TÀI LIỆU KIỂM THỬ WORKFLOW ĐĂNG NHẬP

idv4 + apiv4 | Login, 2FA, xác thực email, backup code, logout

Mục đích	Tài liệu hóa luồng đăng nhập thực tế để đội test đối chiếu chức năng, API, session, 2FA và các case lỗi.
Nguồn nội dung	Workflow đăng nhập thực tế theo code trong idv4 và API apiv4 do BA/dev cung cấp.
Ngày tạo	26/05/2026
Phạm vi	GET/POST /login, API /v1/login, Google Authenticator 2FA, xác thực email device, backup code, logout.

Ghi chú cho đội test: Nên dùng tài liệu này như checklist đối chiếu hành vi thực tế trên UI, API response, session/cache và database. Các đường dẫn/file/line trong tài liệu là mốc tham chiếu để dev kiểm tra lại khi phát sinh bug.

Mục lục

1. Tổng quan luồng đăng nhập

Luồng đăng nhập có 4 nhánh chính: login thường, login cần Google Authenticator, login cần xác thực email device, và login bằng backup code. idv4 chịu trách nhiệm UI/session; apiv4 chịu trách nhiệm validate, tìm account, kiểm tra password/IP, phát hành JWT và xử lý challenge 2FA.

Bước	Thành phần	Hành động chính	Kết quả cần kiểm tra
1	idv4 UI	User vào GET /login, nhập user/pass và submit POST /login.	Form gửi đúng field user, pass; client chỉ check rỗng trước submit.
2	UsersController	Validate login_id/password, chống login lại khi đã authenticated, reset pending 2FA nếu có reset_2fa=1.	Sai validate thì hiển thị Flash error và không gọi API.
3	ApiClient	POST /login tới API_BASE_URL = https://api-id.vinaten.vn/v1/ với login_id, password, ip, browser.	Request JSON, không kèm JWT vì auth=false.
4	apiv4	Validate, tìm account, verify password, check whitelist IP, xác định có cần 2FA/email hay không.	Response code=10000 + token nếu thành công; lỗi nghiệp vụ dùng code 1002/1001/1003/1004.
5	idv4	Dựa vào security_status/fa_status để redirect sang verify2Fa, confirm2FaToMail hoặc lưu session login.	Session ApiAuth/Auth được tạo đúng; redirect về redirect hợp lệ hoặc dashboard.

1.1 Sơ đồ luồng rút gọn

```
GET /login
  -> render templates/Users/login.php
POST /login
  -> UsersController::login()
  -> getValidatedCredentials()
  -> ApiClient POST https://api-id.vinaten.vn/v1/login
  -> apiv4 V1/AuthController::login()
  -> AuthService::authenticate()
      -> validate input
      -> find account/contact
      -> verify password
      -> check whitelist IP
      -> issue access token OR 2fa_challenge token
  -> idv4 resolveTwoFactorChallenge()
      -> no 2FA: persistAuthenticatedSession() -> /
      -> Google Authenticator: /users/verify2-fa
      -> Email device code: Users::confirm2FaToMail()
      -> Backup code: /confirm-backup
```

2. Route, view và controller idv4

2.1 GET /login

• Route idv4 map GET /login tới UsersController::login() trong config/routes.php line 58.

• UsersController cho phép action login không cần auth ở UsersController.php line 32.

• GET request chỉ render templates/Users/login.php.

• Form id là formLoginTenten, action POST /login, input gửi lên gồm user và pass.

• JS trong view chỉ kiểm tra rỗng ở client trước khi submit; validate chính vẫn cần ở controller/API.

```
$this->Form->create(null, [
  'url' => ['controller' => 'Users', 'action' => 'login'],
  'id' => 'formLoginTenten',
])
```

2.2 POST /login

• Luồng chính nằm tại UsersController.php line 38.

• Controller disable layout login khi POST.

• Nếu user đã authenticated thì redirect về target an toàn bằng getSafeRedirectTarget().

• Nếu URL có ?reset_2fa=1 thì xóa session 2FA đang pending.

• getValidatedCredentials() trim user, lấy pass, kiểm tra rỗng và giới hạn độ dài.

```
$loginId = trim((string)$this->request->getData('user'));
$password = (string)$this->request->getData('pass');

Đi□u kiện idv4:
- login_id không được r□ng
- password không được r□ng
- login_id <= 128 ký tự
- password <= 256 ký tự
```

Điểm test quan trọng: màn login không nên chỉ dựa vào JS. Cần verify server-side validate vẫn hoạt động khi tắt JS hoặc gọi trực tiếp POST /login bằng Postman/cURL.

2.3 Gọi API login

• Controller gọi authenticateAgainstApi(login_id, password).

• ApiClient post endpoint /login với login_id, password, ip, browser.

• API_BASE_URL đang là https://api-id.vinaten.vn/v1/, nên endpoint thực tế là POST https://api-id.vinaten.vn/v1/login.

• Header mặc định gồm Accept: application/json và User-Agent: TentenIdClient/1.0.

• Login truyền auth=false nên request không kèm JWT.

```
$this->ApiClient->post('/login', [
    'login_id' => $loginId,
    'password' => $password,
    'ip' => (string)($this->getClientIp() ?? ''),
    'browser' => $_SERVER['HTTP_USER_AGENT'] ?: 'unknown',
], [], false);
```

3. API apiv4 xử lý login

3.1 Route và controller API

• Route POST /v1/login map tới V1/AuthController::login() trong apiv4/config/routes.php line 51.

• AuthController::login() lấy data request, tạo AuthService và gọi authenticate(data).

• Data API nhận gồm login_id, password, ip, browser.

```
{
  "login_id": "user nhập",
  "password": "mật kh□u",
  "ip": "IP client",
  "browser": "User-Agent browser"
}
```

3.2 Validate và tìm account

• AuthService::authenticate() ở AuthService.php line 155 dùng AuthValidator.

• login_id bắt buộc; nếu là email thì validate email; nếu là username thì dài 3-32 ký tự và chỉ gồm chữ/số/_.

• password bắt buộc và tối đa 50 ký tự ở API.

• Chỉ cho account_type trong [1, 4].

• Nếu login_id không phải email thì tìm Accounts.login_id; nếu là email thì tìm qua Contacts.email.

• Điều kiện chính: Accounts.status=1, Contacts.contact_type=0, Accounts.account_type IN (1,4).

Lưu ý lệch validate: idv4 cho password tối đa 256 ký tự, nhưng API giới hạn password tối đa 50 ký tự. Đội test nên có case riêng để xác nhận thông báo lỗi/handling khi password từ 51-256 ký tự.

3.3 Verify password

• Nếu type_password=bcrypt thì dùng password_verify().

• Nếu password cũ dạng md5 thì so sánh md5(plain); nếu đúng thì nâng cấp sang password_hash() và set type_password=bcrypt.

• Sai password trả code=1002, message Invalid username or password.

```
{
  "code": 1002,
  "message": "Invalid username or password",
  "data": [],
  "errors": {
    "error": {"error": "Invalid username or password"}
  }
}
```

3.4 Check whitelist IP

• Sau password, API gọi checkIpInList(data['ip'], user).

• Nếu account có list_ip và IP hiện tại không nằm trong list thì trả lỗi: Your IP address is not allowed to access this account.

• idv4 map lỗi này sang tiếng Việt: Địa chỉ IP của bạn không được phép truy cập tài khoản này.

4. Token, session và xác định 2FA

4.1 Xác định có cần 2FA/email device

• API kiểm tra checkRememberDevice(browser, user).

• Nếu security_status=1: không dùng remember email device, bắt Google Authenticator.

• Nếu security_status=0: kiểm tra bảng AccountRemember theo login_id + http_user_agent + expiration_date > now.

• Nếu chưa có remember device hợp lệ thì cần xác thực email.

• Sau đó API tạo JWT access hoặc JWT purpose=2fa_challenge.

4.2 Không cần 2FA

• JWT có purpose=access, exp=+8 giờ.

• API ghi cache key dạng {user_id}_login_{hash(token)}.

• idv4 nhận token access, xóa pending 2FA cũ, renew session và ghi ApiAuth/Auth.

```
Session sau login thành công:
- ApiAuth.jwt  = JWT access từ API
- ApiAuth.user = thông tin user dùng cho frontend
- Auth         = identity c□a Cake Authentication plugin
Redirect v□ redirect query n□u hợp lệ, n□u không thì v□ /
```

4.3 Cần 2FA/email

• JWT có purpose=2fa_challenge, exp=+300 giây, jti=random.

• API ghi cache key dạng {user_id}_2fa_{hash(token)} trong cache two_factor.

• Nếu là xác thực email device, API tạo mã 6 số, lưu accounts.fa_code, accounts.fa_code_expiration=+10 phút và publish queue gửi mail.

• idv4 chỉ coi login/verify thành công khi code=10000 và data.token không rỗng.

```
Response success c□n 2FA/email:
{
  "code": 10000,
  "message": "Success",
  "data": {
    "token": "JWT purpose=2fa_challenge",
    "security_status": 0 hoặc 1,
    "login_id": "...",
    "type": 1,
    "fa_status": 1,
    "security_count": 0,
    "credit": 0
  }
}
```

4.4 resolveTwoFactorChallenge() ở idv4

```
if ($userData['security_status'] === 1) {
    return 'verify2Fa';
}

if ($userData['fa_status'] === 1) {
    return 'confirm2FaToMail';
}

return null;
```

5. Các nhánh chức năng cần test

Nhánh	Điều kiện	UI/Endpoint	Expected chính
Login thường	security_status != 1 và fa_status != 1	POST /login -> API /v1/login	Lưu ApiAuth.jwt, ApiAuth.user, Auth; redirect về redirect hợp lệ hoặc /.
Google Authenticator	security_status = 1	/users/verify2-fa -> POST /v1/verify-2fa	Nhập TOTP 6 số đúng thì trả access token mới; sai thì code=1002.
Email device code	security_status=0 và fa_status=1	Users::confirm2FaToMail -> POST /v1/confirm-2fa-to-email	Mã email 6 số đúng thì access token mới; remember=on lưu AccountRemember 60 ngày.
Backup code	User ở màn Google Authenticator và chọn /confirm-backup	POST /v1/verify-backup-code	Backup code 8 ký tự đúng thì tạo access token, xóa code đã dùng khỏi DB.
Logout	User đã login	/logout	idv4 xóa Authentication, ApiAuth, Auth, Pending2Fa và cache pending_2fa nếu có; redirect /login.

5.1 Google Authenticator 2FA

• idv4 gọi storePendingTwoFactorSession(), không lưu token tạm trực tiếp trong session.

• storePendingTwoFactorSession() tạo pendingKey random, ghi cache pending_2fa gồm token, user, redirect, expires_at=time()+300.

• Session chỉ lưu Pending2Fa=pendingKey.

• View templates/Users/2fa.php có form nhập code_confirm là mã Google Authenticator 6 số.

• Controller gọi API verify-2fa với Authorization: Bearer JWT purpose=2fa_challenge.

• Middleware API yêu cầu purpose=2fa_challenge, path hợp lệ và cache two_factor còn tồn tại.

```
POST /v1/verify-2fa
Authorization: Bearer JWT purpose=2fa_challenge
Payload:
{
  "code2Fa": "mã Google Authenticator 6 s□",
  "remember": "on hoặc r□ng",
  "browser": "User-Agent"
}
```

5.2 Xác thực email device

• Dùng khi security_status=0 nhưng fa_status=1.

• View templates/Users/confirm_2fa_to_mail.php nhập code_confirm và remember.

• API tìm account theo accountId từ JWT, fa_code, security_status=0, status=1, del_flg=0.

• Nếu remember=on thì ghi/cập nhật AccountRemember theo login_id, http_user_agent, ip, expiration_date=+60 ngày.

• Sau khi xác thực đúng, accounts.fa_code được set null.

```
POST /v1/confirm-2fa-to-email
Authorization: Bearer JWT purpose=2fa_challenge
Payload:
{
  "code2Fa": "mã email",
  "remember": "on hoặc r□ng",
```

```
  "browser": "User-Agent"
}
```

5.3 Backup code

• Từ màn Google Authenticator có link /confirm-backup.

• View nhập code_confirm là backup code 8 ký tự chữ/số.

• API validate alphanumeric 8 ký tự, tìm AccountAuthBackupCodes theo login_id và status=1.

• Nếu backup_code_type=hash thì password_verify(), nếu không thì hash_equals().

• Code đúng thì tạo access token mới, xóa cache challenge token, ghi cache login token và xóa backup code đã dùng khỏi danh sách.

```
POST /v1/verify-backup-code
Authorization: Bearer JWT purpose=2fa_challenge
Payload:
{
  "code2Fa": "backup code",
  "browser": "User-Agent"
}
```

5.4 Logout

• Route /logout gọi UsersController::logout().

• idv4 gọi Authentication->logout(), clearAuthSession(), clearPendingTwoFactorSession() rồi redirect /login.

• Session bị xóa gồm ApiAuth, Auth, Pending2Fa và cache pending_2fa nếu có.

• Lưu ý: idv4 logout hiện không thấy gọi API /v1/logout, nên cache _login_ phía API có thể còn tới khi hết TTL/cache, trừ khi nơi khác gọi logout API.

6. API endpoint và response code

Endpoint	Auth	Payload chính	Kết quả/ghi chú
POST /v1/login	Không JWT	login_id, password, ip, browser	Trả access token hoặc 2fa_challenge token. Chỉ success khi code=10000 và data.token có giá trị.
POST /v1/verify-2fa	Bearer 2fa_challenge	code2Fa, remember, browser	Verify TOTP 6 số. Đúng thì trả access token mới; xóa challenge cache.
POST /v1/confirm-2fa-to-email	Bearer 2fa_challenge	code2Fa, remember, browser	Verify mã email 6 số. remember=on thì lưu thiết bị 60 ngày.
POST /v1/verify-backup-code	Bearer 2fa_challenge	code2Fa, browser	Verify backup code 8 ký tự. Đúng thì xóa backup code đã dùng.

6.1 Response code chính

Code	Ý nghĩa	Cách test/check
10000	Success	Cần có data.token. Nếu thiếu token, idv4 phải coi là fail.
1001	Invalid data	Sai format input/validate API.
1002	Lỗi nghiệp vụ	Sai username/password, sai mã 2FA, lỗi IP không được phép, v.v.
1003	System error	Lỗi hệ thống/API exception.
1004	Permission error	Sai quyền/token/challenge path hoặc token hết hạn.

7. Checklist test đề xuất

Checklist dưới đây dùng để đội test tick theo từng nhóm. Có thể copy ID test case vào bug report để dev trace lại luồng tương ứng.

ID	Nhóm	Điều kiện/Input	Expected	OK
LGN-001	UI Login	GET /login	Trang login render đúng, form action POST /login, field user/pass tồn tại.	[ ]
LGN-002	Client validate	Bỏ trống user/pass	JS hiển thị lỗi rỗng; không submit khi chưa đủ dữ liệu.	[ ]
LGN-003	Server validate	Tắt JS hoặc dùng Postman POST /login rỗng	Controller chặn, Flash error, không gọi API.	[ ]
LGN-004	Server validate	login_id > 128 ký tự	idv4 trả lỗi dữ liệu đăng nhập không hợp lệ, không gọi API.	[ ]
LGN-005	Validate lệch FE/API	password 51-256 ký tự	idv4 có thể pass nhưng API reject. Cần thông báo lỗi hợp lý, không lỗi trắng.	[ ]
LGN-006	API login	Login đúng, account không cần 2FA	API trả code=10000 + access token; idv4 tạo ApiAuth/Auth và redirect / hoặc redirect query hợp lệ.	[ ]
LGN-007	API login	Sai password	API trả code=1002; UI hiển thị Sai ID/mật khẩu hoặc message đã map.	[ ]
LGN-008	API login	Account status != 1 hoặc account_type ngoài [1,4]	Không login được, không tạo session.	[ ]
LGN-009	Email login	Nhập login_id là email của contact hợp lệ	API tìm qua Contacts.email với contact_type=0; login đúng nếu account hợp lệ.	[ ]
LGN-010	Username login	Nhập login_id là username hợp lệ	API tìm Accounts.login_id và login đúng nếu account hợp lệ.	[ ]
SEC-001	IP whitelist	Account có list_ip, IP hiện tại không nằm trong list	API trả lỗi IP; idv4 hiển thị 'Địa chỉ IP của bạn không được phép truy cập tài khoản này.'.	[ ]
SEC-002	IP whitelist	Account có list_ip, IP hiện tại nằm trong list	Qua bước IP và tiếp tục login/2FA bình thường.	[ ]
SEC-003	JWT	API code=10000 nhưng data.token rỗng	idv4 phải coi là fail, không tạo session.	[ ]
2FA-001	Google Auth	security_status=1	idv4 redirect /users/verify2-fa, session chỉ lưu Pending2Fa, token nằm trong cache pending_2fa.	[ ]
2FA-002	Google Auth	Nhập TOTP đúng 6 số	API /v1/verify-2fa trả access token mới; idv4 clear pending và tạo session login.	[ ]
2FA-003	Google Auth	Nhập TOTP sai	API code=1002, UI báo mã không đúng, không tạo session.	[ ]
2FA-004	Google Auth	Token challenge hết hạn sau 300 giây	API reject auth/challenge, UI xử lý quay lại login hoặc báo hết hạn hợp lý.	[ ]
2FA-005	Middleware	Dùng challenge token gọi endpoint ngoài whitelist	API reject permission/auth.	[ ]
2FA-006	Middleware	Dùng access token gọi /v1/verify-2fa	API reject vì purpose không phải 2fa_challenge.	[ ]
MAIL-001	Email 2FA	security_status=0, fa_status=1	Redirect confirm2FaToMail, gửi/publish mã email 6 số, token purpose=2fa_challenge.	[ ]
MAIL-002	Email 2FA	Nhập mã email đúng còn hạn	Trả access token, clear challenge, set accounts.fa_code=null.	[ ]
MAIL-003	Email 2FA	Nhập mã email sai/hết hạn	Không login, UI báo lỗi phù hợp.	[ ]
MAIL-004	Remember device	Tick nhớ thiết bị	Ghi/cập nhật AccountRemember với expiration_date +60 ngày.	[ ]
MAIL-005	Remember device	Lần login sau cùng browser còn hạn	Không yêu cầu mã email lại nếu remember device hợp lệ.	[ ]
BKP-001	Backup code	Từ màn Google Auth bấm /confirm-backup	Hiển thị màn nhập backup code.	[ ]
BKP-002	Backup code	Nhập backup code 8 ký tự đúng	Trả access token, xóa code đã dùng khỏi DB/danh sách.	[ ]
BKP-003	Backup code	Nhập code sai format/không tồn tại/đã dùng	Không login, báo lỗi phù hợp.	[ ]

ID	Nhóm	Điều kiện/Input	Expected	OK
SES-001	Session	Login thường thành công	ApiAuth.jwt, ApiAuth.user, Auth tồn tại; session renew đã chạy.	[ ]
SES-002	Session	Có pending 2FA rồi vào /login?reset_2fa=1	Pending2Fa/cache pending bị xóa.	[ ]
SES-003	Redirect	Có redirect query hợp lệ	Sau login redirect đúng target an toàn.	[ ]
SES-004	Redirect	redirect query không an toàn/external	Không redirect ra domain ngoài; fallback về /.	[ ]
OUT-001	Logout	User gọi /logout	Xóa ApiAuth, Auth, Pending2Fa/cache pending và redirect /login.	[ ]
OUT-002	Logout/API cache	Logout xong dùng token API cũ	Cần xác nhận cache _login_ phía API còn hiệu lực đến TTL hay có cơ chế logout API khác.	[ ]

8. Gợi ý log bug khi test

Khi log bug, nên ghi rõ nhánh login, account test, security_status/fa_status/list_ip, endpoint được gọi, response code/message, token purpose nếu có, session/cache kỳ vọng và thực tế.

Trường	Nội dung nên ghi
Môi trường	DEV/STG/PROD, URL idv4, API_BASE_URL đang dùng.
Account test	login_id/email, account_type, status, security_status, fa_status, có list_ip hay không.
Bước thực hiện	GET /login, POST /login, màn 2FA/email/backup, logout, v.v.
Request/Response	Endpoint, payload đã che password/token, response code/message/data.token có hay không.
Session/cache	ApiAuth/Auth/Pending2Fa, cache pending_2fa/two_factor/login token nếu có quyền kiểm tra.
Expected vs Actual	Expected theo tài liệu này; Actual ghi ảnh/video/log cụ thể.

Các điểm cần chú ý khi test bảo mật: Không đưa password thật, JWT thật, mã 2FA thật hoặc thông tin nhạy cảm vào bug report/video public. Khi cần gửi log cho dev, hãy mask token/password/mã 2FA.