

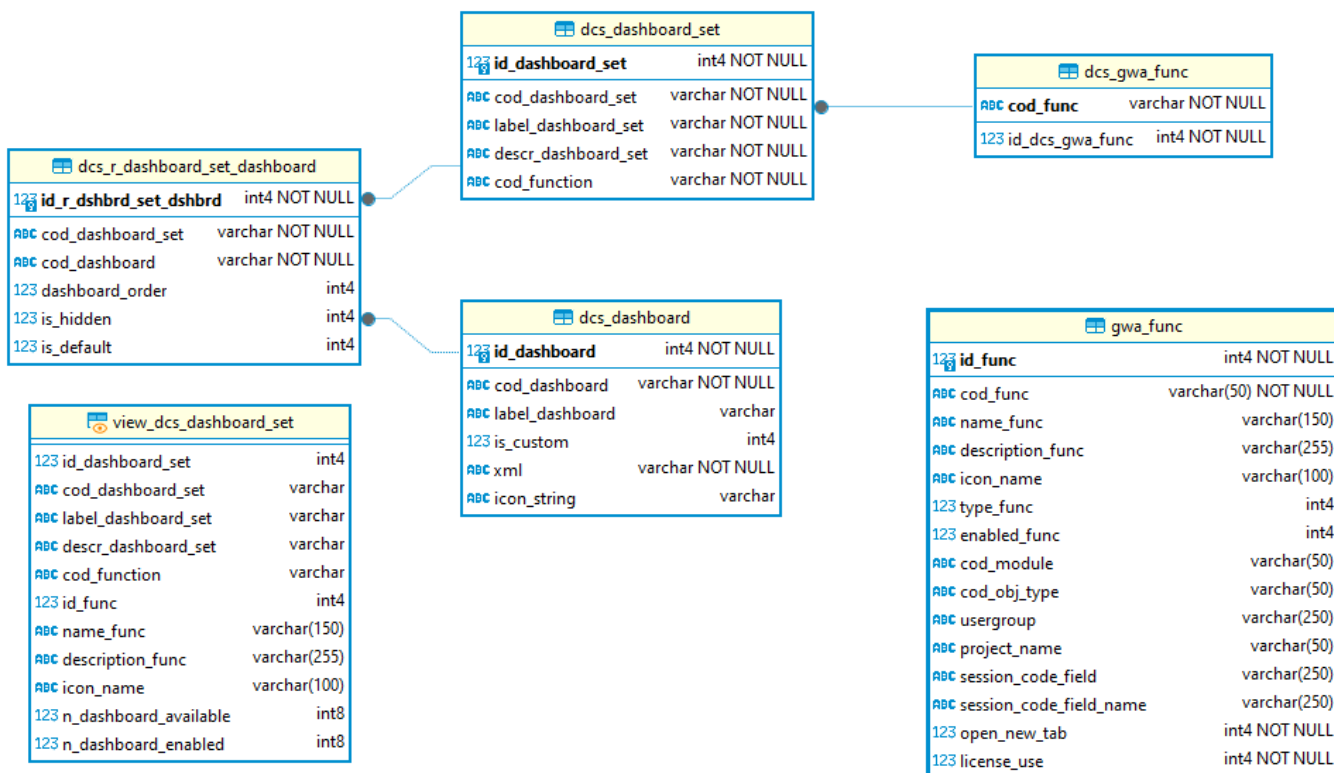
Dashboard Set Configurabili - Messa in esercizio

Questa pagina si riferisce alle operazioni preliminari volte alla prima messa in opera delle [Dashboard Set Configurabili](#).

Modello Dati

Overview entità coinvolte:

- Dashboard (*dcs_dashboard/dcs_dashboard_admin*)
- Dashboard Set (*dcs_dashboard_set/dcs_dashboard_set_admin*)
- relazioni Dashboard e Dashboard Set (*dcs_r_dashboard_set_dashboard*)
- Funzioni (*gwa_func*, presa dal modello dati delle pagine di accesso)
- Funzioni con Dashboard (*dcs_gwa_func*)



Installare gli script

Questo passaggio è necessario solo per eventuali porting alla 4.7.3 da versioni precedenti: La presenza della tabella *gwa_func* è un prerequisito. Dovrebbe già essere presente in quanto facente parte del modello dati delle pagine di accesso. In caso contrario usare lo script dedicato. Gli script sono resi disponibili come allegato a questo documento nella folder *script/*.

[postgres.sql](#)

```
-- issue #1428
```

```
-----  
-----  
--- Script GW 4.7.3
```

```
--- REPLACE xxx_metadata with the metadata schema name  
--- REPLACE xxx_data with the data schema name  
--- GSC, 2023.12  
-----  
-----
```

```
-----  
-----  
-- I M P O R T A N T  
-- gwa_function relation should already exist  
-----  
-----
```

```
-----  
-----  
-- DROP TABLE xxx_data.dcs_dashboard;
```

```
CREATE TABLE IF NOT EXISTS xxx_data.dcs_dashboard  
(  
    id_dashboard INTEGER NOT NULL,  
    cod_dashboard CHARACTER VARYING COLLATE pg_catalog."default" NOT  
NULL,  
    label_dashboard CHARACTER VARYING COLLATE pg_catalog."default",  
    is_custom INTEGER,  
    xml CHARACTER VARYING COLLATE pg_catalog."default" NOT NULL,  
    icon_string CHARACTER VARYING COLLATE pg_catalog."default",  
    CONSTRAINT dcs_dashboard_pkey PRIMARY KEY (id_dashboard),  
    CONSTRAINT uk_cod_dashboard UNIQUE (cod_dashboard)  
)  
WITH (  
    OIDS = FALSE  
)  
TABLESPACE pg_default;  
  
ALTER TABLE xxx_data.dcs_dashboard  
    OWNER TO xxx_data;
```

```
-- DROP TABLE xxx_data.dcs_dashboard_set;

CREATE TABLE IF NOT EXISTS xxx_data.dcs_dashboard_set
(
    id_dashboard_set INTEGER NOT NULL,
    cod_dashboard_set CHARACTER VARYING COLLATE pg_catalog."default"
NOT NULL,
    label_dashboard_set CHARACTER VARYING COLLATE pg_catalog."default"
NOT NULL,
    descr_dashboard_set CHARACTER VARYING COLLATE pg_catalog."default"
NOT NULL,
    cod_function CHARACTER VARYING COLLATE pg_catalog."default" NOT
NULL,
    CONSTRAINT dcs_dashboard_set_pkey PRIMARY KEY (id_dashboard_set),
    CONSTRAINT uk_cod_dashboard_set UNIQUE (cod_dashboard_set),
    CONSTRAINT uk_cod_function UNIQUE (cod_function),
    CONSTRAINT fk_cod_func FOREIGN KEY (cod_function)
        REFERENCES xxx_data.dcs_gwa_func (cod_func) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE xxx_data.dcs_dashboard_set
    OWNER TO xxx_data;

-----

-- DROP TABLE xxx_data.dcs_gwa_func;

CREATE TABLE IF NOT EXISTS xxx_data.dcs_gwa_func
(
    cod_func CHARACTER VARYING COLLATE pg_catalog."default" NOT NULL,
    id_dcs_gwa_func INTEGER NOT NULL,
    CONSTRAINT uk_cod_func UNIQUE (cod_func),
    CONSTRAINT fk_gwa_func FOREIGN KEY (cod_func)
        REFERENCES xxx_data.gwa_func (cod_func) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
        NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;
```

```
ALTER TABLE xxx_data.dcs_gwa_func
  OWNER TO xxx_data;

-----

-- DROP TABLE xxx_data.dcs_r_dashboard_set_dashboard;

CREATE TABLE IF NOT EXISTS xxx_data.dcs_r_dashboard_set_dashboard
(
  id_r_dshbrd_set_dshbrd INTEGER NOT NULL,
  cod_dashboard_set CHARACTER VARYING COLLATE pg_catalog."default"
NOT NULL,
  cod_dashboard CHARACTER VARYING COLLATE pg_catalog."default" NOT
NULL,
  dashboard_order INTEGER,
  is_hidden INTEGER,
  is_default INTEGER,
  CONSTRAINT dcs_r_dashboard_set_dashboard_pkey PRIMARY KEY
(id_r_dshbrd_set_dshbrd),
  CONSTRAINT fk_cod_dashboard FOREIGN KEY (cod_dashboard)
REFERENCES xxx_data.dcs_dashboard (cod_dashboard) MATCH SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID,
  CONSTRAINT fk_cod_dashboard_set FOREIGN KEY (cod_dashboard_set)
REFERENCES xxx_data.dcs_dashboard_set (cod_dashboard_set) MATCH
SIMPLE
ON UPDATE CASCADE
ON DELETE CASCADE
NOT VALID
)
WITH (
  OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE xxx_data.dcs_r_dashboard_set_dashboard
  OWNER TO xxx_data;

-----
```

Installare i groovy

Tutti i groovy (prefisso dcs_) sono nuovi e possono essere copiati senza conseguenze. Sono resi disponibili come allegato a questo documento nella folder groovy/.

[dsc-groovy.zip](#)

Importare metadati geoweb

E' messa a disposizione la configurazione dei metadati come allegato a questo documento [dsc-gw-metadati.zip](#), dedicato alle Dashboard personalizzate. La presenza della classe gwa_func è un prerequisito. In caso di mancanza il relativo script va ripreso dal modello dati delle pagine di accesso, e i metadati dalla ultima versione di prodotto. Una base per solo scopo di sviluppo/test è disponibile anche nell'ambiente di test 4.7.*. File presenti come allegato a questo rilascio:

- **gw_exp.projects_sm+gadmin.xml** (import consigliato) contiene I due project
 - dashboard_management
 - dashboard_management_admin
 - con dentro il theme DCS - DASHBOARD SET con le relative class:
 - dcs_dashboard
 - dcs_dashboard_admin
 - dcs_dashboard_set
 - dcs_dashboard_set_admin
 - dcs_gwa_func
 - dcs_r_dashboard_set_dashboard

con permessi su ambito di default

- **gw_exp.projects_sm+gadmin+2test.xml** contiene tutte le risorse del punto precedente, con l'aggiunta di due project che simulano le funzioni degli utenti finali:
 - dashboard_set_test_1
 - dashboard_set_test_2
- **gw_exp.themes_DCS - DASHBOARD SET (permissions).xml** contiene solo il theme DCS - DASHBOARD SET con tutte le relative classi elencate sopra ed i relativi permessi
- **gw_exp.themes_DCS - DASHBOARD SET.xml** contiene solo il theme DCS - DASHBOARD SET con tutte le relative classi elencate sopra senza permessi

L'utente configuratore deve inizialmente importare i progetti consigliati nell'ambiente di sviluppo del prodotto: i progetti dedicati al SM (*dashboard_management*) e all'utente GWADMIN (*dashboard_management_admin*).

I permessi di azioni e progetti dedicati al SM, sono impostati al gruppo *SOLUTION_MANAGER*. I permessi di azioni e progetti dedicati al GWADMIN, sono impostati al gruppo *GW_ADMIN*.

Fare gli eventuali aggiustamenti del caso.

Raffinare la configurazione metadati

La funzione dedicata al SM va abilitata al Solution Manager nel modello dati delle pagine di accesso. La funzione dedicata al GW ADMIN va abilitata all'eventuale utente di amministrazione ne modello dati delle pagine di accesso. Raffinare la configurazione sulle classi presenti nel tema DCS -

DASHBOARD SET nel webadmin.

Rivedere i permessi

Eventualmente rivedere tutti i permessi alle varie classi, considerando che escono con gruppo = 'SOLUTION_MANAGER'.

Esplicitare le funzioni con Dashboard Set

La tabella **dcs_gwa_func**, di cui è rilasciata anche la classe di gestione, **serve a censire esplicitamente tutte le funzioni che erogano un dashboard set.**

E' quindi onere dell'utente configuratore scremare in origine le classi sulle quali si può generare un Dashboard set.

From: <https://wiki.geowebframework.com/> - **GeowebFramework**

Permanent link: https://wiki.geowebframework.com/doku.php?id=gwusermanual:interface:dashboard_set_configurabili_setup

Last update: **2024/12/09 12:42**

