

VITROSPLAT – API & Bootstrap Notes (v1)

Stato: **work** | Data: 2025-10-02 | Ambito: vv_work

1) Architettura di base

Entry-point (pagine e AJAX) includono **una sola volta** `lib/bootstrap.php`.

Librerie (in `inc/` o file "lib" non entry) **non** includono nulla: esportano funzioni/classi senza side-effects.

Ordine di bootstrap (definitivo)

1. `config.php` (root) → sole **costanti** + log base + version info (nessuna funzione app)
2. `lib/db_utilities.php` → PDO & helper DB
3. `lib/helper.php` → utility pure (senza DB)
4. `lib/functions.php` → funzioni applicative (possono usare DB & helper)
5. `lib/config.php` → sessione/ruoli/utente, auto `BASE_URL` / `VV_ENV`, **`current_user()`**

Motivazione: `lib/config.php` dipende da `vv_app_meta()` e da funzioni app; perciò va **dopo** `functions.php`.

2) File e responsabilità

- `config.php` (**root**): costanti di ambiente, DB creds, logging, `vittros_load_version_info()`.
- `lib/db_utilities.php`: `db_select_row`, `db_select_all`, `db_exec`, `db_insert`, ... + API bozze (`draft_*`).
- `lib/helper.php`: utility pure (es. `h()`, `vv_base_url()`, `vv_url()`), nessun accesso DB, nessuna sessione.
- `lib/functions.php`: logica applicativa e servizi (meta app, temi, thumbs, playlist, suggerisci cartelle, ecc.).
- `lib/config.php`: sessione, ruoli, `current_user()`, `current_role()`, `role_gte()`, `require_role()`, visibilità post, badge info.
- `lib/core.php` (opzionale): minimi setup (guard `VV_ENV`, `mb_internal_encoding`, `session_start()` difensivo). Usare solo se serve.

Regola: evitare doppioni. Le funzioni devono vivere in **un solo** file; altrove solo `if (!function_exists(...))` come guardia.

3) Costanti principali (root `config.php`)

- `BASE_URL`, `BASE_FS`
- `DB_HOST`, `DB_NAME`, `DB_USER`, `DB_PASS`
- `DEBUG_LOGFILE`, `DEBUG_VITROS`

- APP_VERSION, APP_RELEASE, APP_ENV (facoltativa), PUBLIC_ORIGIN
- VV_ENV viene determinata da lib/config.php (auto in base allo slug: _test, _work, default prod).

4) Sessione & Utente (in lib/config.php)

- vv_session_get(string \$k, \$default=null)
- _vv_user_arr(): ?array → legge \$_SESSION e restituisce {id, username, ruolo} o null
- is_logged_in(): bool
- current_user(): ?array
- current_role(): string → default ospite
- role_gte(string \$needed): bool
- require_role(string \$role): void

Helper ruoli consigliati (unificati su current_user()):

```
function current_user_role(): string { return current_role(); }
function is_admin(): bool { return role_gte('admin'); }
function can_edit_post(array $post): bool {
    if (is_admin()) return true;
    $u = current_user();
    if (!$u) return false;
    return (int)($post['autore_id'] ?? -1) === (int)($u['id'] ?? -2);
}
```

Collocazione: lib/config.php (non in helpers.php). Così c'è **una sola fonte di verità** per l'utente.

5) Utilities (in lib/helper.php)

- h(\$s): string → HTML escape corto.
- vv_base_url(): string → da BASE_URL.
- vv_url(string \$path='/'): string
- (facoltativo) vv_path(string \$rel=''): string se non già fornita altrove.

Evita duplicati: h() deve esistere in un solo file (qui) con guardia if (!function_exists('h')) { ... }.

6) Funzioni applicative (in lib/functions.php)

- **Meta & versioni:** vv_app_meta(), vv_latest_changelog(), vv_list_versions(), vv_next_version_no(), vv_save_post_version() ...

- **Tema & stile:** `vv_theme_context()`, `vv_enqueue_theme_css(array $opts=[])` (in un solo posto!), `emoji_ruolo()`.
- **DB high-level:** `getPDO()`, `getValoreConfigurazione($campo)`, `aggiornaConfigurazione($campo,$valore)`.
- **Galleria:** `suggerisci_cartelle()`, `elenco_cartelle_con_immagini()`, `has_images_in_dir()`.
- **Log:** `debug_log($msg, $livello='debug')` (unica definizione qui).
- **Varie:** `vv_cache_bust_token()`, `vv_client_ip()`, `get_user_ip()`, `vv_year_span()`.

Se una funzione usa il DB o meta app, sta **qui**. `helpers.php` resta "pure".

7) Visibilità post (in `lib/config.php`)

- Costanti: `VIS_PUBBLICO`, `VIS_PRIVATO`, `VIS_NAT`.
- `can_view_post(array $post): bool` → usa `is_logged_in()`, `role_gte('amik_nat')`, autore per bozze, ecc.

8) Convenzioni di include

- Entry-point root: `require_once __DIR__ . '/lib/bootstrap.php';`
- Entry-point AJAX: `require_once __DIR__ . '/../lib/bootstrap.php';`
- Librerie (`inc/*.php`): **nessun** `require` → esportano funzioni.

Controlli automatici consigliati:

```
grep -RIInE 'lib/(db_utilities|functions)\.php' . | grep -vE 'bootstrap|\.bak'
# Nessuna lib deve includere questi file.
```

9) Best practices

- **Zero output** in `lib/bootstrap.php` e file `lib/*`: evita "headers already sent".
- `require_once` sempre con `__DIR__` e path relativo **corretto**.
- Ogni funzione con **guardia**:

```
if (!function_exists('debug_log')) { function debug_log(...) { ... } }
```

- **Tipi morbidi:** evita `declare(strict_types=1)` in legacy condiviso, inseriscilo solo in file nuovi ben isolati.
- **Log:** punta `DEBUG_LOGFILE` al percorso condiviso; usa `debug_log()` con contesto (ID utente, file:line).

10) Esempi d'uso

Recupero lista post (guest)

```
require_once __DIR__ . '/lib/bootstrap.php';
require_once __DIR__ . '/inc/diario_list_lib.php';

$params = [ 'q' => $_GET['q'] ?? '', 'limit' => 30, 'page' => 1 ];
$data = vv_fetch_posts($params);
foreach ($data['rows'] as $r) {
    vv_render_post_item($r);
}
```

Controllo permessi

```
require_once __DIR__ . '/lib/bootstrap.php';
require_role('editor');
```

Log applicativo

```
require_once __DIR__ . '/lib/bootstrap.php';
debug_log('azione: salva_post', ['post_id'=>$id, 'utente'=>current_user()]);
```

11) TODO / Pulizia tecnica

- [] **Deduplicare** definitivamente `h()`, `vv_base_url()`, `vv_enqueue_theme_css()`, `debug_log()` → una sola sede.
- [] **Spostare** `current_user_role()`, `is_admin()`, `can_edit_post()` in `lib/config.php` (basate su `current_user()`).
- [] Consolidare `diario_list_lib.php` (root vs `inc/`) → una sola versione in `inc/`.
- [] Uniformare variabile tema: `$tema_attivo` definita in `lib/config.php` via `getValoreConfigurazione('tema_attivo')` con fallback `default`.
- [] Aggiungere script CI locale: `grep include vietati, php -l` su tutto, test smoke CLI.

12) Checklist migrazione (sicura)

1) **Backup** (`tar -caf ... vittrosviaggi_work`). 2) Allinea `lib/bootstrap.php` all'ordine sopra. 3) Rimuovi include diretti di `lib/functions.php` / `lib/db_utilities.php` dagli entry-point → usa solo `bootstrap.php`. 4) Rimuovi `bootstrap.php` da **librerie**. 5) Dedupe funzioni (`h`, `vv_base_url`, `vv_enqueue_theme_css`, `debug_log`) con guardie e posizione unica. 6) Smoke test: `guest`, `modifica_post.php`, `media_popup.php`, 2 AJAX core.

13) Standard di codice

- Naming: `vv_*` per helpers/app; verbi italiani per funzioni bacino storico (`suggerisci_cartelle`).
- Error handling: eccezioni PDO e `try/catch` dove ha senso, altrimenti log + HTTP 500 con messaggio generico.
- Sicurezza: `h()` per output HTML, `prepared statements` ovunque, no dati non sanificati in query.

*Questa pagina è pensata per essere allegata alle "Developing Notes" del progetto e pubblicata come README della piattaforma **VITTROSPLAT**.*