



# CHURCHcommunity

B U I L D E R

## API Implementer's Guide

### Individual Profile Services

Introduction	1
Public Web Tools	2
Calendar Listing	
Group Finder	
Community Login	
Form Listing	
Mobile Carrier Listing	
Custom Field Label Listing	
Update Custom Field Labels	
Online Giving	
Lookup Table Services	
Where to Start	16
Accessing the Services	
Login & Password	

# Introduction



This implementer's guide provides information about the Church Community Builder Application Programming Interface (API) Group services.

This document will specify the necessary values for requests and responses to and from the API, and will provide sample code for accessing the API.

# Public Web Tools



The Church Community Builder Public Web Tools API allows you to extend your public Church Community Builder calendar to other websites, build a kiosk application that lists your small groups, or other tools you desire to create.

Access to the Public Web Tools is free for the asking for all church partners. However, you still need to have a login and password assigned from your application. All API services are protected by HTTP Basic Authentication. If your application can not provide credentials through Basic Authentication, then it will not be able to communicate with the API.

All service calls to the API require a parameter named `srv` that has the value of the service being called. All API services accept an optional parameter named `describe_api` with a value of 1. This will cause the service to return the values it accepts as parameters and the format of the service response.

The following services are available to all current Church Community Builder partners:

## Calendar Listing

The Calendar Listing service will take a range of two dates as input and produce a listing of all calendar events occurring in the given date range. If the two dates are identical, all events for the given day will be returned.

### Service Name

---

**public\_calendar\_listing**

### Required Parameters

---

**date\_start**

### Optional Parameters

---

**date\_end**

### Sample Request

---

```
curl 'http://localhost/api.php?srv=public_calendar_listing&date_start=2008-11-25' -u user:pwd
```

### Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
<request>
  <parameters>
    <argument name="srv" value="public_calendar_listing" />
    <argument name="date_start" value="2008-11-25" />
  </parameters>
</request>
<response>
<service>public_calendar_listing</service>
<service_action>execute</service_action>
<availability>public</availability>
<items count="1">
```

```

<item>
  <date>2008-11-25</date>
  <event_name ccb_id="607">adult choir</event_name>
  <event_description></event_description>
  <start_time>18:30</start_time>
  <end_time>18:30</end_time>
  <event_duration>0</event_duration>
  <event_type>Open To All</event_type>
  <location>Room 219</location>
  <group_name ccb_id="1">Ellerbe Road Church</group_name>
  <group_type>Church</group_type>
  <grouping_name>All Church</grouping_name>
  <leader_name ccb_id="1357">Hank H&#039;Arris</leader_name>
  <leader_phone>(760) 555-1212 </leader_phone>
  <leader_email>dev@ccbinc.us</leader_email>
</item>
</items>
</response>
</ccb_api>

```

## Group Finder

The Group Finder service allows you to pass in a set of criteria and returns a listing of the groups that best match the criteria.

Service Name

---

**group\_search**

Required Parameters

---

None

Optional Parameters

---

area_id	udf_pulldown_3_id	order_by_3_sort
childcare	limit_records_start	
meet_day_id	limit_records_per_page	
meet_time_id	order_by_1	
department_id	order_by_2	
type_id	order_by_3	
udf_pulldown_1_id	order_by_1_sort	
udf_pulldown_2_id	order_by_2_sort	

Sample Request

---

```
curl 'http://localhost/api.php?srv=group_search' -u user:pwd
```

Sample Response

---

```

<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>

```

```

<request>
  <parameters>
    <argument name="srv" value="group_search" />
  </parameters>
</request>
<response>
<service>group_search</service>
<service_action>execute</service_action>
<availability>public</availability>
  <items count="1">
    <item>
      <item_id type="groups">69</item_id>
      <id>69</id>
      <name>20 &amp; Something&#039;s Interest &#039;Here&#039;</name>
      <group_type_name ccb_id="11">Interest</group_type_name>
      <grouping_name></grouping_name>
      <area_name ccb_id="20">Edge Ministries</area_name>
      <meet_day_name></meet_day_name>
      <meet_time_name></meet_time_name>
      <description>Interested in EDGE?</description>
      <status_id>1</status_id>
      <messaging_type>Interact</messaging_type>
      <membership_type>Request</membership_type>
      <owner_name ccb_id="9">Marguerite Alder</owner_name>
      <owner_email_primary></owner_email_primary>
    </item>
  </items>
</response>
</ccb_api>

```

## Community Login

The Community Login service will return the individual ID for an individual in the Church Community Builder system. The parameters are the username and password assigned for the individual.

### Service Name

---

**individual\_id\_from\_login\_password**

### Required Parameters

---

**login**  
**password**

### Optional Parameters

---

**None**

### Sample Request

---

```
curl 'http://localhost/api.php?srv=individual_id_from_login_password&login=temporary&password=1temporary1' -u user:pwd
```

## Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
<request>
  <parameters>
    <argument name="srv" value="individual_id_from_login_password" />
    <argument name="login" value="temporary" />
    <argument name="password" value="1temporary1" />
  </parameters>
</request>
<response>
<service>individual_id_from_login_password</service>
<service_action>execute</service_action>
<availability>public</availability>
  <items count="1">
    <item>
      <item_id type="individual">1</item_id>
      <id>1</id>
    </item>
  </items>
</response>
</ccb_api>
```

## Form Listing

The Form Listing service returns a listing of all forms defined in the Church Community Builder system along with the web address to access that form.

### Service Name

---

**form\_list**

### Required Parameters

---

**None**

### Optional Parameters

---

**None**

### Sample Request

---

```
curl 'http://localhost/api.php?srv=form_list' -u user:pwd
```

## Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
<request>
  <parameters>
    <argument name="srv" value="form_list" />
```

```

    </parameters>
</request>
<response>
<service>form_list</service>
<service_action>execute</service_action>
<availability>public</availability>
  <items count="1">
    <item>
      <form_id>30</form_id>
      <form_title>New Form Filled Test</form_title>
      <form_description></form_description>
      <maximum_attendance>1</maximum_attendance>
      <url>http://dev.ccbchurch.com/w_form_response.php?form_id=30</url>
    </item>
  </items>
</response>
</ccb_api>

```

## Mobile Carrier Listing

The Mobile Carrier Listing service returns a listing of all mobile carriers defined in the Church Community Builder system along with the email domain used by the carrier.

Service Name

---

**mobile\_carrier\_list**

Required Parameters

---

**None**

Optional Parameters

---

**None**

Sample Request

---

```
curl 'http://localhost/api.php?srv=mobile_carrier_list' -u user:pwd
```

Sample Response

---

```

<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
  <request>
    <parameters>
      <argument name="srv" value="mobile_carrier_list" />
    </parameters>
  </request>
  <response>
    <service>mobile_carrier_list</service>
    <service_action>execute</service_action>
    <availability>public</availability>
    <mobile_carriers count="11">

```



```

<mobile_carrier id="1">
  <name>Alltel</name>
  <email>message.alltel.com</email>
</mobile_carrier>
<mobile_carrier id="2">
  <name>AT&T</name>
  <email>txt.att.net</email>
</mobile_carrier>
<mobile_carrier id="3">
  <name>Boost Mobile</name>
  <email>boostmobile.com</email>
</mobile_carrier>
<mobile_carrier id="4">
  <name>Sprint</name>
  <email>messaging.sprintpcs.com</email>
</mobile_carrier>
<mobile_carrier id="5">
  <name>T-Mobile</name>
  <email>tmomail.com</email>
</mobile_carrier>
<mobile_carrier id="6">
  <name>Verizon</name>
  <email>vtext.com</email>
</mobile_carrier>
<mobile_carrier id="7">
  <name>Virgin Mobile</name>
  <email>vobl.com</email>
</mobile_carrier>
<mobile_carrier id="8">
  <name>Cricket</name>
  <email>sms.mycricket.com</email>
</mobile_carrier>
<mobile_carrier id="9">
  <name>US Cellular</name>
  <email>email.uscc.net</email>
</mobile_carrier>
<mobile_carrier id="10">
  <name>Telus</name>
  <email>msg.telus.com</email>
</mobile_carrier>
<mobile_carrier id="11">
  <name>Bell Mobility</name>
  <email>txt.bellmobility.ca</email>
</mobile_carrier>
</mobile_carriers>
</response>
</ccb_api>

```

## Custom Field Label Listing

The Custom Field Label Listing service returns a listing of all custom field labels defined in the Church Community Builder system.

## Service Name

---

**custom\_field\_labels**

## Required Parameters

---

**None**

## Optional Parameters

---

**None**

## Sample Request

---

```
curl 'http://localhost/api.php?srv=custom_field_labels' -u user:pwd
```

## Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
  <request>
    <parameters>
      <argument name="srv" value="custom_field_labels" />
    </parameters>
  </request>
  <response>
    <service>custom_field_labels</service>
    <service_action>execute</service_action>
    <availability>public</availability>
    <custom_fields count="16">
      <custom_field>
        <name>udf_text_1</name>
        <label>T-Shirt Size</label>
        <admin_only>false</admin_only>
      </custom_field>
      <custom_field>
        <name>udf_text_2</name>
        <label>Second email</label>
        <admin_only>false</admin_only>
      </custom_field>
      <custom_field>
        <name>udf_text_3</name>
        <label></label>
        <admin_only>true</admin_only>
      </custom_field>
      <custom_field>
        <name>udf_text_4</name>
        <label>Test1</label>
        <admin_only>true</admin_only>
      </custom_field>
      <custom_field>
        <name>udf_text_5</name>
        <label></label>
```

```

    <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_text_6</name>
  <label></label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_date_1</name>
  <label>Date approved a</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_date_2</name>
  <label>Date began in m</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_date_3</name>
  <label>Date we receive</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_pulldown_1</name>
  <label>Deacon Cluster</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_pulldown_2</name>
  <label>TShirt Size</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_pulldown_3</name>
  <label>Do u believe me</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_group_pulldown_1</name>
  <label>Big Choices</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_group_pulldown_2</name>
  <label>Campus</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_group_pulldown_3</name>
  <label></label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_resource_pulldown_1</name>

```

```
        <label></label>
        <admin_only>false</admin_only>
    </custom_field>
</custom_fields>
</response>
</ccb_api>
```

## Update Custom Field Labels

The Update Custom Field Labels service will accept form-encoded data representing changes to the custom field labels and update the labels in the Church Community Builder system. NOTE: It is possible to set custom field labels for custom fields that the package you subscribe to does not allow you to use. The fields available to be set are udf\_text\_##\_label and udf\_text\_##\_admin, where ## can be 1-12; udf\_date\_##\_label and udf\_date\_##\_admin, where ## can be 1-6; udf\_pulldown\_##\_label and udf\_pulldown\_##\_admin, where ## can be 1-6; udf\_group\_pulldown\_##\_label and udf\_group\_pulldown\_##\_admin, where ## can be 1-3; udf\_resource\_pulldown\_1\_label and udf\_resource\_pulldown\_1\_admin.

This service MUST be invoked with an HTTP POST and the data for the new individual contained in the post data MUST be encoded in the mime format application/x-www-form-urlencoded.

### Service Name

---

**update\_custom\_field\_labels**

### Required Parameters

---

**None**

### Optional Parameters

---

**None**

### Sample Request

---

```
curl -u user:pwd "http://localhost/api.php?srv=update_custom_field_labels" -d "udf_text_1_label=Custom+Field+First&udf_text_1_admin=true"
```

### Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
  <request>
    <parameters>
      <argument name="srv" value="update_custom_field_labels" />
    </parameters>
  </request>
  <response>
    <service>update_custom_field_labels</service>
    <service_action>execute</service_action>
    <availability>public</availability>
    <custom_fields count="16">
      <custom_field>
        <name>udf_text_1</name>
        <label>Custom Field Fi</label>
```

```

    <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_text_2</name>
  <label>Second email</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_text_3</name>
  <label></label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_text_4</name>
  <label>Test1</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_text_5</name>
  <label></label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_text_6</name>
  <label></label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_date_1</name>
  <label>Date approved a</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_date_2</name>
  <label>Date began in m</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_date_3</name>
  <label>Date we receive</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_pulldown_1</name>
  <label>Deacon Cluster</label>
  <admin_only>true</admin_only>
</custom_field>
<custom_field>
  <name>udf_pulldown_2</name>
  <label>TShirt Size</label>
  <admin_only>>false</admin_only>
</custom_field>
<custom_field>
  <name>udf_pulldown_3</name>

```

```

    <label>Do u believe me</label>
    <admin_only>true</admin_only>
  </custom_field>
  <custom_field>
    <name>udf_group_pulldown_1</name>
    <label>Big Choices</label>
    <admin_only>>false</admin_only>
  </custom_field>
  <custom_field>
    <name>udf_group_pulldown_2</name>
    <label>Campus</label>
    <admin_only>>false</admin_only>
  </custom_field>
  <custom_field>
    <name>udf_group_pulldown_3</name>
    <label></label>
    <admin_only>>false</admin_only>
  </custom_field>
  <custom_field>
    <name>udf_resource_pulldown_1</name>
    <label></label>
    <admin_only>>false</admin_only>
  </custom_field>
</custom_fields>
</response>
</ccb_api>

```

## Online Giving

The Online Giving service allows a gift to be entered directly into the Church Community Builder system. This service requires the id of the individual providing the gift, the category id the gift is applied to, and the amount of the gift. A separate service is available to get a list of categories and IDs from the Church Community Builder system. The Community Login service can be used in conjunction with the Online Giving service to allow your church to have a page where the user provides their login credentials along with information for the gift. Your users never have to know their own individual id, and gifts don't have to go through the normal matching process because the software already knows who is giving the gift.

**Be advised:** This service does **NOT** process any payments. It will merely record a new payment that has been processed outside the Church Community Builder system into the Online Gifts section of the Financial Admin page.

The payment\_method\_type is an optional parameter that specifies the credit card type of the gift. The valid values for this parameter are MC, VISA, DISC, AMEX, or Other.

## Service Name

---

**online\_giving\_insert\_gift**

## Required Parameters

---

**coa\_category\_id**  
**individual\_id**  
**amount**

## Optional Parameters

---

<b>merchant_transaction_id</b>	<b>last_name</b>	<b>zip</b>
<b>merchant_authorization_code</b>	<b>street_address</b>	<b>email</b>
<b>merchant_notes</b>	<b>city</b>	<b>phone</b>
<b>first_name</b>	<b>state</b>	<b>payment_method_type</b>

## Sample Request

---

```
curl 'http://localhost/api.php?srv=online_giving_insert_gift&coa_category_id=8&individual_id=1&amount=12.85' -u user:pwd
```

## Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
<request>
  <parameters>
    <argument name="srv" value="online_giving_insert_gift" />
    <argument name="coa_category_id" value="8" />
    <argument name="individual_id" value="1" />
    <argument name="amount" value="12.85" />
  </parameters>
</request>
<response>
<service>online_giving_insert_gift</service>
<service_action>execute</service_action>
<availability>public</availability>
  <items count="1">
    <item>
      <gift_id>10</gift_id>
      <confirmation_code>FQR</confirmation_code>
      <amount>12.85</amount>
      <coa_category ccb_id="8">General Offering</coa_category>
      <individual ccb_id="1">Master Admin</individual>
      <email>free@churchcommunitybuilder.com</email>
      <address>Family Home 2050 Rancho Corte</address>
      <city>Vista</city>
      <state>CA</state>
      <postal_code>12345-6789</postal_code>
      <phone_number>(719) 555-1211 </phone_number>
    </item>
  </items>
</response>
</ccb_api>
```

```
</response>
</ccb_api>
```

Lookup Table Services

Services are provided to get the contents of different lookup value lists from the Church Community Builder system. These lookup values can come from the following areas:

- Group Types
  - Group Meeting Days
  - Group Meeting Times
  - User Defined Fields for Groups
- COA Categories
  - Group Departments
  - Group Areas of Town

Service Name

ability_list	meet_day_detail	udf_grp_pulldown_1_list
ability_detail	meet_time_list	udf_grp_pulldown_1_detail
activity_list	meet_time_detail	udf_grp_pulldown_2_list
activity_detail	membership_type_list	udf_grp_pulldown_2_detail
age_bracket_list	membership_type_detail	udf_grp_pulldown_3_list
age_bracket_detail	music_style_list	udf_grp_pulldown_3_detail
area_list	music_style_detail	udf_ind_pulldown_1_list
area_detail	passion_list	udf_ind_pulldown_1_detail
church_service_list	passion_detail	udf_ind_pulldown_2_list
church_service_detail	reason_left_church_list	udf_ind_pulldown_2_detail
department_list	reason_left_church_detail	udf_ind_pulldown_3_list
department_detail	school_list	udf_ind_pulldown_3_detail
event_grouping_list	school_detail	udf_ind_pulldown_4_list
event_grouping_detail	significant_event_list	udf_ind_pulldown_4_detail
gift_list	significant_event_detail	udf_ind_pulldown_5_list
gift_detail	spiritual_maturity_list	udf_ind_pulldown_5_detail
how_joined_church_list	spiritual_maturity_detail	udf_ind_pulldown_6_list
how_joined_church_detail	style_list	udf_ind_pulldown_6_detail
how_they_heard_list	style_detail	udf_resource_pulldown_1_list
how_they_heard_detail	transaction_grouping_list	udf_resource_pulldown_1_detail
meet_day_list	transaction_grouping_detail	

Required Parameters

List service: None  
Detail service: «base service name»\_id

Optional Parameters

None

Sample Request

```
curl 'http://localhost/api.php?srv=meet_day_list' -u user:pwd
curl 'http://localhost/api.php?srv=meet_day_detail&meet_day_id=1' -u user:pwd
```



## Sample Response

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
<request>
  <parameters>
    <argument name="srv" value="meet_day_list" />
  </parameters>
</request>
<response>
<service>meet_day_list</service>
<service_action>execute</service_action>
<availability>public</availability>
  <items count="24">
    <item>
      <item_id type="meet_day">1</item_id>
      <id>1</id>
      <name>Varies</name>
      <order>1</order>
    </item>
  </items>
</response>
</ccb_api>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ccb_api>
<request>
  <parameters>
    <argument name="srv" value="meet_day_detail" />
    <argument name="meet_day_id" value="1" />
  </parameters>
</request>
<response>
<service>meet_day_detail</service>
<service_action>execute</service_action>
<availability>public</availability>
  <items count="1">
    <item>
      <item_id type="meet_day">1</item_id>
      <id>1</id>
      <owner_id>1</owner_id>
      <name>Varies</name>
      <order_by>1</order_by>
      <editable>1</editable>
      <listed>1</listed>
      <creator_id>1</creator_id>
      <modifier_id>1</modifier_id>
      <date_created>2001-12-06 22:20:45</date_created>
      <date_modified>2001-12-06 22:20:45</date_modified>
    </item>
  </items>
</response>
</ccb_api>
```

# Where to Start



The Public Web Tools API is available on your existing Church Community Builder account. With the exception of the Online Giving service, all services are read-only and cannot damage any of your existing data. The Online Giving service will create new records in the Online Gift section of the software, so care should be taken when using that service.

After creating and assigning API accounts, use that information to create applications that communicate with your API.

## Accessing the Services

The API services can be accessed via the https (port 443) protocol. For the security of church partner data, Church Community Builder has chosen to encrypt all transmissions of data, instead of using the standard http (port 80) port.

The address to be used for accessing the services is displayed in the API Admin section of the Church Community Builder application.

## Login & Password

Logins and passwords are created and assigned through the API Admin section within the Church Community Builder application. Logins are assigned access to each service individually so control can be kept on what services are available if, for instance, the service login is for an external web development company.

This login and password is not the same as the Church Community Builder User Community account and will not allow a user access to the solution.