

Cerbo API:

## Supplement Endpoints:

Please note: All dates listed are listed in UTC, so you'll want to adjust for your local timezone. Also, the API responses include some routes that may not be available for your instance. Supplements can also be accessed through the `/alternate_plans/` endpoints, but because most clinics handle supplements separately you will want to use the `/supplements/` endpoints unless you're actively managing other products. Also, these endpoints pertain only to supplement definitions – if you're looking for patient-associated prescriptions for supplements (including prescribing details) please use the `/patients/:pt_id/supplements` endpoints.

Also note that while you have full usage rights over any supplement that was created/defined by your practice, and data sets that were installed or provided by a third party are subject to usage restrictions in the EMR.

### 1. Get all supplements

*Endpoint:* `/api/v1/supplements`

*Method:* GET

*Arguments:*

- **limit:** (optional) defaults to 100, but can be overridden (max is 100)
- **offset:** (optional) If the returned data indicates that there is more than the limit responses (`has_more: true`), then it will offset the list to allow you to cycle through long supplement list
- **active\_only:** (optional) If this parameter is set it will only include ACTIVE supplements
- **vendor\_code:** (optional) If this parameter is set it will only supplements with matching vendor code (note, this is different from the `external_ref_id`).
- **class:** (optional) If this parameter is set it will only supplements with that class specified
- **external\_ref\_id:** (optional) If this parameter is set it will only include items with the designated external identifier – generally you will want to make this unique so it only includes a single item.

*Sample Call:*

```
curl -X GET https://un:pwd@subdomain.md-hq.com/api/v1/supplements?
limit=10&class=sampleClass
```

### 2. Search all supplements

*Endpoint:* `/api/v1/supplements/:terms`

*Method:* GET

*Notes:* You can enter several terms with spaces between each (make sure to url-encode the search string, so spaces would be `%20`). Each term is treated as potentially a wildcard search so "vitam%20D" ("vitam D") would return "Vitamin D", "Vitamin D1000", etc.

*Arguments (same as List all Supplements call):*

- **limit:** (optional) defaults to 50, but can be overridden (max is 100)

- **offset**: (optional) If the returned data indicates that there is more than the limit responses (`has_more: true`), then it will offset the list to allow you to cycle through long supplement list
- **active\_only**: (optional) If this parameter is set it will only include ACTIVE supplements
- **vendor\_code**: (optional) If this parameter is set it will only supplements with matching vendor code (note, this is different from the `external_ref_id`).
- **class**: (optional) If this parameter is set it will only supplements with that class specified
- **external\_ref\_id**: (optional) If this parameter is set it will only include items with the designated external identifier – generally you will want to make this unique so it only includes a single item.

*Sample Call:*

```
curl -X GET https://un:pwd@subdomain.md-hq.com/api/v1/supplements/search/vitam%20d?limit=10&class=sampleClass
```

### 3. Get a single supplement

*Endpoint:* `/api/v1/supplements/:supplement_id`

*Method:* GET

*Arguments:* [none]

*Sample Call:*

```
curl -X GET https://un:pwd@subdomain.md-hq.com/api/v1/supplements/2
```

### 4. Create a new supplement

*Endpoint:* `/api/v1/supplements`

*Method:* POST

*Data:* property-value key-pairs (match the structure from the GET API call)

*Notes:* At a minimum, you must provide a name, and the name must be unique (it will throw an error if there is a duplicate note)

*Sample Call:*

```
curl --data '{"name": "Test Supplement", "nicknames": "MySup", "vendor_code": "TESTA", "description": "This is the short description of this entry", "external_ref_id": "12345", "inactive": false}' -X POST https://un:pwd@subdomain.md-hq.com/api/v1/supplements
```

### 5. Update a supplement

*Endpoint:* `/api/v1/supplements/:supplement_id`

*Method:* PATCH

*Data:* property-value key-pairs (match the structure from the GET API call) – any value which is provided will update, so do not add empty fields to the patch data unless you want these fields to be set to empty.

*Sample Call:*

```
curl --data '{"name": "Updated Name", "vendor_code": "NewCode"}' -X PATCH https://un:pwd@subdomain.md-hq.com/api/v1/supplements/12
```