
確認	TEST0425	TEST0425TEST0425	2023-04-25	2099-12-31

🏠 首頁 銀行(查詢) ✕ 銀行(新增) ✕ 銀行(17T) ✕ 銀行(9999999) ✕ > ✕

📄 確認儲存(S) ✕ 取消

基本訊息

確認 —

通匯銀行編號	17T	銀行管理碼*	111
銀行簡稱	17TEST	銀行全稱	11111
英文全稱	17T	英文簡稱	測試青蛙
啟用日期	2023-01-17	停用日期	2099-12-31
		啟用否	<input checked="" type="checkbox"/>

進階設定

確認 —

啟用否	<input checked="" type="checkbox"/>	隸屬銀行	9999999 117711117777 🔍	銀行類別*	
ATM代碼	17T	Swift NO.	17T		
網址	WW				
email	TEST1234@GMAIL.COM				

設定i18n、欄位寬度

I18N

幫忙完成語系的轉換
位置放在boot中的
app/assets/i18n 資料夾中

由bbd的page下的分類
bcb裡面的
bankaccts entity/query
對應到這邊
bcb裡面的名稱

✓ assets	2407	"bcb": {
> ckeditor	2408	"title": "工作日曆",
> css	2409	"bankaccts": {
> file	2410	"title": "銀行帳戶",
> fileexplorer	2411	"isworkflow": "審批否",
✓ i18n	2412	"cmp": "公司",
{.} zh-CN.json	2413	"bankaccts": "帳戶號碼",
{.} zh-TW.json M	2414	"abbrname": "帳戶簡稱",
> images	2415	"fullname": "帳戶全稱",
> stimulsoft	2416	"bank4par": "銀行編號",
.gitkeep	2417	"opendate": "開戶日期",
{.} mf.manifest.json	2418	"edate": "停用日期",
> environments	2419	"ismajor": "主要否",
TS bootstrap.ts	2420	"isactive": "啟用否",
favicon.ico	2421	"rdate": "立案日期",
index.html	2422	"bdate": "啟用日期",
TS main.ts	2423	"appddate": "批准日期",
polyfills.ts	2424	"bankacno": "管理帳戶",
styles.css	2425	"curr": "幣別"
TS test.ts	2426	},
TS typings.d.ts		
.editorconfig		

使用dev管理系統查看form以及i18n對應

<http://localhost:30108/dev/dmsschemas>

生產環境專用

選擇: BankEntity - bbd030

選擇要產生的部分，後端會給資料的型態定義，只有entity需要

複製

複製語系

```
export class UserRef {  
  pk: string;  
  id: string;  
  name: string;  
}
```

生產環境專用

選擇: BankEntity - bbd030

點複製語系則會給i18n對應

複製

複製語系

```
"BankEntity": {  
  "pk": "資料唯一key",  
  "id": "代碼",  
  "name": "簡稱",  
  "bankid": "銀行編號",  
  "abbrname": "銀行簡稱",  
  "fullname": "銀行全稱",  
  "ver": "ver",  
  "aliasname": "aliasname",  
  "firstname": "firstname",  
  "lastname": "lastname",  
  "middlename": "middlename",  
  "createuser": "創建人",  
  "createdate": "創建日期",  
  "updateuser": "更新人",  
  "updatedate": "更新日期",  
  "rectype": "資料狀態",  
  "cstatusuid": "狀態",  
  "isworkflow": "工作流狀態",  
  "bankabbrcode": "銀行管理碼",  
  "isheadquarters": "總部否",  
  "bank4par": "隸屬銀行",  
  "swiftno": "Swift NO.",  
  "bankatmno": "ATM代碼",  
  "homepage": "網址",  
  "email": "email",  
  "isactive": "是否啟用",  
  "bdate": "bdate",  
  "edate": "edate",  
  "twname": "繁體名稱",  
  "usname": "英文名稱"  
}
```

bankquery

```
super(httpProxyService, router, activatedRoute, worktaskdataSvc, changeDetecter);
this.title = "bcb.bank.title";
this._querygrid = this.querygrid;
// grid設定
this.addcol([
  { field: 'rectype', maxWidth: 80, minWidth: 20, headerName: '狀態', valueGetter: (data) => data[rectype] },
  { field: 'bankid', maxWidth: 250, minWidth: 70, headerName: this.translate.instant('bcb.bank.bankid'), valueGetter: (data) => data[bankid] },
  { field: 'abbrname', maxWidth: 250, minWidth: 70, headerName: this.translate.instant('bcb.bank.abbrname'), valueGetter: (data) => data[abbrname] },
  { field: 'bdate', maxWidth: 150, minWidth: 70, headerName: this.translate.instant('bcb.bank.bdate'), valueGetter: (data) => data[bdate] },
  { field: 'edate', headerName: this.translate.instant('bcb.bank.edate'), valueGetter: (data) => data[edate] },
]);
// graphql查詢路徑
this.datasource = new GraphQLDataSource("/bcb030/graphql", "banklist");
}
```

setoptions 是設定進階查詢的地方
this.translate.instant可以將英文轉換成中文

querySuper類都**必要實作query方法**
addField是設定要查詢的欄位要那些
toFilter是設定查詢的條件

addcol用來設定ag-grid的欄位
GraphQLDataSource設定graphql查詢路徑

```
super.ngOnInit();
console.log(this.datasource);

// 設定進階查詢的部分
if (this.queryoptions.options.length == 0)
  this.queryoptions.setoptions([
    { field: 'pk', value: '' },
  ]);
}

// 繼承querySuper需要實作的地方，用來做graphql查詢
public query(event) {
  this.datasource.clear();
  this.datasource.expFilter(event);

  // 設定查詢欄位
  this.datasource.addField("pk", "bankid", "abbrname", "bdate", "edate", 'rectype');

  this.queryoptions.toFilter().forEach(filter => {
    this.datasource.addFilter(filter.field, filter.operation, filter.value);
  });
  this.doQuery();
}
```

```
export abstract class QuerySuper {
  public abstract query(event: any): void;
}
```


bankquery

hb-querylayout:
標籤分頁下方的整個區塊

hb-querybar:
按鈕、匯出等功能的查詢功能列

hb-querysearchbar:
搜尋框

hb-querygrid:
下方ag-grid顯示表單內容

hb-querypager:
首頁、上下頁的分頁按鈕

```
<hb-querylayout [queryBarTemplateRef]="queryBar" [queryStatusBarTemplateRef]="queryStatusBar"
  [queryTableTemplateRef]="queryTable" [queryPagerTemplateRef]="queryPager">
</hb-querylayout>

<ng-template #queryBar>
  <hb-querybar #querybar [title]="title" [sreachinput]="locationInputValue" [filterOptions]="filterOpti
    [filterbttns]="queryFilterBttns" [filterSortName]="filterSortName" [filterSortIcon]="filterSortIcon"
    (toNewEvent)="toNew()" (toExitEvent)="toExit()" (toQueryEvent)="query(null)"
    (dataLocationsEvent)="dataLocations($event)" (toFilterEvent)="querysearchbar.changeShowFilter()"
    (toExportEvent)="export()"></hb-querybar>
</ng-template>

<ng-template #queryStatusBar>
  <hb-querysearchbar #querysearchbar [(queryoptions)]="queryoptions.options" (queryEvent)="query(null)"
  </hb-querysearchbar>
</ng-template>

<ng-template #queryTable>
  <div style="height: calc(100% - 100px);">
    <hb-querygrid #querygrid style="height: 100%;" [columns]="columns" [gridData]="datasource.data"
      rowHeightMode='normal' (queryRowSelect)="toEdit($event)" (sortChanged)="gridSort($event)">
    </hb-querygrid>
  </div>
</ng-template>

<ng-template #queryPager>
  <hb-querypager (pageEvent)="query($event)" [hasMore]="datasource.hasMore" [pageIndex]="datasource.pag
  </hb-querypager>
</ng-template>
```

bankentity

```
bankDialogFormGroup!: BankEntity.BankEntityFormGroup;

getFormGroup(): BankEntity.BankEntityFormGroup {
  let formGroup = BankEntity.BankEntityFormGroup.create();
  return formGroup;
}

// 放api路徑
getApiPath(): string {
  return "/bbd030/bcb/bank"
}
```

在繼承editSuper的時候
getApiPath()、getFormGorup()為必要實做內容

在form有定義需要使用到的資料模型
這邊就需要使用到所用到的 FormGroup
來操作要使用到的 FormControl

ApiPath則需要放入要去後端讀取的 api路徑

```
static create(){
  return new BankEntityFormGroup({
    ver: new FormControl(0,{ updateOn: "blur" }),
    pk: new FormControl(null,{ updateOn: "blur" }),
    createuser: new FormControl(null,{ updateOn: "blur" }),
    createdate: new FormControl(null,{ updateOn: "blur" }),
    updateuser: new FormControl(null,{ updateOn: "blur" }),
    updatedate: new FormControl(null,{ updateOn: "blur" }),
    rectype: new FormControl(null,{ updateOn: "blur",validators:[Validators.required]}),
    cstatusuid: new FormControl(null,{ updateOn: "blur" }),
    isworkflow: new FormControl(false,{ updateOn: "change",validators:[Validators.required]}),
    bankabbrcode: new FormControl(null,{ updateOn: "blur",validators:[Validators.maxLength(24), Validator
    abbrname: new FormControl(null,{ updateOn: "blur",validators:[Validators.maxLength(20), Validat
    fullname: new FormControl(null,{ updateOn: "blur",validators:[Validators.maxLength(70)]}),
    isheadquarters: new FormControl(false,{ updateOn: "change" }),
    bank4par: new FormControl(null,{ updateOn: "blur" }),
    swiftno: new FormControl(null,{ updateOn: "blur",validators:[Validators.maxLength(24)]}),
    bankatmno: new FormControl(null,{ updateOn: "blur" }),
    homepage: new FormControl(null,{ updateOn: "blur",validators:[Validators.maxLength(128)]}),
    email: new FormControl(null,{ updateOn: "blur" }),
    isactive: new FormControl(false,{ updateOn: "change",validators:[Validators.required]}),
    bdate: new FormControl(null,{ updateOn: "blur" }),
    edate: new FormControl(null,{ updateOn: "blur" }),
    twname: BankParty2NameTwDependEntityFormGroup.create(),
    usname: BankParty2NameUSDdependEntityFormGroup.create(),
  })
}
```

```
//子類創建提供api路由地址
abstract getApiPath(): string;
//子類創建FormGroup
abstract getFormGroup(): T;
```

bankentity

```
<ng-template #editBarTemplateRef>  
  <hb-editbar [amb]="amb" [rectype]="options.controls.rectype.getRawValue()" (toNewEvent)="toNew()" (doEditEvent)="doEdit()" (doDelEvent)="doDelete()" (doExitEvent)="doExit()" (doSaveEvent)="doSave()" (doUndoEvent)="doUndo($event)" (doDraftEvent)="doDraft()" (doBanEvent)="doBan()" (doCancelEvent)="doCancel()" [formGroup]="options">  
  </hb-editbar>  
</ng-template>
```

```
<hb-form-field titleKey="bcb.bank.bankabbrcode" [require]="true" md="6" xs="12">  
  <input type="text" pInputText hInputId formControlName="bankabbrcode" />  
</hb-form-field>
```

```
<hb-form-field titleKey="bcb.bank.abbrname" md="6" xs="12">  
  <form [formGroup]="options.controls.twname">  
    <input type="text" pInputText hInputId formControlName="abbrname" />  
  </form>  
</hb-form-field>
```

```
export class BankParty2NameTWDependEntityForm {  
  ver: FormControl<number | null>;  
  pk: FormControl<string | null>;  
  abbrname: FormControl<string | null>;  
  aliasname: FormControl<string | null>;  
  firstname: FormControl<string | null>;  
  fullname: FormControl<string | null>;  
  lastname: FormControl<string | null>;  
  middlename: FormControl<string | null>;  
}
```

- * hb-editbar是上方的編輯功能列
- * hb-form-field是輸入框欄位
- * 在前面form定義有twname、username這種屬於另一個FormGroup的需要另外使用一個form來包含住，令formFroup可以順利取得內容

bankentity

```
<hb-form-field titleKey="bcb.bank.bank4par" md="4" xs="12">
  <dlg-commoninput #dialogInput [formGroup]="options" field="bank4par" inputfield="bankid"
    displayfield="abbrname">
    <ng-template #input_template>
      <dlg-proxy #hbDynamicHost dlgtemplate="dlg-bankdialog"
        dialogHeader="{{ 'bcb.bank.bank4par' | translate }}"></dlg-proxy>
    </ng-template>
  </dlg-commoninput>
</hb-form-field>
```

- * dlg-commoninput則是彈跳視窗的輸入框
- * dlg-proxy是設定參照的彈跳視窗
- * dlgtemplate是參考的模板

bankentity

```
override doSave() {  
    // .value 和 .getRawValue 差別在於disabled value會取不到值  
    this.options.controls.abbrname.patchValue(  
        this.options.controls.twname.controls.abbrname.getRawValue() +  
        this.options.controls.twname.controls.fullname.getRawValue()  
    );  
  
    if(!this.options.controls.twname.controls.fullname.getRawValue()){  
        this.options.controls.twname.controls.fullname.patchValue('');  
        console.log('set');  
    }  
  
    super.doSave();  
}
```

設定存檔(doSave)時會將abbrName設定成
銀行簡稱+銀行全稱

藉由formcontrol取得內容值, 再將值套用
合併到abbrname中

下方則是檢查全稱沒有輸入則用空字串取代

值是透過super類的
patchValueToForm來填充表單的值
而這邊需要設定非新增的話
銀行編碼一律唯讀
所以設定amb為a新增時就將bankid disable

```
override patchValueToForm(data) {  
    this.options.patchValue(data);  
    if (this.amb !== 'a') {  
        this.options.controls.bankid.disable();  
    }  
}
```

Thank you