

Zendesk Ticket Viewer

Tech Stack and Applications: DotNet Core 3.1, Visual Studio, Postman, xUnit [Unit Tests]

Zendesk API details:

API: <https://zccinternshiptest.zendesk.com/api/v2/tickets>

Authorization Header: Basic {email}/token:{token}

Project details:

Request:

API: <http://localhost:5000/api/v1.0/tickets>

Verb: GET

Query Parameters: These are optional fields

- pageNumber [Default Value: 1, Min Value: 1, Max Value: 25]
- pageSize [Default Value: 25, Min Value: 1, Max Value: 100]

http://localhost:5000/api/v1.0/tickets?pageNumber=1&pageSize=3

Save

GET http://localhost:5000/api/v1.0/tickets?pageNumber=1&pageSize=3 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	pageNumber	1			
<input checked="" type="checkbox"/>	pageSize	3			
	Key	Value	Description		

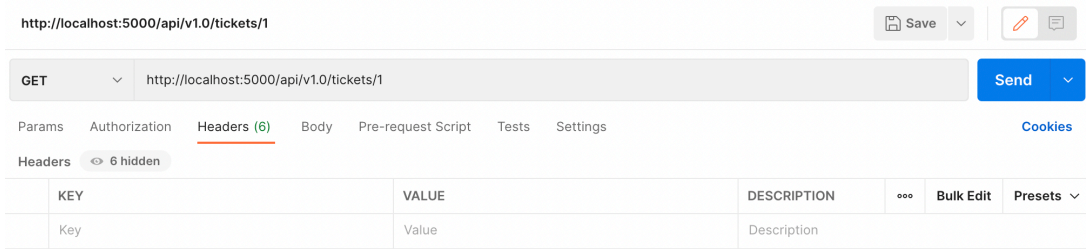
Response:

```
{
  "tickets": [
    {
      "url": "https://zccinternshiptest.zendesk.com/api/v2/tickets/1.json",
      "id": 1,
      "externalId": null,
      "createdAt": "2021-11-25T04:56:33Z",
      "updatedAt": "2021-11-25T04:56:34Z",
      "type": "incident",
      "subject": "Sample ticket: Meet the ticket",
      "rawSubject": "Sample ticket: Meet the ticket",
      "description": "Hi there,\n\nI'm sending an email because I'm having a problem setting up your new product. Can you help me troubleshoot?\n\nThanks,\n\nThe Customer\n\n",
      "priority": "Sample ticket: Meet the ticket",
      "status": "open",
      "recipient": null,
      "requesterId": "1524181961082",
      "submitterId": "1524278450401",
      "assigneeId": "1524278450401",
      "organizationId": null,
      "groupId": "4414094671255",
      "collaboratorIds": [],
      "followerIds": [],
      "emailCCIds": [],
      "forumTopicId": null,
      "problemId": null,
      "hasIncidents": false,
      "isPublic": true,
      "dueAt": null,
      "tags": [
        "sample",
        "support",
        "zendesk"
      ],
      "customFields": [],
      "sharingAgreementIds": [],
      "fields": [],
      "followupIds": [],
      "satisfactionRating": null,
      "ticketFormId": "1500003396922",
      "brandId": "1500002352562",
      "allowChannelBack": false,
      "allowAttachments": true
    }
  ],
  "pagingInfo": {
    "pageNumber": 1,
    "pageSize": 20,
    "totalRecords": 1
  }
}
```

Request:

API: <http://localhost:5000/api/v1.0/tickets/{ticketId}>

Verb: GET

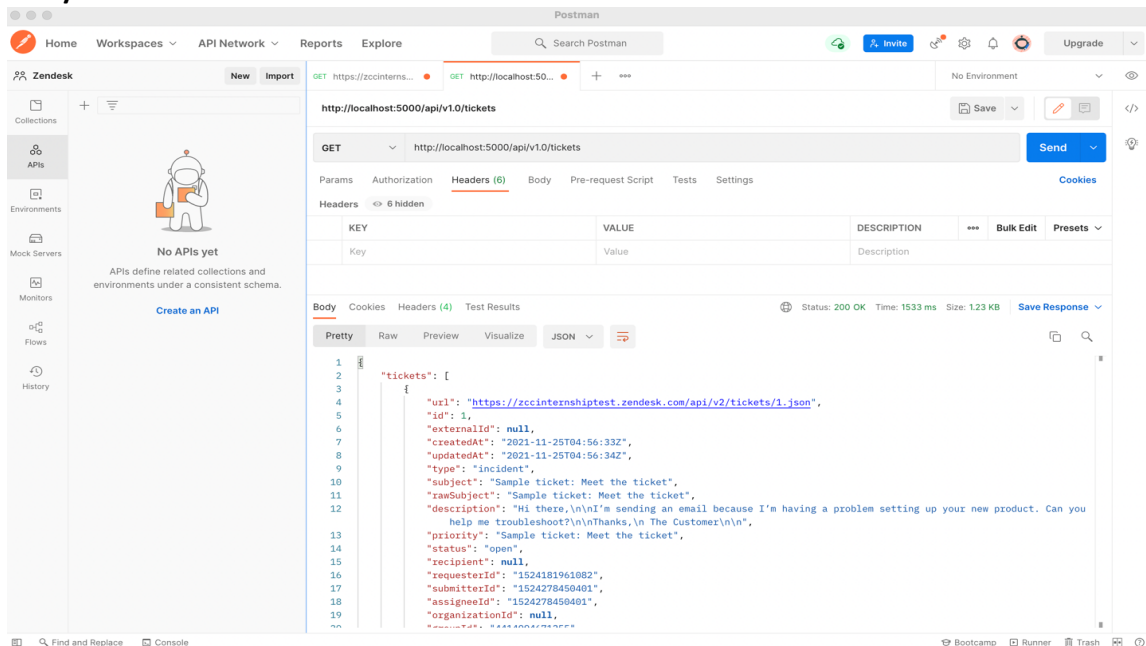


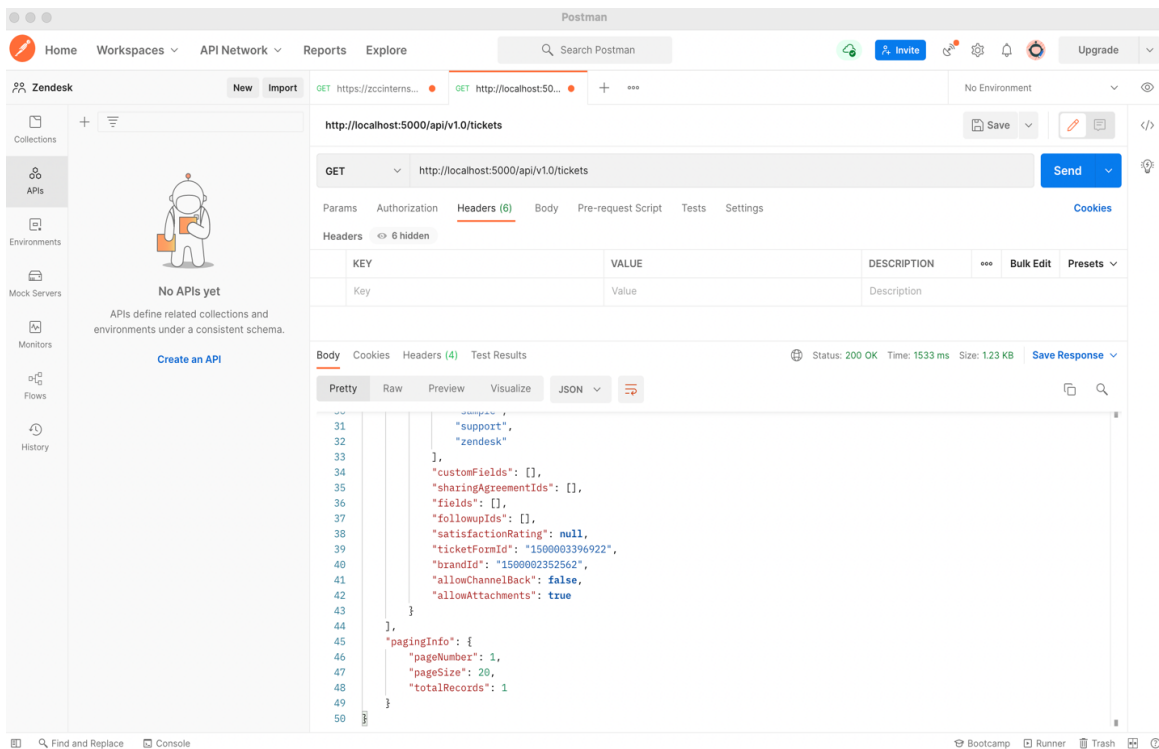
Response:

```
{
  "url": "https://zccinternshiptest.zendesk.com/api/v2/tickets/1.json",
  "id": 1,
  "externalId": null,
  "createdAt": "2021-11-25T04:56:33Z",
  "updatedAt": "2021-11-25T04:56:34Z",
  "type": "incident",
  "subject": "Sample ticket: Meet the ticket",
  "rawSubject": "Sample ticket: Meet the ticket",
  "description": "Hi there,\n\nI'm sending an email because I'm having a problem setting up your new product. Can you help me troubleshoot?\n\nThanks,\n\nThe Customer\n\n",
  "priority": "Sample ticket: Meet the ticket",
  "status": "open",
  "recipient": null,
  "requesterId": "1524181961082",
  "submitterId": "1524278450401",
  "assigneeId": "1524278450401",
  "organizationId": null,
  "groupId": "4414094671255",
  "collaboratorIds": [],
  "followerIds": [],
  "emailCCIds": [],
  "forumTopicId": null,
  "problemId": null,
  "hasIncidents": false,
  "isPublic": true,
  "dueAt": null,
  "tags": [
    "sample",
    "support",
    "zendesk"
  ],
  "customFields": [],
  "sharingAgreementIds": [],
  "fields": [],
  "followupIds": [],
  "satisfactionRating": null,
  "ticketFormId": "1500003396922",
  "brandId": "1500002352562",
  "allowChannelBack": false,
  "allowAttachments": true
}
```

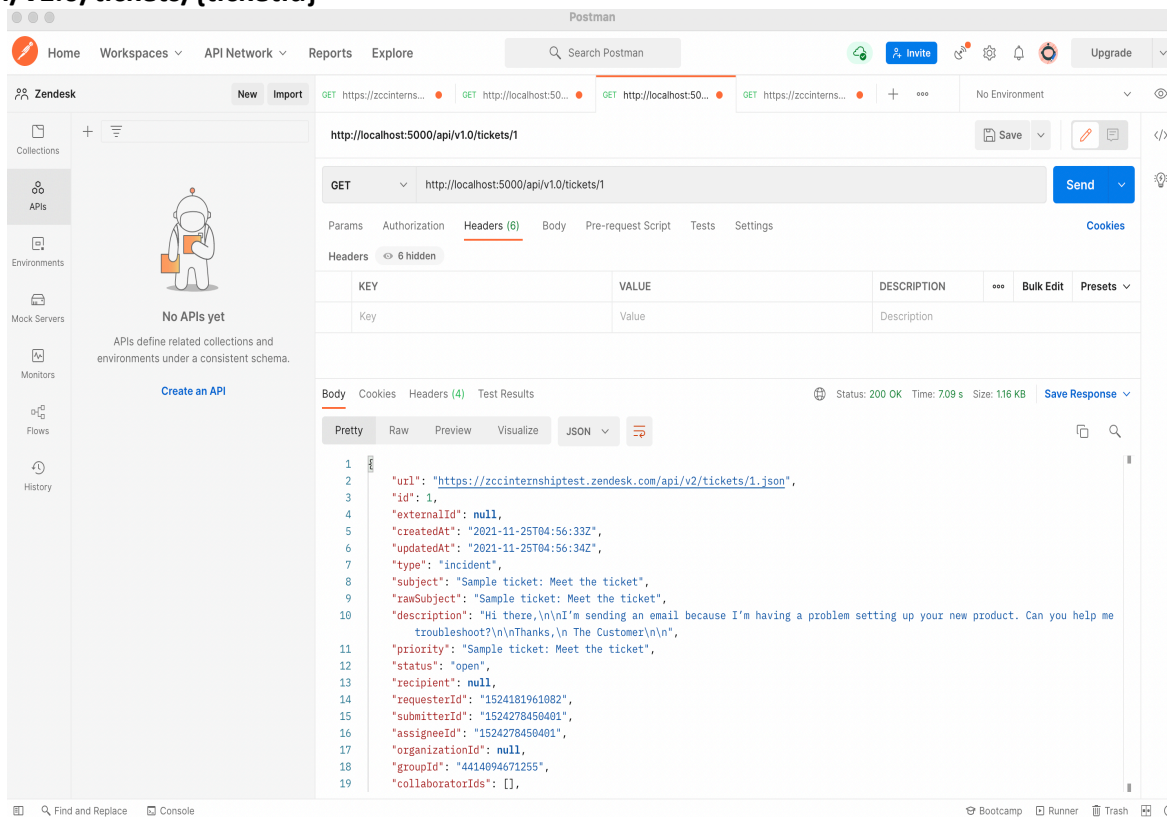
Output:

[api/v1.0/tickets](http://localhost:5000/api/v1.0/tickets)





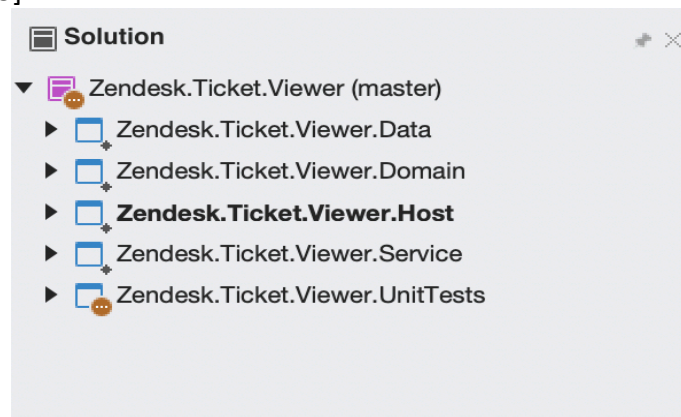
api/v1.0/tickets/{ticketId}



Solution details:

Structure:

- **Host:** This is the startup project and has all the Middleware along with Controller
- **Service:** Contains the service layer implementation. This layer gets request from Host layer and calls the Data layer for ticket lists.
- **Domain:** This project has the implementation of common Models, Contracts and Interface used by Service and Data Layer.
- **Data:** This project has the implementation of Web Caller client which calls the Zendesk API to get the tickets and returns the response after translation to service layer. The ZendeskAdapter implements interface IDataAdapter and using this approach we can easily replace the data layer by any other data store if it comes in future. The Authorization header details are hard coded in the project however we should use Consul [consul.io] or AWS Parameter Store for fetching the details from there and then using them in code.
- **UnitTests:** This layer has the unit test implementation of the project using xUnit, Moq, Builder Design Pattern [<https://codeburst.io/builder-pattern-and-moq-in-unit-tests-47281fa5b513>]



Unit Test is written using Builder Pattern. All the code is not covered in Unit Test because of time limitation.

