**Zendesk Account Creation**

Creating an account on Zendesk involves a few simple steps. Here's how you can create an account:

**1. Go to Zendesk Website**

* Open your web browser and visit the Zendesk website: <https://www.zendesk.com/in/login/>.

**2. Click on "Get Started" or "Start Free Trial"**

* On the Zendesk homepage, look for the "Get Started" or "Start Free Trial" button (usually in the top-right corner).
* Zendesk typically offers a 30-day free trial for their service, so you can try it out before committing.

**3. Sign Up with Your Information**

* You'll be prompted to provide basic information such as:
  + **Your Name** (first and last name)
  + **Business Email Address** (this will be used for login and communication)
  + **Password** (choose a strong password)
  + **Company Name** (optional, depending on the plan)
  + **Phone Number** (optional)
* Enter all necessary details and proceed.

**4. Choose a Zendesk Plan**

* Zendesk offers different plans for customer service management, including a basic plan and various premium options.
* Select the plan you want to try or purchase.

**5. Set Up Your Domain**

* Zendesk will ask you to set up your domain, where your support team will be able to manage customer service tickets.
* You may be asked to choose a subdomain for your Zendesk account (e.g., yourcompany.zendesk.com).

**6. Customize Your Account**

* Once your account is created, you’ll be taken to your Zendesk dashboard, where you can customize settings, integrate with your systems, and start using the platform.
* You can configure your support channels (email, chat, etc.) and set up knowledge base articles.

**7. Invite Team Members (Optional)**

* If you're creating the account for a business, you may want to invite your team members to join. You can add agents and give them the appropriate access permissions.

**8. Start Using Zendesk**

* After setup, you can start using Zendesk to manage customer support, track tickets, and use other tools depending on your plan.

By following these steps, you'll have a fully functioning Zendesk account to start managing customer service requests.

**Zendesk API’s**

Creating an API token in Zendesk allows you to securely authenticate requests made to the Zendesk API. Here's how you can create a Zendesk API token:

**1. Log in to Your Zendesk Account**

* Open your web browser and go to your Zendesk account's login page (e.g., https://yoursubdomain.zendesk.com).
* Enter your admin credentials (email and password) to log in.

**2. Access the Admin Panel**

* Once logged in, click on the **Admin** icon (gear icon) located in the left-hand sidebar of the Zendesk dashboard.

**3. Navigate to the API Settings**

* In the Admin panel, scroll down and find the **Apps and Integrations** section.
* Click on Zendesk **API** under the Apps and Integrations section.

**4. Enable API Access**

* On the Zendesk API page, if it is not already enabled, toggle the switch to **Enable API**. This will allow you to use the Zendesk API and generate an API token.

**5. Generate a New API Token**

* Under the **Token Access** section, you will see an option to **Add API Token**.
* Click the **Add Token** button.
* A new token will be generated for you. Make sure to **copy the token** immediately, as it will not be visible again after you leave the page.

**6. Save the Token**

* Store the API token securely in a password manager or a secure location. It is essential for making authenticated API requests.

**7. Use the Token in API Requests**

* You can now use this API token in your API requests. The token is used in place of a password when making API calls.
* When authenticating via the API, use your **Zendesk email address** and the API token. The format for authentication is:
  + **Username:** your email address
  + **Password:** API token

**Example of Authentication Using the API Token:**

For example, using **cURL** to authenticate with an API request, you would do something like this:

Bash

curl -u your-email/token:your-api-token https://yoursubdomain.zendesk.com/api/v2/tickets.json

* Replace your-email with your Zendesk login email.
* Replace your-api-token with the token you created.
* Replace yoursubdomain with your Zendesk account's subdomain.

By following these steps, you'll have successfully created and can use an API token to authenticate with Zendesk's API.   
Note:- (support links : <https://support.zendesk.com/hc/en-us/articles/4408889192858-Managing-access-to-the-Zendesk-API>)

**Zendesk Tickets**

To manage tickets in Zendesk using their API, you can use the GET, POST, PUT, and DELETE HTTP methods for ticket-related operations. Below are examples of how to create, update, get, and delete tickets using Zendesk's REST API.

### Prerequisites:

* You should have a **Zendesk account** with admin privileges.
* You should have created an **API token** (as explained in the previous response).
* You’ll need to authenticate using your **email** and **API token**.

### Authentication:

For all API requests, you’ll use basic authentication with your Zendesk **email** and **API token**. The syntax for authentication is:

* **Username:** your Zendesk email address (e.g., youremail@example.com)
* **Password:** the API token you generated

### Base URL:

Replace yoursubdomain with your actual Zendesk subdomain. The base URL for all API requests will be:

https://yoursubdomain.zendesk.com/api/v2

### 1. ****Create a Ticket (POST)****

To create a new ticket, you use the POST method to send a request to the /tickets endpoint.

#### Request:

bash

curl -u youremail@example.com/token:your-api-token \

-H "Content-Type: application/json" \

-X POST \

-d '{

"ticket": {

"subject": "Example Ticket",

"description": "This is a description of the ticket.",

"priority": "normal"

}

}' \

https://yoursubdomain.zendesk.com/api/v2/tickets.json

#### Explanation:

* **POST** to /api/v2/tickets.json
* The request body contains the ticket data (subject, description, priority).
* Replace youremail@example.com with your Zendesk email and your-api-token with your generated API token.

#### Response (Success):

{

"ticket": {

"id": 12345,

"subject": "Example Ticket",

"description": "This is a description of the ticket.",

"priority": "normal",

"status": "new",

"created\_at": "2024-12-11T12:00:00Z",

"updated\_at": "2024-12-11T12:00:00Z"

}

}

### 2. ****Get a Ticket (GET)****

To retrieve details of an existing ticket, use the GET method to access the /tickets/{ticket\_id} endpoint.

#### Request:

curl -u youremail@example.com/token:your-api-token \

https://yoursubdomain.zendesk.com/api/v2/tickets/12345.json

#### Explanation:

* **GET** from /api/v2/tickets/{ticket\_id}.json (replace 12345 with the actual ticket ID you want to retrieve).

#### Response:

{

"ticket": {

"id": 12345,

"subject": "Example Ticket",

"description": "This is a description of the ticket.",

"priority": "normal",

"status": "new",

"created\_at": "2024-12-11T12:00:00Z",

"updated\_at": "2024-12-11T12:05:00Z"

}

}

### 3. ****Update a Ticket (PUT)****

To update an existing ticket, you use the PUT method on the /tickets/{ticket\_id} endpoint.

#### Request:

curl -u youremail@example.com/token:your-api-token \

-H "Content-Type: application/json" \

-X PUT \

-d '{

"ticket": {

"subject": "Updated Ticket Subject",

"description": "Updated ticket description.",

"priority": "high"

}

}' \

https://yoursubdomain.zendesk.com/api/v2/tickets/12345.json

#### Explanation:

* **PUT** to /api/v2/tickets/{ticket\_id}.json (replace 12345 with the actual ticket ID).
* The request body contains the fields you want to update (subject, description, priority, etc.).

#### Response (Success):

{

"ticket": {

"id": 12345,

"subject": "Updated Ticket Subject",

"description": "Updated ticket description.",

"priority": "high",

"status": "open",

"created\_at": "2024-12-11T12:00:00Z",

"updated\_at": "2024-12-11T12:10:00Z"

}

}

### 4. ****Delete a Ticket (DELETE)****

To delete a ticket, you use the DELETE method on the /tickets/{ticket\_id} endpoint.

#### Request:

curl -u youremail@example.com/token:your-api-token \

-X DELETE \

https://yoursubdomain.zendesk.com/api/v2/tickets/12345.json

#### Explanation:

* **DELETE** from /api/v2/tickets/{ticket\_id}.json (replace 12345 with the actual ticket ID).
* This will permanently delete the ticket.

#### Response (Success):

{

"deleted": true

}

### Summary of Endpoints:

* **Create a Ticket:** POST /api/v2/tickets.json
* **Get a Ticket:** GET /api/v2/tickets/{ticket\_id}.json
* **Update a Ticket:** PUT /api/v2/tickets/{ticket\_id}.json
* **Delete a Ticket:** DELETE /api/v2/tickets/{ticket\_id}.json

By using these API calls, you can manage tickets in Zendesk programmatically.

Note:- (Support links : [https://developer.zendesk.com/api-reference/apps/apps-support-api/ticket\_sidebar/#ticket-object](https://developer.zendesk.com/api-reference/apps/apps-support-api/ticket_sidebar/%23ticket-object), <https://developer.zendesk.com/documentation/apps/build-an-app/build-your-first-support-app/part-4-getting-data/>, <https://developer.zendesk.com/documentation/apps/build-an-app/build-your-first-support-app/part-3-creating-and-inserting-templates/>, <https://developer.zendesk.com/api-reference/apps/apps-support-api/ticket_sidebar/>, <https://developer.zendesk.com/api-reference/apps/apps-core-api/client_api/>, <https://developer.zendesk.com/api-reference/apps/apps-core-api/core_api/> )

Above links are used to get the ticket and other details into the Custom App and also Zendesk API’s.

**Zendesk Custom App**

Creating a custom app in Zendesk using zcli (Zendesk Command Line Interface) involves several steps. The zcli tool allows you to develop, test, and deploy custom apps for Zendesk in a simplified way.

### Prerequisites:

1. **Zendesk account** with admin privileges.
2. **Zendesk Apps Framework (ZAF)** installed on your computer.
3. **Node.js** installed (since zcli depends on Node.js).
4. **Zendesk Developer Account** and access to the Zendesk App Framework (ZAF) environment.

**Command to for Zendesk custom APP**

-> npm install -g @zendesk/zcli // for installing zcli in locally

-> zcli apps:login -i // for connecting you account

-> zcli apps:new // to create new application locally

-> zcli apps:server --port=XXXX // to run the sever and test locally

-> zcli apps:create // Packaging and installing a private Zendesk app

-> zcli apps:update // Updating a private Zendesk app

### 1. ****Install**** zcli

First, you need to install zcli globally using npm (Node Package Manager).

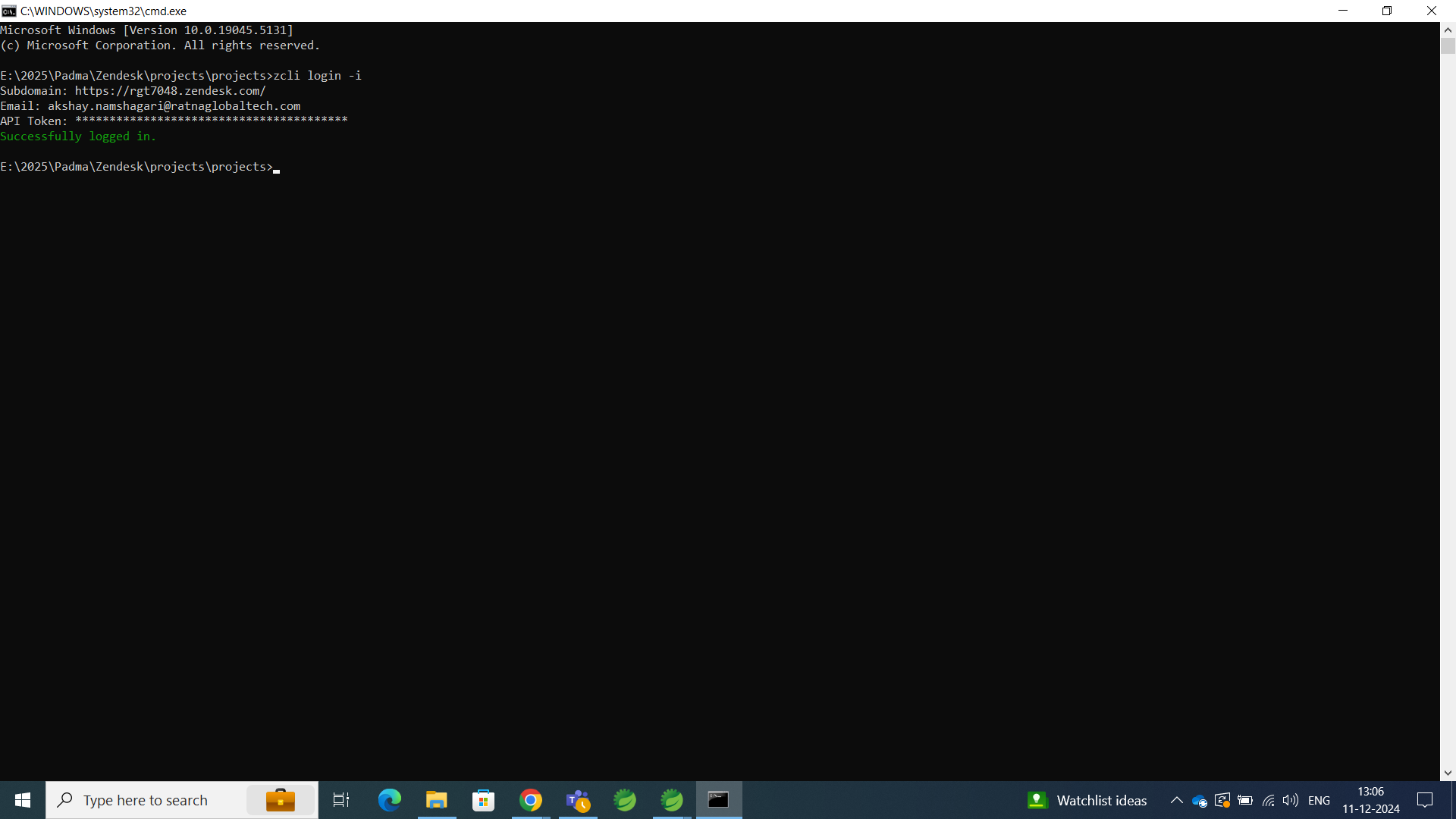
#### Command to install zcli:

npm install -g @zendesk/zcli

This will install the zcli command globally, making it available from any directory in your terminal.

### 2. ****Login to Zendesk with**** zcli

Once you have zcli installed, you need to authenticate it with your Zendesk account.



#### Command:

zcli apps:login -i

* You will be prompted to enter your Zendesk subdomain (e.g., yoursubdomain.zendesk.com).
* After that, you'll be asked to authenticate with your Zendesk credentials and your API token.

### 3. ****Create a New Zendesk App****

To create a new app, you can use the zcli command to scaffold a new app using the new command.

#### Command to create a new app:

zcli apps:new

This command will prompt you for the following:

* **App name**: Name your app (e.g., "My Custom App").
* **App description**: Provide a description for the app.
* **App type**: Choose the type of app (usually Custom App).

This will generate a new app template with the necessary structure and files. The basic file structure typically looks like this:

bash

Copy code

/my\_custom\_app

/assets

/dist

/templates

manifest.json

### 4. ****Develop the Custom App****

Once you have created the app, you can begin editing it to add functionality.

* **manifest.json**: This file describes the app and its configuration, such as where the app should be displayed and the app’s permissions.
* **assets/**: Any CSS or images for your app go here.
* **templates/**: If you're using handlebars or other templating engines, the templates go here.

You can start modifying these files to suit your app’s requirements. For example, you could add a button in the app's interface, listen to events, or make API calls to Zendesk.

### 5. ****Test the App Locally****

Before deploying, it's important to test your app locally to ensure it functions correctly.

#### Command to start the app locally:

zcli apps:server

This will start a local development server, and you'll get a URL to view your app within your Zendesk instance (e.g., http://localhost:4751). You can then access the app in your Zendesk interface and see changes in real-time.

### 6. ****Install the App in Your Zendesk Account****

Once you're happy with the development and testing of your app, you can install it in your Zendesk account.

To install your app locally in Zendesk:

#### Command:

zcli apps:install

This will upload your app to your Zendesk instance and install it. You’ll see your app in the **Manage Apps** section of Zendesk.

### 7. ****Deploy the App****

Once you're ready to deploy the app to your production Zendesk environment or share it, you can use the following command:

#### Command to deploy the app:

zcli apps:push

This command will push your app to the Zendesk App Marketplace (if you want to share it publicly) or to a private environment (if it's for internal use only).

If you're publishing your app to the marketplace, you might need to follow additional steps related to app review and approval.

### 8. ****Troubleshooting and Logs****

If you run into issues, you can inspect logs for more information. While running the app locally, the terminal will show logs for debugging purposes.

#### Command to view logs:

zcli apps:logs

This will display the logs for your app so you can identify any issues.

### Example Manifest (manifest.json)

Here's an example of a simple manifest.json that defines a custom app for the ticket sidebar:

{

  "name": "test",

  "author": {

    "name": "akshay",

    "email": "akshay.namshagari@ratnaglobaltech.com",

    "url": "https://rgt7048.zendesk.com/"

  },

  "defaultLocale": "en",

  "private": true,

  "location": {

    "support": {

      "ticket\_sidebar": {

        "url": "assets/iframe.html",

        "flexible": true

      }

    }

  },

  "version": "1.0.0",

  "frameworkVersion": "2.0"

}

### 9. ****Submit to Zendesk Marketplace (Optional)****

If you want to share your app with others or make it available for public use, you can submit it to the **Zendesk Marketplace**. To do this, you need to follow the app submission process, which involves preparing your app, writing a description, and going through a review process.

You can find more details about app submission in Zendesk's documentation.

### Summary of Commands:

* **Install zcli**: npm install -g @zendesk/zcli
* **Login**: zcli apps:login
* **Create New App**: zcli apps:new
* **Run the App Locally**: zcli apps:server
* **Install the App in Zendesk**: zcli apps:install
* **Deploy the App**: zcli apps:push
* **View Logs**: zcli apps:logs

By following these steps, you can easily create, test, and deploy custom apps for Zendesk using the zcli tool.

**Zendesk Custom APP**

To enable and install a custom app in your Zendesk account, follow these steps:

**Prerequisites:**

* You've already created your custom app using zcli or the Zendesk Apps Framework (ZAF).
* You have zcli installed and your app is developed and tested.
* You have admin access to your Zendesk account to install and enable apps.

**Steps to Enable and Install a Custom App in Zendesk:**

**1. Log in to Your Zendesk Account**

* First, ensure that you are logged into your Zendesk account with admin privileges.

**2. Access the Admin Settings**

* From your Zendesk dashboard, click on the **Admin** icon (the gear icon) located in the lower-left corner of the sidebar.

**3. Go to the "Manage Apps" Section**

* In the **Admin Center**, navigate to the **Apps and integrations** section. This may be listed under **Apps** or **Manage apps**, depending on your Zendesk interface.

**4. Install the Custom App Locally via zcli**

* If you've created the app using zcli, you can install it directly into your Zendesk account by running the following command from your terminal:

zcli apps:install

* This command will install the app to your Zendesk environment using the local app you've developed.
* Ensure that you are logged in to zcli with the correct Zendesk account and subdomain.

**5. Enable the Custom App in Zendesk**

After installing your app, follow these steps to enable it in your Zendesk interface:

1. **Go to the "Manage Apps" Section in Zendesk**:
   * In the **Admin Center**, find the section for managing apps (this might be listed as **Apps** or **Manage Apps**).
   * Under this section, you should see the **Installed Apps**.
2. **Locate Your Custom App**:
   * You should see your custom app listed there if the installation was successful. If not, refresh the page.
3. **Enable the App**:
   * Click on the **Settings** (gear icon) next to your custom app in the list of installed apps.
   * In the settings page for the app, you can enable it for the appropriate areas of Zendesk (e.g., ticket sidebar, agent interface, etc.).
   * Make sure to **activate or enable** the app as required. Some apps require configuration or permissions to work properly, so ensure these are set up before enabling.

**6. Test the App**

* Once the app is enabled, visit the relevant section (such as a ticket page) to ensure that your app works correctly.
* Check that your custom app appears in the desired location (ticket sidebar, admin interface, etc.) and functions as expected.

**7. Deploy the App (Optional)**

If you're satisfied with how the app is running locally, and you want to deploy it to production or share it, you can **push** the app to your Zendesk environment using:

zcli apps:push

This command will deploy the app to your live Zendesk account or to the Zendesk App Marketplace if you're sharing it.

**8. Revisit App Settings for Final Configuration**

If your app interacts with specific parts of the Zendesk interface, such as the **ticket sidebar**, you can ensure the app is properly displayed and accessible for agents or customers.

* Go back to **Admin Center** > **Apps and Integrations** > **Zendesk Support Apps** and configure any additional settings for your app.

**Troubleshooting:**

* If the app doesn't appear after installation, try refreshing your Zendesk instance or re-checking the installation process using zcli apps:install.
* If your app is not working correctly, review the logs in the terminal using zcli apps:logs for any error messages.

**Summary:**

* **Log in to Zendesk** and access the **Admin Center**.
* Use zcli apps:install to install the app locally into your Zendesk account.
* Go to **Manage Apps** in the **Admin Center** and enable the custom app.
* **Test** the app to ensure it functions as expected.
* Optionally, use zcli apps:push to deploy the app to your production Zendesk environment.

By following these steps, you should be able to enable and use your custom app within your Zendesk account.

**ZCLI Custom App Customization**

To get and use the **tickets JSON** within a custom app in Zendesk, you can leverage the Zendesk Apps Framework (ZAF) and the **Zendesk REST API** in combination with zcli. Below is a detailed guide on how to retrieve ticket data and use it in your custom app using zcli.

### Prerequisites:

1. You should have already created a custom app using zcli (as explained earlier).
2. You need the **ticket-related data** to be available within your app. This requires **API requests** to Zendesk to fetch the ticket data.

Example code:-

<script>

    // Initialise Apps framework client. See also:

    // https://developer.zendesk.com/apps/docs/developer-guide/getting\_started

    var client = ZAFClient.init();

    client.invoke('resize', { width: '100%', height: '200px' });

    client.context().then(function (context) {

      var settings = {

        url: 'https://rgt7048.zendesk.com/api/v2/tickets/' + context.ticketId + '.json',

        type: 'GET',

        dataType: 'json',

      };

      client.request(settings).then(

        function (data) {

          console.log(data);

        },

        function (response) {

          console.error(response);

        }

      );

    });

</script>

### Steps to Get and Use Tickets JSON in Your Custom App Code

### 1. ****Install ZCLI and Create Your Custom App (If Not Done Already)****

First, ensure that zcli is installed and you have a custom app scaffolded. If you haven't done this yet:

npm install -g @zendesk/zcli

zcli apps:new

This will generate a new custom app template.

### 2. ****Understand App Context in Zendesk****

When you build a custom app for Zendesk using ZAF (Zendesk Apps Framework), the app interacts with the Zendesk interface via **contexts**. For example, you might want to display ticket information on the **ticket sidebar** or within the **ticket interface**.

Zendesk provides a **client object** in the app that can be used to interact with the interface, including accessing ticket data. The key part here is using the **ZAF client API** to retrieve ticket data.

### 3. ****Modify Your App Code to Access Ticket Data****

To interact with ticket data, your custom app will need to use the ZAF client to retrieve information about the current ticket. Here's how to fetch and use ticket JSON data:

#### Modify app.js to Fetch Ticket Data:

In your app’s main JavaScript file (app.js), you will use the client object to access ticket data. The ZAF client makes it easy to access the current ticket data, including JSON format.

javascript

// iframe.html => inside Script tag

(function() {

// Initialize ZAF client

var client = ZAFClient.init();

// Listen for the ticket context when the app is loaded

client.invoke('ticketFields').then(function(ticketFields) {

console.log(ticketFields); // You can inspect ticket fields here

// You can also fetch other ticket details

client.get('ticket').then(function(ticketData) {

// Here we have the full ticket data as a JSON object

console.log(ticketData);

// For example, we can access the ticket's subject

var ticketSubject = ticketData.ticket.subject;

document.getElementById('ticketSubject').innerText = ticketSubject;

// We can also access other parts of the ticket data, like status or priority

var ticketStatus = ticketData.ticket.status;

var ticketPriority = ticketData.ticket.priority;

console.log(`Ticket Status: ${ticketStatus}, Ticket Priority: ${ticketPriority}`);

});

});

})();

#### Explanation:

* **ZAFClient.init()** initializes the Zendesk Apps Framework client.
* **client.invoke('ticketFields')** will provide details about ticket fields.
* **client.get('ticket')** retrieves the entire **ticket JSON** object, which contains details such as:
  + ticket.subject
  + ticket.status
  + ticket.priority
  + ticket.id
  + Other ticket-related data

You can use this data to display ticket information dynamically within your app or manipulate it as needed.

### 4. ****Make Custom API Calls (Optional)****

If you want to **fetch additional data** related to the ticket (e.g., ticket comments, custom fields, etc.) or make a custom API request, you can use the **Zendesk REST API** within your app.

Here is an example of how you can make a **custom API request** to retrieve ticket comments and use them in your app:

(function() {

var client = ZAFClient.init();

// Use the client to get the current ticket data

client.get('ticket').then(function(ticketData) {

var ticketId = ticketData.ticket.id;

// Use the Zendesk REST API to get comments for this ticket

var apiUrl = '/api/v2/tickets/' + ticketId + '/comments.json';

client.request({

url: apiUrl,

type: 'GET'

}).then(function(response) {

console.log(response); // The response contains the comments in JSON format

// You can then display the comments in your app

var comments = response.comments;

comments.forEach(function(comment) {

console.log(comment.body); // Print each comment body

// Optionally, append to the DOM or use as needed

});

}).catch(function(error) {

console.error('Error fetching comments:', error);

});

});

})();

#### Explanation:

* **client.request()** is used to make a REST API call to the Zendesk API to fetch ticket comments (/tickets/{ticket\_id}/comments.json).
* The response contains the **comments JSON**, which you can loop through and display as needed in your app.

### 5. ****Test the App Locally****

To test your custom app and see if it correctly fetches and uses the ticket JSON, you can run the app locally using zcli:

zcli apps:server

This will start a local development server. You can then access your app within your Zendesk account (usually on the ticket sidebar or another location depending on the manifest.json configuration).

### 6. ****Install and Enable the App****

Once you are happy with your app, you can install it in your Zendesk account:

zcli apps:install

This command will upload your app to your Zendesk instance. After installation, enable it through the **Admin Center** (Apps > Manage Apps), and it should start working within your Zendesk interface.

### 7. ****Deploy the App****

If you want to deploy the app to production, use the following command:

zcli apps:push

This will push the app to your production Zendesk environment.

### Example of Manifest for Ticket Sidebar Location (manifest.json):

{

  "name": "test",

  "author": {

    "name": "akshay",

    "email": "akshay.namshagari@ratnaglobaltech.com",

    "url": "https://rgt7048.zendesk.com/"

  },

  "defaultLocale": "en",

  "private": true,

  "location": {

    "support": {

      "ticket\_sidebar": {

        "url": "assets/iframe.html",

        "flexible": true

      }

    }

  },

  "version": "1.0.0",

  "frameworkVersion": "2.0"

}

### Summary:

1. **Access ticket data** using client.get('ticket') to retrieve the ticket's JSON object.
2. You can **make custom API requests** (e.g., to fetch comments) using client.request().
3. **Test** the app locally using zcli apps:server.
4. **Install** and **enable** the app in Zendesk using zcli apps:install.
5. **Deploy** the app to your production environment using zcli apps:push.

By following these steps, you can fetch ticket data and use it within your custom app in Zendesk using zcli.

Note:- After running the application locally then only it will work in Zendesk Application.