



Partner API V1.1 Technical Specification

1. Introduction

Welcome to the AlumierMD Partner API technical reference.

The AlumierMD™ Partner API is organised around REST architecture. Our API uses resource-oriented URLs which accept and return JSON-encoded responses. It uses standard HTTP response codes and appropriate verbs.

2. Security

All requests and responses are end-to-end encrypted and therefore we permit transmission only via HTTPS over SSL.

3. Rate Limiting

Endpoints are limited by default to 1000 requests per minute, per partner. If you find yourself reaching the threshold, please reach out to us so that we may increase this value. You'll identify these by a high number of HTTP Status 429 responses.

4. Authentication and Authorisation

Authentication is done using the Client Credential specification for OAuth 2.0. Access requires credentials (key and secret) that grants a temporary access token that can be used to query the API.

See: <https://www.oauth.com/oauth2-servers/access-tokens/client-credentials/>

If you have not received your credentials, please reach out to your AlumierMD account manager and complete the API onboarding form to request access.

Authorisation

Your credentials (and ultimately the temporary access token) are linked to your AlumierMD business platform account, meaning it has access to your data from the AlumierMD website.

4.1 OAuth Token Requests

OAuth Token Url: <https://api.alumierlabs.com/oauth/token>

Scope: basic

Client ID: [your id]

Client Secret: [your secret]

Do not omit scope or attempt other scopes than provided. The default scope available to all partners is 'basic'.

4.2 Token vitality

Note the expiration date (expires_in) on your token. At the time of writing, these tokens are long-lived for 7 days, but that duration might shorten.

It is expected that new tokens are requested when the previous ones expire. To prevent spam, each partner has a limit on the number of valid tokens available, therefore generating a new token before every request or hourly is not feasible. Daily, weekly token updates or generating upon expiration is recommended.

4.3 API Auth using Bearer

Once you've obtained your token, send a basic Authorization header with your token, i.e.

Authorization: Bearer

5. Versioning

This documentation outlines the specification for version 1 of the Partner API. The endpoints are identifiable by being prefixed with "/v1/". For other versions, refer to that version's documentation.

6. Errors

The API uses conventional HTTP response codes to indicate success and failure of the API requests.

API responses in the 200 range indicate success, whereas 400 indicate problems with the request (including authorization) and 500 indicate server errors with AlumierMD.

HTTP Response Codes

200 - OK Executed successfully.

400 - Bad Request The request is not understood by the server due to an incorrect syntax.

401 - Unauthorized No valid API key provided.

403 - Forbidden The API key doesn't have permissions to perform the request.

404 - Not Found The requested resource doesn't exist.

422 - Unprocessable Entity Generally, an issue with the request formatting or execution.

429 - Too Many Requests API queried too many times successively.

500, 502, 503, 504 - Server Errors A server problem on the Partner API Server. These should seldom occur.

Typically, a 422 (Unprocessable Entity) response would indicate a formatting or request error caused by the client. These could return a JSON body response and where more information can be seen, denoting what went wrong, i.e. a missing parameter.

Response Body

Data responses are wrapped in a “data” attribute, errors are wrapped in an “error”.

```
{
  "error": {
    "code": "parameter_missing",
    "message": "Parameter missing: email"
  },
  "data": {}
}
```

Attributes

- code (string): A fixed code identifying this type of error
- message (string): A description for debug purposes. Warning: These might not be user-friendly.

While the “data” node might be present, the content will always be empty in the case of an error.

7. JSON API Endpoint Reference

Note: The AlumierMD API will continue to grow its endpoints over time. At present, we only expose a single API endpoint.

As mentioned before, a temporary access token needs to be submitted with every API request. Please refer to section 4 of this document for more information on how to generate the access token.

7.1 Users Endpoints

Patients can be referred to as a type of user. The Users endpoint permits looking up patient information.

7.1.1 POST /v1/users/verify

Use this API endpoint to verify if a patient belongs to your AlumierMD business platform. If a patient does not belong to your business platform, you will not be allowed to sell AlumierMD products to that patient from your online store.

When you query a user by email, you will receive one of three answers:

1. The user belongs to your AlumierMD business platform (sale allowed)
2. The user belongs to another partner (sale not allowed)
3. The user is not registered with any partner. (sale not allowed, register this patient with your AlumierMD business profile first)

Request URL

POST /v1/users/verify

Request Body (JSON)

email: String | required

The email address of the patient's status you are querying.

Example Request Body

```
{
  "email": "patient@example.com"
}
```

Response Body (JSON)

sale_allowed: boolean

Whether or not sale is permitted to this user. This is the most critical of the lookup values and should be respected. When this value is false, you are not allowed to sell AlumierMD products to this user on your online store.

reason_code: string

Returns one of the following states:

"patient_owned": User belongs to your AlumierMD business platform.

"patient_not_owned": User belongs to another partner.

"patient_not_found": User is not registered with any partner.

message: string

Text description of the state. Used for debugging purposes. Your system should display your own message to your users based on the state.

signup_url: string (optional)

When the user does not exist (patient_not_found), a login URL with your AlumierMD professional join code pre-populated is returned, should you wish to redirect the user to register on the relevant AlumierMD website.

Example Responses

Case 1 - User belongs to the partner

```
{
  "sale_allowed": true,
  "reason_code": "patient_owned",
  "message": "This patient belongs to you."
}
```

Case 2 - User belongs to another partner

```
{
  "sale_allowed": false,
  "reason_code": "patient_not_owned",
  "message": "This patient is registered with another provider."
}
```

Case 3 - Email not registered on system

```
{
  "sale_allowed": false,
```

```

    "reason_code": "patient_not_found",
    "message": "The patient is not registered with AlumierMD.",
    "signup_url": "https://www.example.com/alumiermd /join?code=ABC123TEST"
}

```

7.1.2 POST /v1/users/register

If you've confirmed a patient does not belong to you and does not exist, this API can be used to register a patient with AlumierMD and assigned to your account.

AlumierMD website access and communication

This is a silent registration, meaning the patient will not be notified via email. If they try to log in on the AlumierMD website, they will be guided towards account activation and allowed to set their password. A professional can also log onto the AlumierMD Business Platform, visit the patient's profile, and manually send the activation email.

Request URL

POST /v1/users/register

Request Body (JSON)

first_name: String | required

The first name of the patient. This may not be left blank.

last_name: String | required

The last name of the patient. This may not be left blank.

email: String | required

The email address of the patient.

Example Request Body

```

{
  "first_name": "First",
  "last_name": "Last",
  "email": "success@example.com"
}

```

See Section 8.1.2 for example requests on the staging environment to create specific example scenarios.

Response Body (JSON)

registered: boolean

Whether the registration was successful.

reason_code: string

Returns one of the following states:

"register_success": Registration was successful.

"register_error_email_taken": The email is already registered on the system.

"validation_error": Invalid email, empty first or last name or other validation errors.

Errors generally respond with HTTP 422 or 500 codes. One can assume a customer is registered when you receive both a HTTP 200 OK and **registered** with a value of **true**.

message: string

A human readable description of the request's success or failure.

Example Responses

Case 1 – Registration successful

```
{
  "registered": true,
  "reason_code": "register_success",
  "message": "Patient successfully registered."
}
```

Case 2 – Validation error, such as sending through invalid email and first name.

```
{
  "registered": false,
  "reason_code": "validation_error",
  "message": "Email is invalid. First name can't be blank"
}
```

Case 3 – Error: The email is already registered. The user verify request (7.1.1) can clear up whether this patient belongs and whether a sale is allowed.

```
{
  "registered": false,
  "reason_code": "register_error_email_taken",
  "message": "The email address already in use."
}
```

8. Staging Environment

To make use of your provided Staging Credentials, use the Staging environment by swapping out instances of the Production domain with the Staging domain.

Production domain:

“https://api.alumierlabs.com”

Staging domain:

https://**staging**.api.alumierlabs.com

Perform all authorization and requests using this as your base URL.

Ensure you use the correct credentials for corresponding environment otherwise you will encounter authorization failure.

8.1 Testing your integration

By handling all the various scenarios and errors in the workflow, you can be assured that your integration will work smoothly when you transition to Production credentials.

Details on specific test scenarios below

8.1.1 POST /v1/users/verify

The below body contains various “email” parameters which are designed to trigger exact cases below.

Case 1 - User belongs to the partner

Example Request Body

```
{
  "email": "patient_owned@example.com"
}
```

Case 2 - User belongs to another partner

Example Request Body

```
{
  "email": " patient_not_owned@example.com"
}
```

Case 3 - Email not registered on system

Example Request Body

```
{
  "email": "not_found@example.com"
}
```

8.1.2 POST /v1/users/register

The below body contains various “email” parameters which are designed to trigger exact cases below.

Case 1 – Registration succesful

Example Request Body

```
{
  "first_name": "First",
  "last_name": "Last",
  "email": "success@example.com"
}
```

Case 2 – Validation error Example Request Body

```
{
  "first_name": "",
  "last_name": "Last",
  "email": "everythingelse@example.com"
}
```

Case 3 – Error: The email is already registered.

Example Request Body

```
{  
  "first_name": "First",  
  "last_name": "Last",  
  "email": "taken@example.com"  
}
```

8.2 Moving to Production

When you are satisfied that your Staging integration is fully tested and operational, you can start making use of the Production domain and your accompanying Production credentials at your convenience.

9. Changelog

Date	Notes	Author
27/09/2021	Initial draft	GN
15/02/2022	Finalised V1	GW
17/02/2022	Staging environment added	GN
22/07/2022	Added User Registration	GN