Online Promo code Request form

STEP 1: COMPLETE THE FOLLOWING FIELDS TO GET PROMO CODE AND DISCOUNT

Requester:

Enter a name of requester. This is the name that users will see when asked to access to your application.

Customer:

Enter the name of customer from which you want to approve the request.

Events:

Create an event and Set a date of events as well as add location of event.

Voucher Type:

- In this step, we will select the type of voucher. Here Method list consist of three options:
  - Multiple(same code for all)
  - Single (Unique, one time code)
  - Continual(automatic discount, no need to enter a code)
- Select the method of voucher from method list (Multiple, Single, Continual) which appears as shown.

  (Note: You can select only one method at a time)

Quantity: Select the quantity of product

Discount Amount:

- In this step, write the amount of discount that is offered.
- Select the discount amount options from list (Economy and Business in percentage form)

(Note: we can also select both options to avail discount amount of both categories.)

STEP 2: RESTRICTION

Purchase from:

Add purchase dates in this section by adding start and end time range for this category.

Outbound travel from:

Add Start and End date of outbound travel.

Inbound travel from:

Add Start and End date of inbound travel.

Airports from:

Add location of airports by filling country name and city name.

Airports To:

Add location of airports by filling country name and city name.

Excluding flight ranges:

- Excluding flight ranges comprises of different options (Economy Hot Deal Economy lite Economy Classic Economy Flex Business Lite Business Classic Business Flex First Classic).
- In this step, select excluding flight from the list.

(Note: You can select multiple options)

STEP 3: COMMENTS/SPECIAL REQUESTS

In this section, we will select type of passenger from the list (Child, Infant, and Adult).

STEP 4: JUSTIFICATION

Fill out justification box. This is the short justification of your app that users will see when asked to grant access to it.

STEP 5: SENT TO

In this section, you have to select the name of recipient to which you want to send request.

STEP 6: SUBMIT

Click on the submit button to send request.

Implementing an authorization/Approval flow in your application

Send the Requester to the authorization/approval page

First, your application has to send the requester to the authorization/approval page. The system asks the requester to authorize application. After the requester makes a choice, system sends the choice and a few other bits of information back to your application.

Main Steps:

1) Requester can add request.
2) Requester can view reject or resubmit requests.
3) Requester can view approved requests.

Sub-Steps:

- Requester add request to system by sending email for authorization.
- Your application further authorize the request.

- If the request is authorized, it is further send for approval, but if the request is rejected due to some certain condition, our application send email to requester to notify requester about rejected requests.
- Requestor can view the details of rejected requests with description and reason of rejection.
- Requestor can also resubmit requests by adding the requirements.
- The list of authorized request are further send for approval.
- Our application will also notify requestor about accepted and authorized requests by sending email to requester.
- Requestor can view approved request.
- This process continue, until your application fulfill all requirements of requester.

Research Task:

Four steps:

Step 1 - Send the user to the authorization page

To send the user to the authorization page
Add a link or button in your application that sends the user to the following URL:

```
https://{subdomain}.application-name.com/authorizations/new
```

where {subdomain} is your application subdomain. You can use either a POST or a GET request. Include the following parameters:
response_type - Required. application returns an authorization code in the response, so specify code as the response type. Example: response_type=code.
redirect_uri - Required. The URL that application should use to send the user's decision to grant access to your application. The URL has be absolute and not relative. It also has to be secure (https), unless you're using localhost or 127.0.0.1.
client_id - Required. The unique identifier you obtained when you registered your application with Zendesk. See the section above.
scope - Required. A space-separated list of scopes that control access to the application resources. You can request *read*, *write*, or *impersonate* access to all resources or to specific resources.
state - An arbitrary string included in the response from application after the user decides whether or not to grant access. You can use the parameter to guard against cross-site request forgery (CSRF) attacks. In a CSRF attack, the end user is tricked into clicking a link that performs an action in a web application where the end user is still authenticated. To guard against this kind of attack, add some value to the stateparameter and validate it when it comes back.
Make sure to URL-encode the parameters.
Example GET request

```
https://{subdomain}.application.com/authorizations/new?response_type=code&redirect_ur
```

Step 2 - Handle the requester authorization decision

Your application has to handle the response from application telling it what the user decided. The information is contained in URL parameters in the redirect URL.
If the user decided to grant access to the application, the redirect URL contains an authorization code. Example:

```
{redirect_url}?code=7xqwtlf3rrdj8uyeb1yf
```

The authorization code is valid only for a short time.
If the user decided not to grant access to the application, the redirect URL
contains error and error_description parameters that inform the app that the user denied access:

```
{redirect_url}?error=access_denied&error_description=The+end-
user+or+authorization+server+denied+the+request
```

Use these values to control the flow of your application. If the URL contains a code parameter, get an access token from application as described in the following section. This is the token to include in API calls to application.

Step 3 - Get an access token from application

If your application received an authorization code from application in response to the user granting access, your application can exchange it for an access token. To get the access token, make a POST request to the following endpoint:

```
https://{subdomain}.zendesk.com/oauth/tokens
```

Include the following required parameters in the request:
grant_type - Specify **authorization_code** as the value.
code - Use the authorization code you received from application after the user granted access.
client_id - Use the unique identifier you received when you registered your application with application.
client_secret - Use the Secret value you received when you registered your application with application.
redirect_uri - The same redirect URL as in step 3. For ID purposes only.
scope - Specify **read** as the value.
The request must be over https and the parameters must be formatted as JSON.

Using curl

```
curl https://{subdomain}. application.com/oauth/tokens \
  -H "Content-Type: application/json" \
  -d '{"grant_type": "authorization_code", "code": "{your_code}",
    "client_id": "{your_client_id}", "client_secret": "{your_client_secret}",
    "redirect_uri": "{your_redirect_url}", "scope": "read" }' \
  -X POST
```

Example response

```
Status: 200 OK

{
  "access_token": "gErypPlm4dOVgGRvA1ZzMH5MQ3nLo8bo",
  "token_type": "bearer",
  "scope":"read"
}
```

Step 4 - Use the access token in API calls

The app can use the access token to make API calls. Include the token in an HTTP Authorization header with the request, as follows:
Authorization: Bearer {a_valid_access_token}
For example, a curl request to list tickets would look as follows:

```
curl https://{subdomain}.zendesk.com/api/v2/tickets.json \
  -H "Authorization: Bearer gErypPlm4dOVgGRvA1ZzMH5MQ3nLo8bo"
```

Implicit grant flow

The implicit grant flow is similar to the authorization code grant flow except there's no step 3. You request a token instead of an authorization code. In other words, you set the value of the response_type parameter to "token" instead of "code". If the end user authorizes access, the token is sent immediately in the redirect URL. No endpoint exists to create the token or set its scope. The token grants read and write access to all resources.

Example request

```
https://{subdomain}.
application.com/oauth/authorizations/new?response_type=token&client_id={your_unique_identifier}&scope=read%20write
```

Example responses

If the user grants access to the application, the token is included in the redirect URL.

```
{redirect_url}#access_token=gErypPlm4dOVgGRvA1ZzMH5MQ3nLo8bo&token_type=bearer
```

If the user decides not to grant access to the application, the URL contains error and error_descriptionparameters.

```
{redirect_url}#error=access_denied&error_description=The+end-user+or+authorization+server+denied+the+request
```

Password grant type

Use the password grant type to exchange a application username and password for an access token directly. This grant type should only be used if your application can get application usernames and passwords. This is usually a highly privileged application with application The application should never store the usernames and passwords. It should also be highly secure about how it gets them.

Example request

```
curl https://{subdomain}. application.com/oauth/tokens \
  -H "Content-Type: application/json" \
  -d '{"grant_type": "password", "client_id": "{your_client_id}",
    "client_secret": "{your_client_secret}", "scope": "read",
    "username": "{ application _username}", "password": "{ application _password}"}' \
  -X POST
```

Example response

```
Status: 200 OK

{
  "access_token": "gErypPlm4dOVgGRvA1ZzMH5MQ3nLo8bo",
  "token_type": "bearer",
  "scope":"read"
}
```