

GARMIN INTERNATIONAL

# Garmin Connect Developer Program Courses API

---

*Version 1.0.1*

---

**CONFIDENTIAL**

## Table of Contents

1. Revision History .....	3
2. Getting Started.....	4
2.1. Purpose of the API .....	4
2.2. Consumer Key and Secret .....	4
2.3. User Registration.....	5
2.4. Requesting a Production Key .....	5
2.5. API Rate Limiting or Excessive Usage.....	6
3. Courses API Endpoint Details.....	7
3.1. Courses API Permissions .....	7
3.2. Courses.....	8
3.2.1 Field Definitions .....	8
3.2.2. Example JSON.....	9
3.2.3. Create.....	10
3.2.4. Retrieve.....	11
3.2.5. Update.....	11
3.2.6. Delete.....	12
3.2.7 Acceptance Criteria .....	12

## 1. Revision History

Version	Date	Revisions
1.0	12/01/2020	Initial revision
1.1	02/02/2020	Updated Course Point Types

## 2. Getting Started

### 2.1. Purpose of the API

The Garmin Connect Courses API is the underlying mechanism by which users can opt to import courses from third-party platforms into their Garmin Connect account, making it easy to manage this type of information in a centralized location.

### 2.2. Consumer Key and Secret

Garmin Connect Courses API partners will be provided with a consumer key and secret used to gain access to the Courses API. The consumer key is used to uniquely identify a partner and the consumer secret is used to validate that the requests received are from that partner and not a third-party that has gained unauthorized access to the consumer key. The consumer key can be considered public information, but the consumer secret is private. For the security of users, the consumer secret should be secured and never sent over a network in plain text. It is not permitted to embed the consumer secret into consumer products like mobile apps.

Consumer key credentials are created using the Developer Portal and creating Apps (<https://developerportal.garmin.com/user/me/apps?program=829>). Each app represents a unique consumer key. Your first app will generate an evaluation-level consumer key that is rate limited. Once your integration has been verified for production, subsequent apps will create consumer keys with production-level access. Please see “Requesting a Production Key” below for more information.

**Note:**

Multiple consumer keys should be created to correspond to projects or implementations whose user base is logically separated. A common scenario is for one partner to manage user data from multiple other companies. A new key should be created and associated with each managed company so that Garmin users can make an informed decision to consent to sharing their data with just that company.

## 2.3. User Registration

Before a partner can write data to a user's account, the user must grant the partner write access. Please refer to the detailed Garmin OAuth documentation for details on acquiring, authorizing, and signing with a User Access Token (UAT) to write data to a Garmin user's account.

## 2.4. Requesting a Production Key

The first consumer key generated through the [Developer Portal](#) is an evaluation key. This key is rate limited and should only be used for testing, evaluation, and development. To receive production level credentials, Garmin must review and approve the Courses API integration to ensure a high-quality user experience in Garmin Connect. Garmin also reserves the right to review partner applications and/or websites to ensure proper usage of Garmin assets (e.g. device images) and adherence to Garmin brand guidelines.

Please email Courses API support at [connect-support@developer.garmin.com](mailto:connect-support@developer.garmin.com) to request and schedule a production readiness review. Garmin will review the following technical aspects of the integration:

- Authorization and correct use of UATs for at least two Garmin Connect users;
- No unnecessary or excessive API call utilization or volume;
- Proper handling of quota violations and subsequent retry attempts.

If the technical integration is not approved, any open issues must be corrected, and another review will be required. Once the technical integration is approved, Garmin may conduct a user experience review. This review can be achieved by application demonstration to Garmin via video conference or other mutually agreed upon method. This review is used to confirm the following criteria are met:

- Proper representation of all Garmin trademarked/copyrighted terms;
- Proper representation Garmin products and product images; and
- User experience (UX) flow does not misrepresent Garmin or reflect Garmin poorly.

Once all reviews are approved, production credentials (consumer key and secret) may be requested via the Garmin Connect Developer Portal.

## 2.5. API Rate Limiting or Excessive Usage

To manage capacity and ensure system stability, Garmin Connect Courses API implementations may be subject to rate limiting. If any of the following limits are problematic for your implementation, please contact [connect-support@developer.garmin.com](mailto:connect-support@developer.garmin.com) to discuss alternatives.

Please plan the implementation with the following limitations in mind:

### **Evaluation Rate Limits**

- 100 API call requests per partner per minute - a rolling 60 second window summing the Oauth requests and API calls.
- 200 API call requests per user per day - a rolling 24-hour window excluding Oauth requests.

### **Production Rate limits**

- 6000 API call requests per partner per minute - a rolling 60 second window summing the Oauth requests and API calls.
- 6000 API call requests per user per day - a rolling 24-hour window excluding Oauth requests.

If one or both of the above limits are exceeded by a partner or a specific user, the subsequent API call request attempts will receive an HTTP Status Code 429 (too many requests). The call or calls in question will need to be attempted again later.

## 3. Courses API Endpoint Details

### 3.1. Courses API Permissions

Consumer would have “Course Import” permission set up with Garmin Connect Courses API (if partner is using other APIs, user will have multiple permissions). User while signing up may only opt in for fewer permissions, so this endpoint helps in fetching the permissions for that particular user.

Example response for this endpoint:

```
{ [
  "COURSE_IMPORT"
]}
```

Method & URL: GET <https://apis.garmin.com/userPermissions/>

Response body: The retrieved user permissions in JSON.

Response code:

HTTP Response Status	Description
200	User Permissions retrieved
401	Unauthorized
429	Quota violation / rate-limiting

## 3.2. Courses

### 3.2.1 Field Definitions

Course	Data Type	Description
courseId	Long	a unique identifier for the Courses. This field is not necessary for create operations and will be set automatically.
elapsedSeconds	Double	number of elapsed seconds.

CourseDetails	Data Type	Description
courseName	String	the name of the course, this cannot be empty
description	String	the description of the course
distance	Double	total distance of the course in meters, this cannot be null
elevationGain	Double	total elevation gain of the course in meters and cannot be null
elevationLoss	Double	total elevation loss of the course in meters and cannot be null
geoPoints	List<GeoPoint>	a list of geo-points that constitutes the courses and cannot be empty
activityType	String	activity type of the course, valid Values: RUNNING, HIKING, OTHER, MOUNTAIN_BIKING, TRAIL_RUNNING, ROAD_CYCLING, GRAVEL_CYCLING
speedMetersPerSecond	Double	speed of the total course in meters per second
coordinateSystem	String	valid co-ordinate system values: WGS84, GCJ02, BD09

GeoPoint	Data Type	Description
latitude	Double	latitude for the geo-point
longitude	Double	longitude for the geo-point
elevation	Double	elevation for the geo-point
information	CoursePoint	coursePoint is informational point



CoursePoint	Data Type	Description
coursePointType	String	valid CoursePointType values: GENERIC, SUMMIT, VALLEY, WATER, FOOD, DANGER, FIRST_AID, HORS_CATEGORIE, FOURTH_CATEGORY, THIRD_CATEGORY, SECOND_CATEGORY, FIRST_CATEGORY, SPRINT, SEGMENT_START, SEGMENT_END, SHAPING, CAMPSITE, AID_STATION, REST_AREA, GENERAL_DISTANCE, SERVICE, ENERGY_GEL, SPORTS_DRINK, MILE_MARKER, CHECKPOINT, SHELTER, MEETING_SPOT, OVERLOOK, TOILET, SHOWER, GEAR, SHARP_CURVE, STEEP_INCLINE, TUNNEL, BRIDGE, OBSTACLE, CROSSING, STORE, TRANSITION, NAVAID, TRANSPORT, ALERT, INFO
segmentUuid	String	uniqueId for identifying segment, applicable only when CoursePointType is SEGMENT_START, SEGMENT_END

### 3.2.2. Example JSON

```
{
  "courseId": 30626618,
  "courseName": "olathe gravel cycling",
  "distance": 8561.08,
  "elevationGain": 115.27,
  "elevationLoss": 4.44,
  "geoPoints": [{
    "latitude": 46.425274,
    "longitude": 11.685595,
    "elevation": 1300.0
  },
  {
    "latitude": 46.426752,
    "longitude": 11.684781,
    "elevation": 1300.0
  },
  {
    "latitude": 46.429178,
    "longitude": 11.68578,
    "elevation": 1300.4
  },
  {
    "latitude": 46.430211,
    "longitude": 11.685834,
    "elevation": 1304.2
  },
  {
    "latitude": 46.430463,
    "longitude": 11.685674,
    "elevation": 1305.7
  },
  {
    "latitude": 46.474929,
    "longitude": 11.745668,
```

```

        "elevation": 1410.9
      },
      {
        "latitude": 46.474809,
        "longitude": 11.745766,
        "elevation": 1410.4
      },
      {
        "latitude": 46.474929,
        "longitude": 11.745668,
        "elevation": 1410.9
      },
      {
        "latitude": 46.474929,
        "longitude": 11.745668,
        "elevation": 0.0,
        "distance": 13698.295,
        "information": {
          "name": "water",
          "coursePointType": "WATER"
        }
      }
    ],
    "activityType": "GRAVEL_CYCLING",
    "coordinateSystem": "WGS84"
  }
}

```

### 3.2.3. Create

This request is to *create* a Course

Method & URL: POST <https://apis.garmin.com/training-api/courses/v1/course>

Request body: the new course in JSON. A course ID should not be included.

Content-Type: application/json

Response Body: The newly created course as JSON.

Response code:

HTTP Response Status	Description
200	Course creates successfully
401	User access token doesn't exist
412	User permission error
429	Quota violation / rate-limiting

### 3.2.4. Retrieve

This request is to *retrieve* a Course

Method & URL: GET <https://apis.garmin.com/training-api/courses/v1/course/{courseid}>

Response Body: The retrieved course in JSON.

Response code:

HTTP Response Status	Description
200	Course successfully retrieved
401	User access token doesn't exist
412	User permission error
429	Quota violation / rate-limiting

### 3.2.5. Update

This request is to *update* a Course

Method & URL: PUT <https://apis.garmin.com/training-api/courses/v1/course/{courseid}>

Request body: The full updated course in JSON.

Content-Type: application/json

Response Body: The newly created course as JSON.

Response code:

HTTP Response Status	Description
204	Course successfully updated
401	User access token doesn't exist
404	Not Found
412	User permission error
429	Quota violation / rate-limiting

### 3.2.6. Delete

This request is to *delete* a Course

Method & URL: DELETE <https://apis.garmin.com/training-api/courses/v1/course/{courseid}>

Response code:

HTTP Response Status	Description
204	Course successfully updated
401	User access token doesn't exist
412	User permission error
429	Quota violation / rate-limiting

### 3.2.7 Acceptance Criteria

- As a best practice, partners should provide course points every 100 meters to align with the Garmin Connect course point and elevation calculations. If a partner does not, Garmin Connect will fill in the elevation.
- If the elevation is provided by the partner- it will be used. When not provided – Garmin Connect (GC) will use corrected elevation to supplement.
- Partners should provide segment start and end and a valid UUID, to consider as a valid segment in the course path.
- The courses created from 3rd party will be visible in GC; these courses will be directly synced to the device. The intent of the API is to fundamentally support individual route syncs. However, up to 50 courses may be synced in a single instance. This limit is set because the courses automatically sync to devices, which can cause undesirable sync times to devices.
- For devices that don't support navigations by default, GC calculates navigations and sends it to the FIT file. We don't provide API access for the partners to send us the navigations (right turn, left turn, etc.)

#### *Limitations with regards to size of a course*

- For exporting tcx or gpx files to devices (older devices which are more than 6 years old and only supports tcx and gpx files), there is a limitation of 100 miles for a course.
- We only support embedded turn navigation in the Course FIT file for courses that are 200 miles or less. Some entry-level devices require these turn geo-points to utilize turn-by-turn navigation. Devices with preloaded navigable maps won't need embedded turns to use turn-by-turn navigation.
- Partners can send us around 10,000 geo-points, given any two consecutive geo-points are not more than 100 meters apart, which limits the course to be around 600 miles.