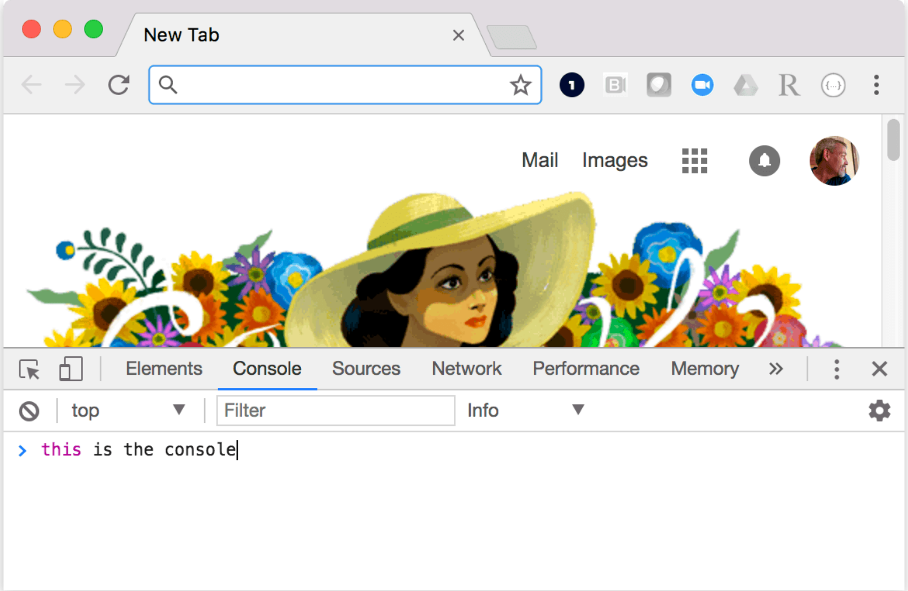
Ticketing API quick start

ON THIS PAGE

In this 10-minute quick start, you'll use the Zendesk Ticketing API to create a few tickets in Zendesk Support.

The tutorial is intended for admins and agents with no developer experience. To make the API requests, the tutorial uses your web browser's built-in JavaScript console.



```
> this is the console
```

If you're a more experienced developer, check out the API reference. The reference includes curl examples for each endpoint. The API reference also includes links to related Postman collections. See Exploring Zendesk APIs with Postman.

What you'll need

To complete this tutorial, you'll need the following:

- A Zendesk Support account with agent or admin access. To get a free account for testing, see Getting a trial or sponsored account for development.
- API password access is enabled for the account. See Managing password access to the API in Zendesk help.
- A web browser, such as Chrome, Firefox, Safari, or Microsoft Edge. These browsers include a built-in JavaScript console.

Creating tickets

To create a ticket, use the JavaScript console to make an HTTP POST request to the Ticketing API's Create Ticket endpoint.

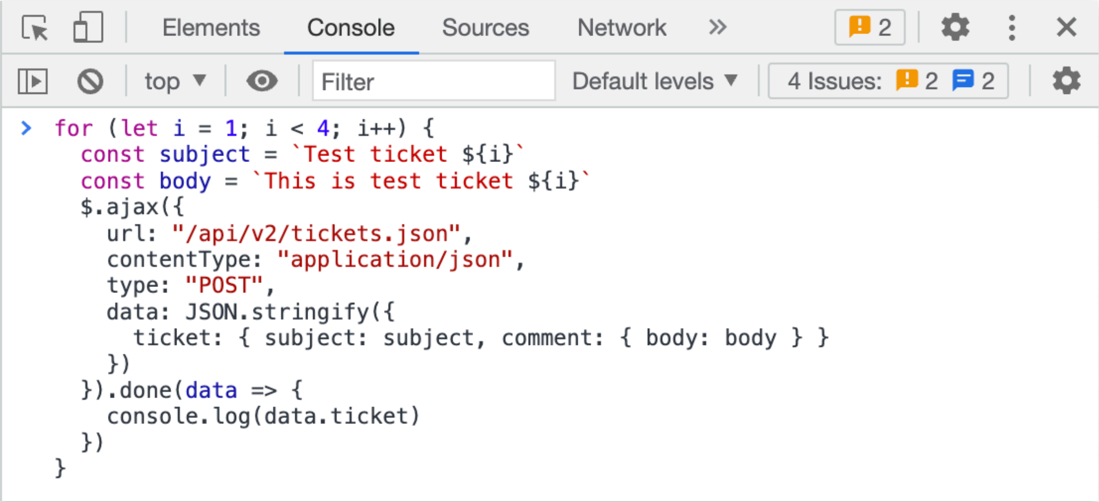
1. Sign in to your Zendesk account as an admin or agent. From the Zendesk Agent Workspace, open your browser's JavaScript console:

    - Chrome: **View** > **Developer** > **JavaScript Console**
    - Firefox: **Tools** > **Browser Tools** > **Web Developer Tools**. Then click the **Console** tab.
    - Safari: **Develop** > **Show JavaScript Console**
    - Microsoft Edge: **Tools** > **Developer** > **JavaScript Console**

2. Paste the following script into your browser's console.

```
for (let i = 1; i < 4; i++) {
  const subject = `Test ticket ${i}`
  const body = `This is test ticket ${i}`
  $.ajax({
    url: "/api/v2/tickets.json",
    contentType: "application/json",
    type: "POST",
    data: JSON.stringify({
      ticket: { subject: subject, comment: { body: body } }
    })
  }).done(data => {
    console.log(data.ticket)
  })
}
```

   Here's what the code looks like after pasting it into Google Chrome's JavaScript console:

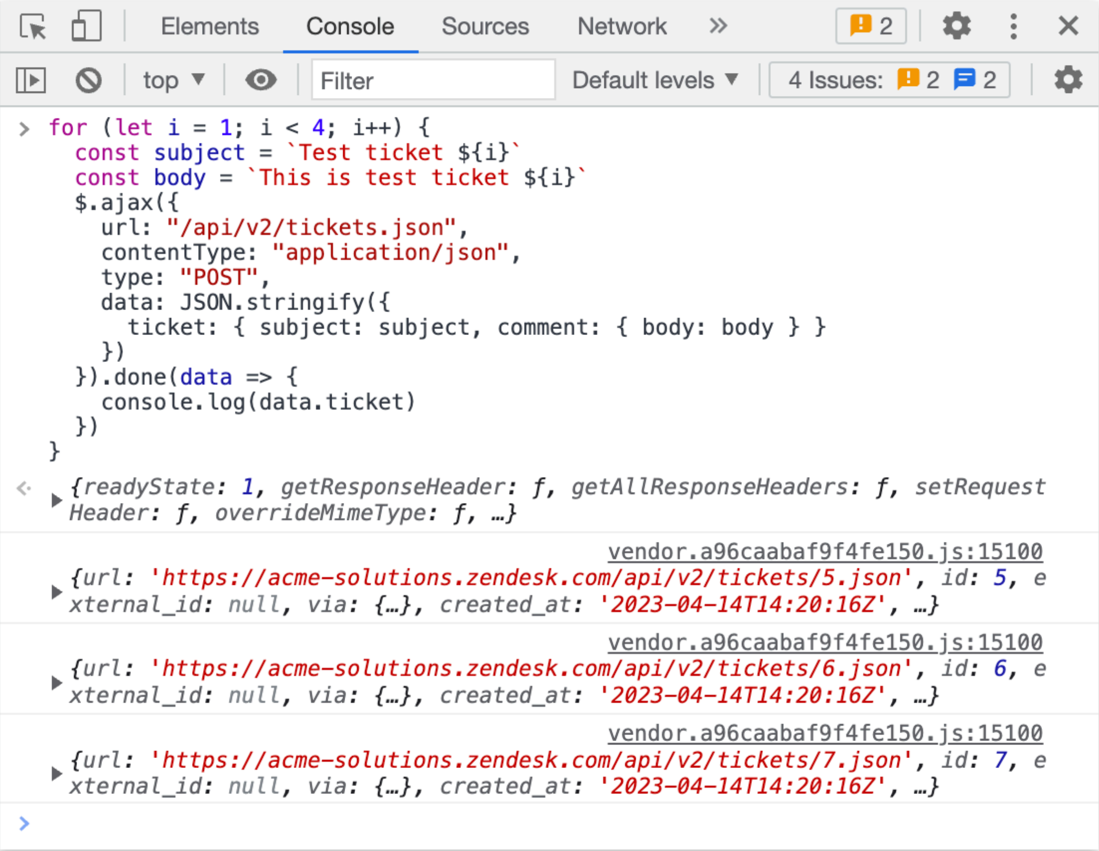


```
> for (let i = 1; i < 4; i++) {
    const subject = `Test ticket ${i}`
    const body = `This is test ticket ${i}`
    $.ajax({
      url: "/api/v2/tickets.json",
      contentType: "application/json",
      type: "POST",
      data: JSON.stringify({
        ticket: { subject: subject, comment: { body: body } }
      })
    }).done(data => {
      console.log(data.ticket)
    })
  }
```

The code snippet creates a loop that makes three requests to the Create Ticket endpoint. Each request creates a ticket in Zendesk Support.

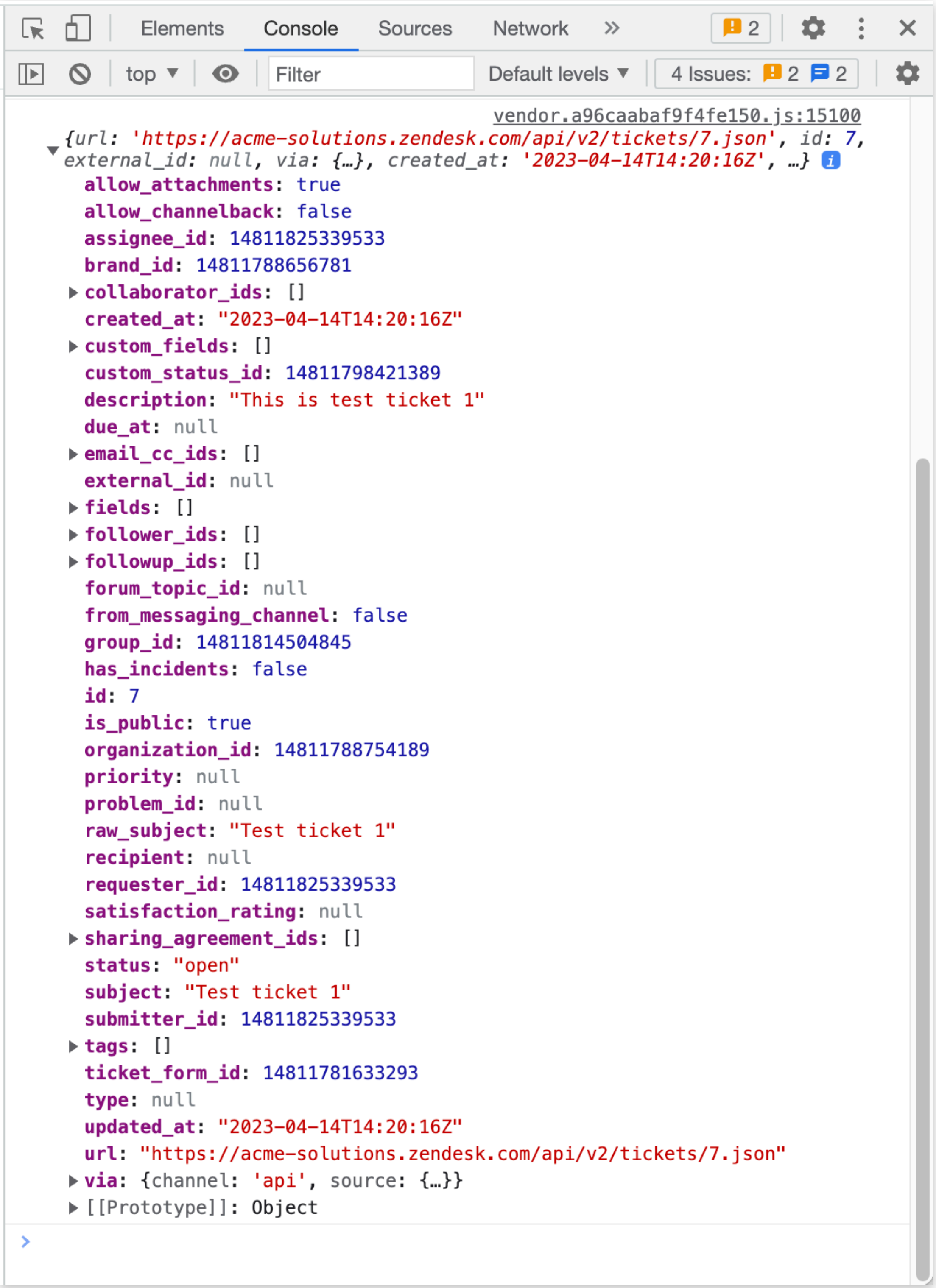
3. Press Enter.

If successful, the console lists the tickets as JavaScript objects.



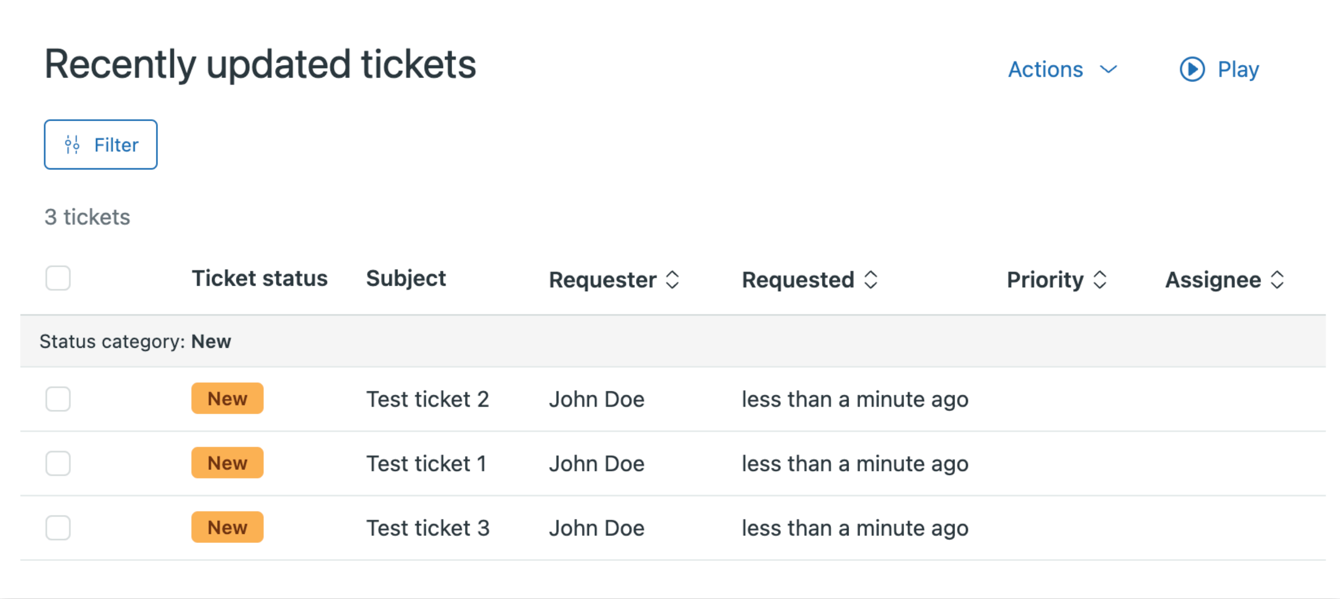
```
> for (let i = 1; i < 4; i++) {
    const subject = `Test ticket ${i}`
    const body = `This is test ticket ${i}`
    $.ajax({
      url: "/api/v2/tickets.json",
      contentType: "application/json",
      type: "POST",
      data: JSON.stringify({
        ticket: { subject: subject, comment: { body: body } }
      })
    }).done(data => {
      console.log(data.ticket)
    })
  }
<· ▶ {readyState: 1, getResponseHeader: f, getAllResponseHeaders: f, setRequest
     Header: f, overrideMimeType: f, …}
                                                  vendor.a96caabaf9f4fe150.js:15100
   ▶ {url: 'https://acme-solutions.zendesk.com/api/v2/tickets/5.json', id: 5, e
     xternal_id: null, via: {…}, created_at: '2023-04-14T14:20:16Z', …}
                                                  vendor.a96caabaf9f4fe150.js:15100
   ▶ {url: 'https://acme-solutions.zendesk.com/api/v2/tickets/6.json', id: 6, e
     xternal_id: null, via: {…}, created_at: '2023-04-14T14:20:16Z', …}
                                                  vendor.a96caabaf9f4fe150.js:15100
   ▶ {url: 'https://acme-solutions.zendesk.com/api/v2/tickets/7.json', id: 7, e
     xternal_id: null, via: {…}, created_at: '2023-04-14T14:20:16Z', …}
>
```

You can expand each object to see the ticket's properties.



```
                                                    vendor.a96caabaf9f4fe150.js:15100
▼{url: 'https://acme-solutions.zendesk.com/api/v2/tickets/7.json', id: 7,
 external_id: null, via: {…}, created_at: '2023-04-14T14:20:16Z', …} i
   allow_attachments: true
   allow_channelback: false
   assignee_id: 14811825339533
   brand_id: 14811788656781
  ▶collaborator_ids: []
   created_at: "2023-04-14T14:20:16Z"
  ▶custom_fields: []
   custom_status_id: 14811798421389
   description: "This is test ticket 1"
   due_at: null
  ▶email_cc_ids: []
   external_id: null
  ▶fields: []
  ▶follower_ids: []
  ▶followup_ids: []
   forum_topic_id: null
   from_messaging_channel: false
   group_id: 14811814504845
   has_incidents: false
   id: 7
   is_public: true
   organization_id: 14811788754189
   priority: null
   problem_id: null
   raw_subject: "Test ticket 1"
   recipient: null
   requester_id: 14811825339533
   satisfaction_rating: null
  ▶sharing_agreement_ids: []
   status: "open"
   subject: "Test ticket 1"
   submitter_id: 14811825339533
  ▶tags: []
   ticket_form_id: 14811781633293
   type: null
   updated_at: "2023-04-14T14:20:16Z"
   url: "https://acme-solutions.zendesk.com/api/v2/tickets/7.json"
  ▶via: {channel: 'api', source: {…}}
  ▶[[Prototype]]: Object
>
```

4. To view the tickets in the Agent Workspace, reload the browser tab. Then click **Views** > **Recently updated tickets**.



Recently updated tickets

3 tickets

□	Ticket status	Subject	Requester	Requested	Priority	Assignee
Status category: **New**						
□	New	Test ticket 2	John Doe	less than a minute ago		
□	New	Test ticket 1	John Doe	less than a minute ago		
□	New	Test ticket 3	John Doe	less than a minute ago		

When you're done trying out the API, delete the test tickets by selecting them in the list and clicking **Delete**.

Troubleshooting tips

If the script returned an error, try the following troubleshooting tips:

- If you get a 404 (Not Found) response, sign in to your Zendesk account before running the script.
- If you get a `TypeError: $.ajax not a function` error, ensure you run the script from the Agent Workspace in Zendesk Support.
- If you get a JavaScript error indicating that `$.ajax` is not defined, ensure you don't have any browser settings or plugins that block scripts from loading.

Next steps

The best way to learn how to use the Zendesk APIs is to try different requests on your own. You can use any of the following resources along the way:

- To see what's possible using Zendesk APIs, check out the API reference.
- Use Postman to explore the Zendesk APIs. Postman is a popular application for testing APIs and making API requests. See Exploring Zendesk APIs with Postman.
- Install curl to run example requests from the API reference documentation. curl is a command-line tool for making HTTP requests without a web browser. See Installing and using curl.
- Speed up development with an API client. An API client takes care of a lot of the low-level details of making requests and handling responses and lets you focus on the logic of your application.
- Learn about the most common tasks developers perform with the Ticketing API.
- Ask questions or look for answers in the Zendesk APIs community.

Join our developer community

Forum Blog Slack

Zendesk 181 Fremont Street, 17th Floor, San Francisco, California 94105

Privacy Policy | Terms & Conditions | System Status | Cookie Settings