Authentication

Every request to the Planning Center API is made on behalf of a user. Permissions are dependent on the access of this user. This also means that when the access a user is granted has changed, their permission in the API will change as well.

There are two ways to authenticate with the API: **Personal Access Token** and **OAuth**.

Personal Access Token and OAuth secrets should never be posted publicly. If you need to revoke your tokens, visit your developer accountand delete the impacted tokens. If at any time they are found to be publicly available they will be disabled automatically.

Personal Access Token

A Personal Access Token (PAT) allows you to use the API from your own account. This is useful if you are writing a script that runs on your own computer or a server you control. These tokens cannot be used in client-side requests from a web page.

You may not use a Personal Access Token if you are integrating with multiple churches—instead use OAuth (see below).

Important: Never share your Personal Access Token with another user or company. It is the equivalent of giving someone your username and password.

1. Visit your developer account.
2. Create a new "Personal Access Token"
3. Pass your token and secret in every request using HTTP Basic Auth.

You can pass your personal access token with `curl` like this:

```
curl -u app_id:secret https://api.planningcenteronline.com/people/v2/people
```

OAuth 2.0

If you are distributing your app to multiple churches, you should use OAuth version 2.0. First, your company needs to have a Planning Center account, which represents your organization. You can create one here. This feels a little funny at first, because it is like you are signing up as a church for Planning Center. But having a Planning Center account with a login gives you a place to manage your OAuth application. The name you give your Planning Center account (the "church" name) can just be your company name.

Once you are logged into your Planning Center account, visit api.planningcenteronline.com to create an OAuth application. This client ID and secret will only be created once, and will be reused for every church that signs up for your integration.

Important: Only create one OAuth application. When using OAuth, churches do **not** need to create their own OAuth application. You can integrate with multiple churches using your single OAuth application that you control.

Starting March 22, 2023, only Organization Administrators can create OAuth applications, and all new applications are managed by all Organization Administrators in your organization. If you are part of a church you should consider creating a new free organization to hold your application if the application doesn't belong to that church.

Make sure your app conforms to the OAuth 2.0 specification. Generally, to authenticate a user, follow these steps:

1. Redirect the user's browser to `https://api.planningcenteronline.com/oauth/authorize?client_id=CLIENT_ID&redirect_uri=https://example.com/auth/complete&response_type=code&scope=people` replace `CLIENT_ID` and `https://example.com/auth/complete` with your actual redirect URI).

   If you need different scope, replace `scope=people` appropriately (see "Scopes" section below).

2. Planning Center will redirect the user's browser back to the given redirect URI with a code param.
3. Send a POST request in the background to `https://api.planningcenteronline.com/oauth/token` with the following params:

```
{"grant_type": "authorization_code", "code": "CODE_FROM_STEP_2", "client_id": "CLIENT_ID", "client_secret": "CLIENT_SECRET", "redirect_uri": "https://example.com/auth/complete"}
```

   (replace `CLIENT_ID`, `CLIENT_SECRET`, `CODE_FROM_STEP_2`, and the redirect URI appropriately).

```
curl -X POST https://api.planningcenteronline.com/oauth/token \
              -F grant_type=authorization_code \
              -F code=1234567890 \
              -F client_id=2345678901 \
              -F client_secret=3456789012 \
              -F redirect_uri=https://example.com/auth/complete
```

4. The response you get back will contain the access token and other information.

```
{
  "access_token": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef",
  "token_type": "bearer",
  "expires_in": 7200,
  "refresh_token": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef",
  "scope": "people",
  "created_at": 1469553476
```

```
}
```

12. Use the access token for all API requests by passing it in the `Authorization` header, using the `Bearer` authentication scheme.

```
curl -H 'Authorization: Bearer
1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef'
https://api.planningcenteronline.com/people/v2/people
```

OAuth access tokens expire after 2 hours from the time they are issued, but we also provide a refresh token you can use to get a new access token at any time, even after the access token expires, without forcing the user to re-authorize.

We will honor refresh tokens for up to 90 days after the date its associated access token was issued. This means that, if your app does not issue a refresh for 90 days, then your user will need to re-authorize.

We have some example apps showing how to obtain and use access tokens and refresh tokens.

- Flask + Python
- Sinatra + Ruby
- Slim + PHP

Scopes

Each Planning Center product is a distinct OAuth scope.

PRODUCT	SCOPE
Calendar	`calendar`
Check-Ins	`check_ins`
Giving	`giving`
Groups	`groups`
People	`people`
Publishing	`publishing`
Services	`services`

Note that accessing Webhooks endpoints do not require an OAuth scope.

Authorization URL is https://api.planningcenteronline.com/oauth/authorize Token URL is https://api.planningcenteronline.com/oauth/tokenWhen authorizing, you can request scopes using the `scope` parameter. The value should be a space-separated list of scopes. The value for `response_type` should be `code`.

Refreshing a token

You can refresh an expired Oauth token by using the `refresh_token` returned when you generated the token.

```
curl -X "POST" "https://api.planningcenteronline.com/oauth/token" \
  -H 'Content-Type: application/json' \
  -d $'{
    "client_id": "<your application client id>",
    "client_secret": "<your application secret>",
    "refresh_token": "<your access token refresh token>",
    "grant_type": "refresh_token"
  }'
```

The response body will look like this:

```
{
  "access_token": "<new access token>",
  "token_type": "bearer",
  "expires_in": 7200,
  "refresh_token": "<new refresh token>",
  "created_at": 1540325919
}
```

Current User

To get information about the current user, you can use the `/me` endpoint.

```
curl -H 'Authorization: Bearer ...' https://api.planningcenteronline.com/people/v2/me
```